Desain Implementasi Teknik Kriptografi untuk Pengamanan Basis Data Perusahaan

Chitra Hapsari[1] , Anisa Herdiani[2], dan Ulya Raniarti[3]

Departemen Teknik Informatika
Institut Teknologi Bandung
Jalan Ganesha 10 Bandung 40132

E-mail : if12007@students.if.itb.ac.id[1] ,if12037@students.if.itb.ac.id[2] ,
if12052@students.if.itb.ac.id[3]

Abstrak

Keamanan pada basis data telah menjadi kebutuhan yang penting pada suatu perusahaan. Kebutuhan ini timbul dari semakin banyaknya ancaman terhadap data sensitif yang terdapat pada basis data. Teknik kriptografi merupakan salah satu alternatif solusi yang dapat digunakan dalam pengamanan basis data. Akan tetapi, pengembangan strategi kriptografi pada basis data membutuhkan banyak pertimbangan. Makalah ini memaparkan langkah-langkah implementasi teknik kriptografi dalam basis data, mencakup analisis lingkungan, desain solusi, dan persoalan-persoalan yang ditemui dalam menentukan desain pengamanan basis data.

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

1. Pendahuluan

Basis data merupakan tempat penyimpanan data penting yang dibutuhkan untuk menjamin kelancaran aktivitas suatu perusahaan. Data penting dan vital yang tersimpan pada basis data seringkali menjadi target empuk bagi para penyerang. Serangan yang terjadi dapat dilakukan oleh pihak luar (*hacker*) maupun pihak dalam (pegawai yang tidak puas) [7)]. Selama ini, mekanisme pengamanan basis data diimplementasikan dengan menggunakan kontrol akses terhadap basis data tersebut. Akan tetapi, dengan berkembangnya penggunaan jaringan untuk pertukaran data, diperlukan strategi pengamanan yang lebih kuat daripada sekedar mekanisme kontrol akses. Alasan lain diperlukannya pengamanan basis data adalah berlakunya Undang-Undang yang mengatur perihal kerahasiaan data pelanggan yang biasa disimpan pada basis data perusahaan. Salah satu contohnya adalah peraturan HIPAA (Health Insurance Portability and Accountability Act) yang menstandarkan keamanan data medis dan data individual lainnya [2)]. Dengan adanya UU ini, tiap rumah sakit akan memiliki tanggung jawab lebih pada keamanan datanya.

Salah satu cara untuk mengamankan data pada basis data adalah dengan menggunakan teknik kriptografi yang diterapkan pada data tersebut. Pengamanan menggunakan kriptografi memerlukan banyak pertimbangan dan strategi seperti yang akan dibahas pada makalah ini.

2. Pengamanan Basis data dengan Kriptografi

Memperkuat pengamanan basis data melalui kiptografi seringkali mengakibatkan performansi basis data menjadi menurun, khususnya dalam hal waktu yang diperlukan

untuk mengakses data. Sementara, pengaksesan data yang cepat menjadi hal yang sangat penting bagi proses bisnis sebuah organisasi/perusahaan. Oleh karena itu, diperlukan teknik pengamanan basis data menggunakan kriptografi yang dapat meminimalisir penurunan performansi basis data.

Secara garis besar, terdapat dua tujuan dari pengamanan basis data :

1. *Melindungi kerahasiaan data*

Tujuan utama dari kriptografi pada basis data adalah melindungi data dari pengaksesan oleh pihak-pihak yang tidak memiliki hak otorisasi terhadap data tersebut. Melindungi kerahasiaan data dapat dilakukan dengan melakukan enkripsi terhadap data sensitif. Perlindungan dilakukan dengan cara menjaga kunci enkripsi-dekripsi dari penyerang yang berusaha memperoleh kunci tersebut secara langsung (*direct access*) maupun secara tidak langsung (*indirect access*). *Direct access* dapat dilakukan dengan menduplikasi kunci, sementara *indirect access* dilakukan dengan mengambil ciperteks dari basis data, kemudian berusaha menemukan plainteks dan kuncinya dengan cara kriptanalisis.[4)]

