Meningkatkan Keamanan Basis Data dengan Kriptografi Kurva Eliptik El Gamal

Hayyu' Luthfi Hanifah 13512080[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]hayyuhanifah52@gmail.com

Abstrak—Salah satu cara untuk menjamin keamanan basis data adalah dengan mengatur hak akses untuk setiap pengguna basis data. Hal tersebut memastikan bahwa setiap pengguna hanya dapat mengakses dan mengubah bagian-bagaian yang sesuai dengan perannya. Akan tetapi, administrator basis data memiliki hak untuk mengakses dan mengubah seluruh bagian basis data. Padahal boleh jadi terdapat data yang seharusnya tidak diketahui oleh administrator. Oleh karena itu, dilakukanlah enkripsi pada data yang disimpan di basis data. Algoritma kriptografi kurva eliptik dapat digunakan untuk mengenkripsi data sebelum dikirim ke server basis data untuk disimpan. Algoritma kriptografi ini diimplementasikan pada modul aplikasi yang terpisah dari server basis data (*thick client*).

Kata kunci— DBA *insider threat*, keamanan basis data, kontrol akses, kriptografi kurva eliptik.

I. PENDAHULUAN

Data yang tersimpan pada suatu basis data boleh jadi merupakan data penting dan tidak boleh diketahui oleh sembarang orang. Untuk memastikan data tidak diakses oleh sembarang orang, dapat dilakukan pengaturan hak akses untuk setiap pengguna basis data tersebut. Pengguna dengan peran (*role*) tertentu hanya dapat (berhak) mengakses bagian data tertentu pula. Akan tetapi, pada mekanisme pengaturan peran pengguna ini masih memungkinkan administrator basis data untuk mengakses semua data (termasuk data yang sebenarnya tidak boleh diketahui oleh administrator).

Untuk meningkatkan keamanan basis data, selain menerapkan pemberian hak akses, dapat juga dilakukan enkripsi terhadap data yang akan disimpan. Misalkan, pengguna suatu basis data dibagi ke dalam 2 peran. Peran pertama merupakan pengguna yang cenderung melakukan proses input data. Peran kedua merupakan pengguna yang cenderung melakukan *query* untuk mendapatkan data yang sudah dimasukkan oleh pengguna pertama. Dengan menggunakan kriptografi kunci publik, data yang dimasukkan oleh pengguna pertama dienkripsi terlebih dahulu dengan kunci publik pengguna kedua. Lalu, untuk mendapatkan data yang dimasukkan oleh pengguna pertama, pengguna kedua harus melakukan dekripsi dengan kunci privatnya.

II. DASAR TEORI

A. Kontrol Akses Basis Data

Suatu sistem biasanya digunakan oleh beberapa orang pengguna. Masing-masing pengguna dapat mengakses basis data sistem sesuai dengan keperluan dan hak akses yang dimilikinya. Pengaturan hak akses perlu dilakukan untuk menjaga basis data tetap konsisten walau digunakan oleh banyak orang.

Sistem manajemen basis data telah menyediakan mekanisme yang dapat digunakan oleh administrator basis data untuk mengatur hak akses pengguna basis data. Misalnya, pada MySQL pemberian hak akses dapat dilakukan dengan mengeksekusi *query* berikut

```
GRANT <privilege list>
ON <part name> TO <user/role list>;
```

Privilege list dapat diisi dengan SELECT, INSERT, UPDATE, atau DELETE. *Part name* diisi dengan nama bagian basis data (nama tabel, nama kolom, nama *view*, dll). *User/role list* diisi dengan nama-nama pengguna atau peran pengguna yang akan diberi hak akses.