2. *Menjamin integritas basis data*

Kriptografi dapat mendeteksi modifikasi data oleh pihak yang tidak berhak. Salah satu caranya adalah dengan menggunakan algoritma kunci simetrik. Data terenkripsi yang tidak terdekripsi dengan baik menandakan telah terjadi kerusakan pada data yang dilakukan oleh pihak yang tidak memiliki hak otorisasi untuk memodifikasi data. Sayangnya, cara tersebut tidak dapat mengatasi penyerang yang melakukan penukaran baris ciperteks pada basis data atau menukar informasi yang dimodifikasi dengan informasi milik orang lain. Ketika didekripsi, nilainya akan tetap valid namun sesungguhnya nilai tersebut sudah bukan lagi nilai awal. Cara yang lebih baik adalah dengan menggunakan *Message Authentication Code* (MAC). MAC membangkitkan sebuah ID unik untuk untuk setiap plainteks berdasarkan nomor baris (*row*) pada basis data. Ketika data yang dimodifikasi dan MAC-nya dimasukkan ke tabel, basis data akan memastikan bahwa nilai MAC adalah benar untuk data tersebut, jika tidak basis data akan menolak modifikasi yang dilakukan. [4)]

Terdapat tiga level enkripsi basis data yang meliputi :

1. *Enkripsi pada level penyimpanan (storage)*

Enkripsi data dilakukan pada subsistem *storage* (penyimpanan), baik pada level file maupun pada level blok. Enkripsi level ini cocok untuk mengenkripsi file, folder, media *storage* dan media *tape*. Akan tetapi, serangan yang dapat diatasi hanya terbatas pada serangan yang berupa pencurian media dan sistem penyimpanan. Enkripsi pada level *storage* tidak mampu menangani serangan pada level basis data dan level aplikasi. [8)]

2. *Enkripsi pada level basis data*

Enkripsi dilakukan pada saat data ditulis dan dibaca dari basis data. Enkripsi jenis ini dilakukan pada level kolom pada tabel basis data. Level ini melindungi data pada Database Management System (DBMS) dari berbagai macam serangan. Diperlukan integrasi pada level basis data, termasuk modifikasi skema dan penggunaan *trigger* dan *store procedure* dalam proses enkripsi-dekripsi. Diperlukan beberapa pertimbangan dalam menerapkan strategi ini untuk mengatasi dampak enkripsi pada performansi basis data. Pertimbangan tersebut meliputi

pemilihan *fields* yang sensitif untuk dienkripsi, pemilihan *hardware* untuk meningkatkan pengamanan dan mereduksi penurunan performansi akibat proses kriptografi. Kelemahan utama pada enkripsi jenis ini adalah tidak bisa mengatasi serangan pada level aplikasi, karena fungsi enkripsi hanya terdapat pada level DBMS. [8)]

3. Enkripsi pada level aplikasi

Aplikasi menangani proses enkripsi data. Kelebihannya adalah tidak terjadi penurunan performansi pada basis data, karena DBMS tidak menangani enkripsi data. Akan tetapi, ketika terjadi perubahan strategi enkripsi atau perubahan data yang dienkripsi, akan banyak terjadi modifikasi pada level aplikasi.

Pengamanan basis data menggunakan kriptografi memiliki beberapa resiko dalam implementasinya, yaitu :

1. Hilangnya kunci

Resiko yang paling fatal akibatnya adalah hilangnya kunci, baik karena terhapus, *corrupted*, maupun secara tidak sengaja terbuang. Seiring dengan hilangnya kunci, data terenkripsi tidak akan dapat didekripsi lagi sehingga dapat dikatakan data tersebut hilang. Sayangnya, tidak terdapat *undelete* atau program *data recovery* yang dapat meng-*undo* proses enkripsi. Hanya karena kehilangan beberapa bit kunci, sejumlah *mega byte* data menjadi tidak berarti lagi. Oleh karena itu, manajemen kunci menjadi hal yang vital pada enkripsi basis data. [4)]

2. Lemahnya manajemen kunci

Kelemahan *tools* dan prosedur pada manajemen kunci menjadikan keamanan secara keseluruhan memiliki resiko yang cukup besar. Jika seorang penyerang dapat mengakses kunci baik secara langsung maupun tidak langsung, maka kriptografi pada basis data runtuh seketika. Jika pembangkit kunci tidak melakukan randomisasi dengan baik, maka penyerang akan dengan mudah menganalisis kunci yang digunakan. [4)]

3. Cacat (Bug) dalam implementasi

Jika data lain yang digunakan pada proses enkripsi seperti *Initialization Vector* (IV) tidak begitu diperhatikan strateginya, maka penyerang akan dapat dengan mudah melihat pola enkripsi, yang akhirnya dapat mendekripsi data yang diinginkan. Kasus lain yang dapat terjadi adalah ketika data yang ditulis pada log tidak dibersihkan dari memori sehingga akan menjadi celah tersendiri bagi penyerang.[4)]

Walaupun manajemen kunci telah sempurna dan tidak terdapat bug pada implementasi, akses secara tidak langsung terhadap kunci masih menjadi ancaman bagi keamanan basis data. Oleh karena itu, penting untuk mendesain dan mengimplementasikan infrastruktur kriptografi dengan benar.

3. Langkah-langkah Implementasi Pengamanan Basis data

Teknik kriptografi merupakan salah satu solusi yang dapat dipilih untuk memproteksi basis data perusahaan. Untuk menerapkan teknik kriptografi sebagai bagian yang terintegrasi dengan *security policy* perusahaan, terdapat tiga tahapan yang harus dilakukan yaitu analisis lingkungan, desain solusi dan implementasi.

Berikut akan dijabarkan langkah-langkah untuk tahapan analisis lingkungan dan desain solusi. Tahapan implementasi dilakukan dengan cara menerapkan solusi yang telah

dipilih dengan mempertimbangkan hasil analisis.

3.1 Analisis Lingkungan

Tahapan analisis lingkungan bertujuan untuk mendapatkan gambaran menyeluruh mengenai struktur sistem informasi perusahaan dan kebutuhan pengamanan basis data.

Langkah pertama yang harus dilakukan adalah mengidentifikasi data sensitif yang perlu diproteksi. Pada dasarnya ada dua macam data yang harus diproteksi. Yang pertama adalah data yang terkait dengan kepentingan internal perusahaan, yaitu data yang mengandung informasi mengenai strategi bisnis perusahaan tersebut. Yang kedua adalah data yang berkaitan dengan kepentingan pelanggan. Setiap pelanggan mempunyai hak untuk dijaga kerahasiaan datanya, terutama data yang dapat mengundang pihak ketiga untuk merugikan kepentingan pelanggan seperti informasi kartu kredit, rekening bank, dan kesehatan. Karena adanya *enterprise boundary* seperti yang telah dijelaskan di atas, sangat penting untuk meminimalkan jumlah data yang perlu dienkripsi. Enkripsi terhadap data yang sebenarnya tidak perlu diproteksi akan menurunkan performansi sistem karena enkripsi dan dekripsi data akan menambah beban proses yang harus dijalankan.

Selanjutnya, perlu dilakukan analisis terhadap aliran data perusahaan. Hal ini dapat dilakukan dengan mempelajari arsitektur sistem perusahaan yang biasanya terdiri dari multi-aplikasi dan multi-basis data. Informasi ini akan memberikan gambaran mengenai aliran data pada aplikasi serta komponen sistem yang mengolah data tersebut. Arsitektur sistem dan arsitektur jaringan akan sangat mempengaruhi pemilihan teknik pengamanan yang diperlukan. Sebagai contoh, perusahaan berbasis *e-bussiness* akan memerlukan teknik yang berbeda dengan perusahaan yang jaringannya hanya terdiri dari jaringan internal saja.

Identifikasi pengguna juga merupakan langkah yang perlu dilakukan dalam tahap analisis untuk dijadikan dasar dalam pengaturan *user management*. Definisi pengguna meliputi *end user*, administrator basis data, serta pihak-pihak lain yang diijinkan untuk mengoperasikan data yang diproteksi.