Selain memberikan hak akses, administrator basis data juga dapat menarik kembali hak akses yang telah diberikan kepada pengguna tertentu. Hal ini dapat dilakukan dengan mengeksekusi *query* berikut

```
REVOKE <privilege list>
ON <part name>
FROM <user/role list> <option>;
```

Seorang pengguna basis data (misal, userA) dapat memberikan hak akses yang sama dengan hak aksesnya kepada pengguna lain (misal, userB) apabila userA memiliki GRANT OPTION (hak untuk memberikan hak akses kepada pengguna lain). Jika administrator basis data menarik hak akses tertentu dari userA, maka hak akses yang sama juga akan ditarik kembali dari userB. Hal tersebut dikenal sebagai *cascading revocation*. Untuk mencegah *cascading revocation*, *option* pada *query* penarikan hak akses dapat diisi dengan RESTRICT (selama userA belum menarik hak akses yang diberikannya kepada userB, administrator tidak dapat menarik hak akses userA).

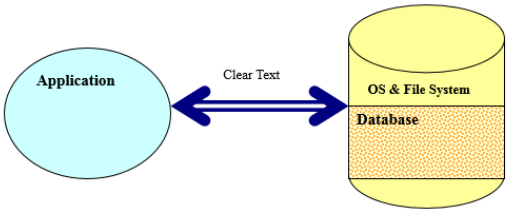
Administrator basis data dapat dikatakan sebagai *superuser* pada suatu basis data. Administrator memiliki hak untuk mengakses dan melakukan perubahan pada basis data serta mengatur hak akses untuk pengguna basis data yang lain. Dengan demikian, administrator dapat melihat isi setiap tabel yang ada pada basis data. Hal ini dapat dianggap sebagai celah bagi keamanan basis data karena boleh jadi ada data yang seharusnya tidak diketahui oleh administrator. Celah keamanan ini dikenal sebagai *Database Administrator (DBA) insider threats*.

B. Enkripsi Basis Data

Untuk memastikan data yang disimpan hanya diketahui oleh orang-orang yang berhak (*data confidentiality*), data tersebut dapat disimpan dalam keadaan terenkripsi. Walaupun administrator basis data dapat melakukan *query* SELECT pada tabel data, dia tidak akan mendapat informasi bermakna dari tabel tersebut selama tidak memiliki kunci untuk melakukan dekripsi.

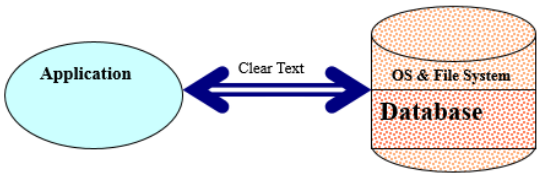
Ada beberapa opsi yang dapat dilakukan untuk mengenkripsi basis data:

a. *Full database encryption*, data dari aplikasi dikirimkan ke server basis data dalam bentuk *plaintext*. Data tersebut kemudian dienkripsi pada server basis data sebelum disimpan ke dalam tabel.



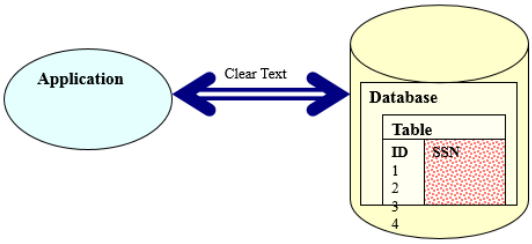
Gambar 1 Enkripsi dilakukan pada seluruh bagian basis data,

b. *Operating system or file system encryption*, data dari aplikasi dikirimkan ke server basis data dalam bentuk *plaintext*. Data tersebut kemudian dienkripsi pada level sistem operasi atau *file system* di server basis data.



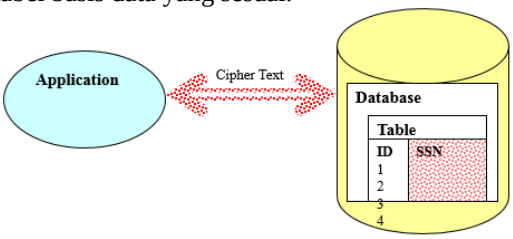
Gambar 2 Enkripsi dilakukan pada level sistem operasi server basis data.

c. *Field level encryption by database or middleware*, data dari aplikasi dikirimkan ke server basis data dalam bentuk *plaintext*. Sebagian dari data tersebut dienkripsi lalu disimpan ke dalam tabel basis data (tidak semua kolom tabel basis data berisi data yang terenkripsi).



Gambar 3 Hanya sebagian basis data yang dienkripsi.

d. *Field level encryption by application*, data dari aplikasi dikirimkan ke server dalam keadaan sudah terenkripsi. Data tersebut kemudian disimpan pada tabel basis data yang sesuai.



Gambar 4 Data dienkripsi oleh aplikasi sebelum dikirim ke server.

Dari keempat opsi tersebut, opsi yang dapat diambil untuk mengatasi *DBA insider threats* adalah opsi keempat karena data yang akan disimpan pada server basis data sudah dienkripsi oleh aplikasi. Selain itu, dengan menerapkan opsi keempat, kontrol akses terhadap basis data dapat diterapkan secara granular (setiap pengguna benar-benar hanya dapat mengakses bagian-bagian basis daya yang dibutuhkannya).

C. Kriptografi Kurva Eliptik

Enkripsi dan dekripsi basis data dapat dilakukan dengan berbagai jenis algoritma kriptografi. Algoritma kriptografi yang biasa digunakan pada enkripsi basis data antara lain adalah DES, AES, dan RC.

Kriptografi kurva eliptik merupakan salah satu algoritma kriptografi kunci asimetri, yaitu algoritma kriptografi yang melibatkan kunci privat dan kunci publik. Algoritma ini menggunakan kurva eliptik dengan parameter tertentu untuk melakukan proses enkripsi dan dekripsi. Selain dapat digunakan untuk melakukan enkripsi, kurva eliptik juga dapat digunakan pada pembangkitan tanda tangan digital.

Pada makalah ini akan dibahas mengenai penggunaan algoritma kriptografi kurva eliptik El Gamal untuk meningkatkan keamanan basis data. Secara singkat, algoritma enkripsi kriptografi kurva eliptik El Gamal adalah sebagai berikut

a. Dua pihak yang akan saling berkirim pesan menyepakati satu kurva eliptik lalu memilih titik basis dari titik-titik yang ada pada kurva eliptik tersebut.
b. Masing-masing pihak memilih kunci privat lalu membangkitkan kunci publik dengan mengalikan kunci privat dengan titik basis.
c. *Plaintext* ditransformasikan ke titik-titik pada kurva eliptik yang digunakan (*encoding*), lalu dienkripsi

dengan kunci publik pihak yang akan dikirimi pesan tersebut.

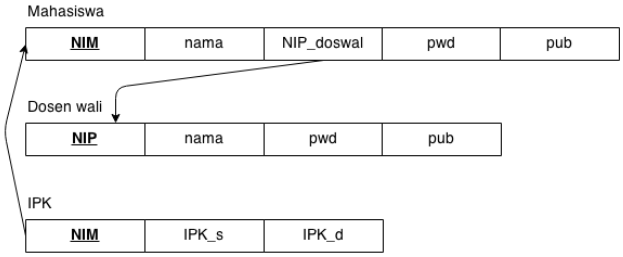
Sedangkan untuk mendekripsi *ciphertext* dilakukan dengan menggunakan kunci privat penerima pesan (*ciphertext*), kemudian dilakukan penerjemahan (*decoding*) dari titik-titik kurva eliptik menjadi karakter-karakter *plaintext*.

Algoritma kriptografi dengan menggunakan kurva eliptik ini tidak lazim digunakan untuk enkripsi basis data. Oleh karena itu, penulis tertarik untuk mengkaji penggunaan algoritma ini dalam proses enkripsi dan dekripsi basis data.

III. Implementasi Kriptografi Kunci Publik pada Basis Data Relasional

A. Rancangan Basis Data

Basis data yang dijadikan kasus uji coba pada makalah ini adalah basis data akademik sederhana. Basis data ini menyimpan data mahasiswa, data dosen wali, dan data indeks prestasi kumulatif setiap mahasiswa. Skema relasional basis data ini kurang lebih sebagai berikut



Gambar 5 Skema basis data akademik sederhana.

dengan `pwd` adalah nilai hash dari *password* pengguna, `pub` adalah kunci publik pengguna, `IPK_s` adalah IPK yang dienkripsi dengan kunci publik mahasiswa, dan `IPK_d` adalah IPK yang dienkripsi dengan kunci publik dosen wali dari mahasiswa yang bersangkutan. Nilai IPK bukan merupakan data yang dapat diinput oleh dosen wali maupun mahasiswa.