Terakhir, diperlukan juga identifikasi terhadap ancaman-ancaman yang potensial terhadap basis data. Seperti yang telah disebutkan sebelumnya, ancaman tidak hanya datang dari pihak eksternal perusahaan melainkan juga bisa datang dari pihak internal yang seharusnya tidak terautorisasi mengakses informasi yang disimpan dalam basis data.

Keseluruhan tahapan analisis sebaiknya juga dilakukan dengan mempelajari *security policy* yang telah diterapkan di perusahaan.

3.2 Desain Solusi

Setelah mendapatkan daftar kebutuhan pengamanan basis data, langkah yang selanjutnya dilakukan adalah mendesain solusi pengamanan basis data dengan teknik kriptografi.

Berdasarkan perkembangan teknologi pengamanan saat ini, terdapat dua strategi alternatif yang dapat digunakan yaitu dengan enkripsi secara internal dalam basis data dengan memanfaatkan fitur yang telah didukung oleh DBMS atau dengan

melakukan enkripsi secara eksternal di luar basis data.

1. *Enkripsi secara internal dalam basis data*

Strategi ini merupakan strategi pengamanan yang paling sederhana karena dilakukan hanya dengan memanfaatkan fitur enkripsi yang telah digunakan oleh DBMS yang bersangkutan atau dengan menggunakan produk *add-on* yang menambahkan fitur enkripsi pada DBMS yang belum memiliki kemampuan tersebut.

Dengan strategi ini, proses enkripsi dan dekripsi data hanya akan berlaku secara internal. Artinya, data dimasukkan ke dalam basis data dalam bentuk plainteks dan akan meninggalkan basis data dalam bentuk plainteks pula. Strategi ini diterapkan dengan menggunakan *database procedure call.* Salah satu contoh penggunaannya adalah dengan menerapkan *trigger* setiap kali pengguna memanggil operasi *select* [4)]. Operasi *select* terhadap data yang diproteksi akan memicu *trigger* untuk mengenkripsi data tersebut. Jika pengguna yang bersangkutan adalah pengguna yang memiliki hak autorisasi untuk melihat data, maka data akan didekripsi ketika ditampilkan. Jika sebaliknya, data akan ditampilkan masih dalam bentuk terenkripsi.

Enkripsi internal basis data dapat dilakukan dengan *whole database encryption* atau dengan *column encryption* [5)]. *Whole database encryption* menerapkan prinsip enkripsi *all-or-nothing* karena hanya ada dua pilihan bagi pengguna yaitu mengenkripsi basis data secara keseluruhan atau tidak melakukan enkripsi sama sekali. Dengan menggunakan *column encryption*, pengguna dapat memilih data mana yang perlu dienkripsi dan mana yang tidak. Strategi ini dapat dikatakan lebih natural dari yang pertama sebab pada umumnya tidak semua data yang disimpan dalam basis data perlu diproteksi dengan enkripsi.

Strategi enkripsi basis data secara internal memiliki beberapa kelemahan. Proses enkripsi dan dekripsi akan menambah beban proses yang harus dijalankan oleh sistem sehingga performansi DBMS akan menurun cukup tajam. Degradasi performansi ini akan lebih terasa jika digunakan *whole database encryption.* Selanjutnya, data yang perlu diproteksi masih rawan diserang saat berpindah dari satu sistem ke sistem lain karena di luar basis data data tersebut berada dalam bentuk plainteks. Penanganan lebih lanjut akan diperlukan untuk mengamankan transfer data di luar basis data.

Dengan enkripsi data secara internal, pada umumnya kunci akan disimpan di dalam tabel di dalam basis data yang sama. Artinya, data yang diproteksi dengan enkripsi tidak terpisah dengan kunci yang digunakan untuk mengenkripsi dan mendekripsinya. Meskipun kunci umumnya disimpan dalam tabel dengan akses terbatas, hal ini tentu saja akan meningkatkan resiko keamanan data karena setiap orang yang dapat mengakses basis data juga dapat mengakses kunci.

Di luar beberapa kekurangan yang disebutkan di atas, enkripsi basis data secara internal masih menjadi pilihan karena kemudahan implementasinya. Saat ini telah banyak DBMS komersial yang telah menyediakan fitur enkripsi. Untuk DBMS yang belum mendukung fitur ini, telah tersedia pula produk-produk *add-on* yang dapat diintegrasikan dengan DBMS untuk menambahkan fitur tersebut pada DBMS yang bersangkutan. Dengan menggunakan DBMS dan produk-produk tersebut, perusahaan tidak perlu lagi melakukan

modifikasi pada aplikasi-aplikasi yang mengakses data maupun pada arsitektur sistem secara keseluruhan.

Untuk memilih DBMS atau produk enkripsi *add-on*, perlu dipastikan bahwa algoritma enkripsi yang digunakan adalah algoritma yang telah terbukti aman dan disediakan oleh penyedia layanan kriptografi yang terpercaya.

2. Enkripsi secara eksternal di luar basis data

Strategi penyimpanan data yang lebih aman adalah dengan menambahkan fungsi enkripsi pada aplikasi. Enkripsi dilakukan di dalam aplikasi sehingga data dapat ditransfer dan disimpan dalam bentuk terenkripsi. Pendekatan ini menyediakan pengamanan *end-to-end* yang baik, namun membutuhkan perubahan pada aplikasi yaitu dengan menambahkan atau memodifikasi fungsi enkripsi dan dekripsi.

Salah satu langkah efektif untuk mengimplementasikan strategi ini adalah dengan membangun server enkripsi yang menyediakan layanan enkripsi secara terpusat (*centralized encryption service*) untuk seluruh *environment* basis data. Cara ini dapat menyederhanakan proses manajemen dan meningkatkan kontrol terhadap *environment* multi-aplikasi yang menggunakan banyak basis data. Server enkripsi dapat dioptimalkan untuk menjalankan operasi kriptografi yang diminta oleh aplikasi. Server menjadi basis fungsi enkripsi yang dapat dipanggil oleh setiap aplikasi pada sistem.

Kelebihan utama dari pengimplementasian strategi ini adalah memberikan sistem pengamanan kunci yang terbaik. Data yang telah dienkripsi dimasukan ke dalam basis data, sedangkan kuncinya tetap berada pada server enkripsi. Hal ini meningkatkan proteksi pada basis data.

Pengimplementasian strategi ini tentunya membutuhkan sistem pengamanan yang ketat terhadap aplikasi dan server enkripsi. Solusinya adalah dengan menerapkan sistem otentikasi sehingga hanya *user* yang memiliki otoritas saja yang dapat mendekripsi data sensitif dengan mengakses kunci yang disimpan dalam server enkripsi. Solusi kedua adalah dengan meningkatkan sensitifitas dari server enkripsi dengan melakukan *monitoring* terhadap aktivitas user yang mencurigakan dan mengaudit log kejadian secara reguler.

Kelebihan lain yang didapat dari strategi ini adalah peningkatan performansi karena server basis data (DBMS) tidak dibebani dengan pemrosesan kriptografi (fungsi enkripsi). Strategi ini juga memungkinkan kemampuan untuk menambah fungsi enkripsi sesuai kebutuhan.

Pembangunan server enkripsi dan modifikasi aplikasi merupakan kerja yang berat dan juga membutuhkan biaya yang cukup besar. Namun, strategi ini memberikan sistem pengamanan dan performansi yang lebih baik. [7)]

3.2.1 Persoalan Umum dalam Desain Pengamanan Basis Data

Hal-hal penting yang harus dipertimbangkan dalam mendesain sistem pengamanan basis data adalah pemilihan algoritma kriptografi, manajemen kunci, strategi otentikasi, dan pemilihan format penyimpanan terhadap data yang dienkripsi.

1. Pemilihan algoritma kriptografi

Untuk memberikan tingkat pengamanan terbaik, data sensitif harus disimpan dalam bentuk terenkripsi. Hal ini dapat diperoleh dengan menggunakan algoritma enkripsi yang teruji dan mendukung sistem industri. Dalam pemilihan algoritma enkripsi ada tiga hal yang harus dipertimbangkan. Pertama, tipe enkripsi yang terbaik sesuai data yang dimiliki. Kedua, pemilihan data yang perlu dienkripsi. Ketiga, kekuatan algoritma enkripsi yang akan diimplementasikan.