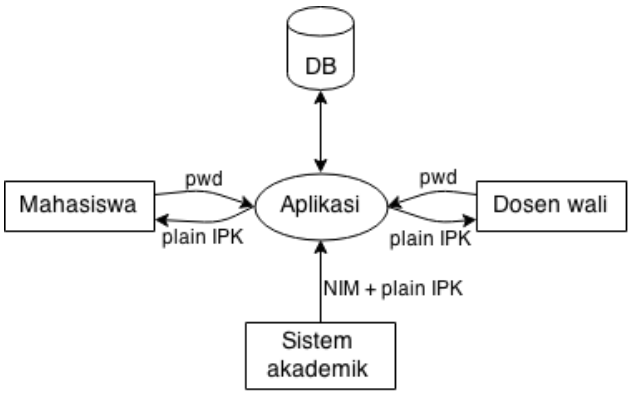
Peran pengguna basis data ini (selain administrator basis data) dibagi menjadi dua, yaitu mahasiswa dan dosen wali. Peraturan terkait hak akses yang dimiliki oleh masing-masing pengguna basis data adalah

a. Setiap mahasiswa hanya dapat mengakses IPK-nya sendiri (tidak dapat mengakses IPK mahasiswa lain).
b. Setiap dosen wali dapat mengakses IPK semua mahasiswa yang menjadi tanggung jawabnya, namun tidak dapat mengakses IPK mahasiswa yang menjadi tanggung jawab dosen lain.

Nilai IPK diperoleh dari perhitungan yang dilakukan di sistem akademik. Agar nilai IPK dapat diakses oleh mahasiswa dan dosen, enkripsi data nilai IPK harus dilakukan dengan menggunakan kunci publik mahasiswa dan kunci publik dosen walinya.

B. Rancangan Modul Aplikasi

Modul aplikasi yang terhubung dengan basis data akademik dapat memfasilitasi mahasiswa dan dosen wali untuk melakukan *query* SELECT terhadap nilai IPK serta memfasilitasi sistem akademik untuk melakukan *query* INSERT atau UPDATE nilai IPK mahasiswa. Karena nilai IPK disimpan pada basis data dalam keadaan terenkripsi, maka aplikasi ini perlu melakukan proses enkripsi dan dekripsi data dengan kriptografi kurva eliptik. Parameter kurva eliptik (nilai variabel *a*, *b*, dan *p* pada persamaan $y^2 = x^3 + ax + b \bmod p$) yang digunakan untuk enkripsi dan dekripsi dijadikan sebagai variabel global pada aplikasi ini. Nilai variabel *p* dipilih sedemikian rupa agar dapat memfasilitasi pilihan kunci privat pengguna. Titik basis yang diambil dari himpunan titik-titik pada kurva eliptik juga disimpan sebagai variabel global pada aplikasi ini. Diagram konteks aplikasi ini adalah sebagai berikut



Gambar 6 Diagram konteks aplikasi.

Saat pertama kali terdaftar pada sistem, mahasiswa dan dosen wali dapat mengatur *password* baru untuk akun mereka. Aplikasi akan melakukan *hash* pada *password* ini kemudian mengirimkan hasilnya ke server basis data untuk disimpan. *Password* ini sekaligus digunakan sebagai kunci privat pengguna, yaitu dengan menjumlahkan representasi desimal dari karakter-karakter ASCII penyusun *password*. Setelah mengatur *password* baru, aplikasi dapat membangkitkan kunci publik untuk pengguna yang bersangkutan lalu mengirimkannya ke server basis data.

Jika sistem akademik akan memperbarui nilai IPK seorang mahasiswa, sistem memberikan NIM dan IPK mahasiswa tersebut ke aplikasi. Kemudian aplikasi akan mencari kunci publik mahasiswa yang bersangkutan dan kunci publik dosen walinya. Saat sudah mendapatkan kedua kunci publik tersebut, aplikasi akan menghasilkan dua *cipher* IPK untuk disimpan pada basis data.

Ketika seorang mahasiswa atau dosen ingin mengakses nilai IPK, mereka harus terlebih dahulu berhasil *login* dengan memasukkan *password* yang tepat. Proses tersebut selain untuk autentikasi pengguna juga digunakan untuk autorisasi pengguna. Jika sudah berhasil login, aplikasi akan melakukan *query* ke basis data untuk mengambil nilai *cipher* IPK milik mahasiswa dengan NIM yang sesuai. *Cipher* IPK yang diambil disesuaikan dengan peran pengguna yang memintanya. Jika yang meminta adalah dosen wali, maka *cipher* IPK diambil dari kolom `IPK_d` pada tabel IPK. Sedangkan jika pengguna yang meminta adalah mahasiswa, maka cipher IPK diambil dari kolom `IPK_s` pada tabel IPK. Setelah mendapatkan *cipher* IPK dari server basis data, aplikasi melakukan proses dekripsi dengan kunci privat pengguna.

IV. ANALISIS

Karena kunci privat pengguna diperoleh dengan menjumlahkan representasi desimal dari karakter-karakter ASCII penyusun *password*, ada kemungkinan satu kunci privat digunakan oleh dua atau lebih pengguna. Hal ini terjadi karena *password* yang berbeda boleh jadi tersusun atas karakter-karakter ASCII yang sama.

Agar aspek *confidentiality* tetap terjaga, untuk setiap pasang mahasiswa dan dosen wali dapat diatur supaya memiliki titik basis yang berbeda. Dengan demikian, walau kunci privatnya sama, kunci publiknya masih mungkin berbeda. Selain titik basis, ada juga parameter *k* pada algoritma kriptografi kurva eliptik El Gamal yang nilainya dapat disesuaikan.

Sistem yang dirancang pada makalah ini dapat dikategorikan sebagai aplikasi *thick client*. Mayoritas proses dilakukan pada aplikasi yang dijalankan pada komputer pengguna. Data yang dikirimkan ke server merupakan data yang sudah dienkripsi bagian-bagian pentingnya. Hal ini tentu dapat mengurangi ancaman *DBA insider* seperti yang telah disinggung pada dasar teori.

Namun demikian, aplikasi *thick client* semacam ini tetap memiliki celah keamanan. Karena *password* digunakan umtuk autentikasi dan autorisasi sekaligus, maka *password* menjadi sesuatu yang harus benar-benar dijaga agar tidak disalahgunakan orang lain. Pengguna sebaiknya tidak menggunakan fasilitas pengingat *password* untuk aplikasi ini.

Pemakaian *password* sebagai kunci privat mewajibkan pengguna untuk tidak boleh lupa *password*. Jika pengguna lupa, aplikasi tidak akan dapat melakukan dekripsi nilai IPK yang tersimpan di basis data.

V. KESIMPULAN

Algoritma kriptografi kurva eliptik dapat digunakan untuk mengenkripsi data sebelum disimpan pada basis data. Dengan demikian, administrator basis data tidak akan dapat membaca data tersebut walau administrator memiliki hak untuk mengakses seluruh isi basis data.

Rancangan aplikasi pada makalah ini masih perlu dikembangkan lagi terutama untuk menangani masalah manajemen kunci (agar tidak ada kunci privat yang sama). Pada rancangan yang akan datang, dapat pula dipertimbangkan mekanisme lain untuk mendapat kunci privat selain dari *password* serta mekanisme penyimpanan kunci yang lebih baik.

REFERENCES

[1] Abraham Silberschatz, et al. *Database System Concepts Sixth Edition.* New York: McGraw-Hill, 2011.

[2] Jingmin He, Min Wang. "Cryptography and Relational Database Management System". IBM T. J. Watson Research Center.

[3] Ralph Durkee. "Protecting Sensitive Information with Database Encryption". New York: Durkee Consulting, Inc.

[4] Rinaldi Munir. "Bahan Kuliah IF4020 Kriptografi: Elliptic Curve Cryptography".

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 11 Mei 2015

Hayyu' Luthfi Hanifah
NIM 13512080