Secara umum didefinisikan tiga kategori dalam algoritma enkripsi, yang pertama adalah algoritma simetri (S*ymmetric-key Ciphers*). Algoritma ini menggunakan kunci yang sama untuk enkripsi dan dekripsi data, umumnya digunakan untuk menyimpan data pada *storage*. Terdapat dua tipe dari algoritma simetri yaitu *block cipher* dan *stream cipher*. *Stream cipher* memiliki kecepatan dua kali lipat dibandingkan dengan *block cipher*, namun tipe ini membutuhkan kunci yang unik. Sedangkan pada *block cipher*, kunci dapat dipergunakan kembali. Sebagian besar DBMS memasukan fungsi enkripsi yang menggunakan bentuk dari teknologi *block cipher*.

Beberapa algoritma simetri yang umum digunakan untuk enkripsi basis data adalah AES, DES, 3DES, dan RC5. Seluruhnya bertipe *block-cipher*. Adapun algoritma RC4 yang merupakan tipe *stream cipher* dapat pula digunakan, namun setiap kali data yang masuk dienkripsi, sebuah kunci yang unik harus disediakan sehingga manajemen kunci menjadi semakin kompleks.

Kategori algoritma enkripsi yang kedua adalah algoritma kunci publik (*Asymmetric-key Ciphers*). Algoritma ini memiliki dua kunci yaitu kunci publik dan kunci privat, umumnya digunakan untuk mengamankan data yang akan melalui proses transmisi. Algoritma ini memiliki performansi yang lebih lambat dibandingkan algoritma simetri sehingga tidak cocok digunakan untuk enkripsi pada basis data.

Kategori yang terakhir adalah algoritma Hashing. Algoritma hashing menghasilkan nilai hash dengan panjang byte yang tetap dan bernilai unik sesuai datanya. Algoritma ini tidak digunakan untuk mengenkripsi data melainkan untuk menjamin integritas data. [7)]

2. Manajemen kunci

Selain kualitas algoritma enkripsi, manajemen kunci juga merupakan hal yang sangat penting untuk diperhatikan dalam pengamanan basis data. Sedikitnya ada empat hal yang harus dipertimbangkan dalam manajemen kunci yaitu jumlah kunci, tempat penyimpanan kunci, perlindungan akses terhadap kunci, dan periode perubahan kunci.

Dalam mempertimbangkan jumlah kunci perlu diingat bahwa semakin sedikit kunci yang digunakan untuk mengenkripsi informasi maka semakin mudah solusi yang dikelola, namun sistem pengamanan akan semakin kritis.

Mengenai tempat penyimpanan kunci, terdapat dua strategi pengamanan yaitu dalam basis data, dimana dengan kebijakan ini harus ada kepercayaan pada DBA (*based on honour code*), dan yang kedua adalah dalam hardware (*hardware storage modul*). Strategi kedua memisahkan kunci dari data yang dienkripsinya dengan menyimpan kunci tersebut dalam hardware. Kelebihan strategi ini adalah data akan tetap berada pada *hardware device*. Kondisi

ini memungkinkan kontrol terhadap pengaksesan data akan lebih ketat sehingga baik administrator maupun penyerang tidak dapat mengambil datanya.

Perlindungan akses terhadap kunci dapat dilakukan dengan menerapkan strategi otentikasi terhadap setiap user yang akan mengakses data. Strategi otentikasi ini akan dibahas pada bagian selanjutnya.

Penentuan periode perubahan kunci juga merupakan hal yang penting dalam strategi pengamanan data. Hal ini penting karena dimungkinkan terdapat pihak-pihak yang berusaha membangkitkan kunci dengan menggunakan algoritma tertentu (kriptanalis). Dengan perubahan kunci ini diharapkan penyerang akan mengalami kesulitan dalam membangkitkan kunci yang bersesuaian.

3. Strategi Otentikasi

Untuk mengontrol akses terhadap kunci enkripsi maka diperlukan strategi dalam otentikasi. Terdapat beberapa metode otentikasi yang dapat digunakan untuk mengamankan kunci.

Metode yang paling umum digunakan adalah *password based.* Metode ini memungkinkan kunci enkripsi disimpan tidak dalam bentuk *clear text*, sehingga untuk dapat membaca kunci enkripsi, user harus mengetahui *password*. Dengan metode ini, semakin banyak *password* yang digunakan sistem akan semakin kompleks untuk dikelola.

Metode kedua adalah *smart cards. Smart cards* menggunakan dua faktor untuk mengotentikasi yaitu sesuatu yang dimiliki oleh user (*smart card*) dan sesuatu yang diketahui oleh *user* (kode PIN atau *password*). *Smart card* digunakan untuk sistem yang membutuhkan banyak *password* karena *smart card* memiliki kemampuan untuk menyimpan kombinasi *username* dan *password*.

Biometrics merupakan metode berteknologi tinggi yang digunakan dalam melakukan otentikasi. Sistem *Biometrics* menggunakan karakteristik fisik tubuh untuk melakukan otentikasi. Sistem ini dapat dikombinasikan dengan password dan smart card untuk meningkatkan keamanan dalam otentikasi. [7)]

Teknik kriptografi yang banyak digunakan untuk otentikasi adalah *Public Key Infrastructure* (PKI). PKI memungkinkan user dari jaringan yang tidak aman seperti internet secara aman melakukan pertukaran data dengan menggunakan pasangan kunci publik dan privat yang disediakan melalui otoritas terpercaya. [6)]

Metode selanjutnya adalah Kerberos. Kerberos merupakan protokol otentikasi pada jaringan yang didesain untuk mengotentikasi aplikasi *client/server* dengan menggunakan kunci privat. Protokol Kerberos menggunakan algoritma kriptografi yang tangguh sedemikian sehingga client dapat menunjukan identitasnya kepada server (juga sebaliknya) melalui koneksi jaringan yang tidak aman sekalipun. Proses enkripsi terhadap keseluruhan data dapat dilakukan setelah client dan server saling percaya terhadap identitas lawan komunikasinya. [3)]

4. Pemilihan format penyimpanan data yang terenkripsi

Kode aplikasi dan skema basis data sangat mempengaruhi perubahan tipe dan

panjang data. Untuk itu perlu ditentukan format penyimpanan enkripsi data yang memudahkan pengelolaan. [8)]

Untuk mengakses data dalam basis data yang telah terenkripsi tanpa mengganggu performansi dan melibatkan terlalu banyak data lain yang sensitif, setiap bagian dari data yang sensitif dapat dikenakan HMAC (*Keyed-Hash Message Authentication Code*). Nilai HMAC kemudian disimpan pada kolom yang lain pada baris yang sama. HMAC merupakan algoritma untuk mengotentikasi pesan yang menggunakan kunci kriptografi yang digabungkan dengan fungsi hash [1)].

Dalam mengenkripsi seluruh tabel dalam database, DBA harus terlebih dahulu men-*drop primary key* dan juga *reference key* yang terdefinisi untuk kemudian dibentuk kembali setelah seluruh data terenkripsi. Hal ini dilakukan karena pada umumnya *primary key* merupakan indeks dari suatu tabel, sehingga enkripsi yang dilakukan pada kolom *primary key* akan secara signifikan menurunkan performansi sistem. Dengan kondisi seperti ini enkripsi terhadap kolom *primary key* sangat tidak dianjurkan. Hal yang sama juga berlaku pada kolom yang merupakan indeks atau bagian dari indeks.

Jika proses enkripsi dilakukan dengan menggunakan metode CBC, *Initialization Vector* (IV) yang diciptakan secara random harus disimpan untuk kemudian digunakan ketika melakukan dekripsi. IV dapat disimpan di dalam basis data karena bukan merupakan hal yang harus dijaga kerahasiaannya. Cara menyimpan IV dapat disesuaikan dengan aplikasinya. Jika aplikasi membutuhkan IV untuk setiap kolom maka IV disimpan di dalam tabel terpisah. Untuk menjaga keamanan ketika melakukan *deployment*, IV dapat disimpan setiap baris bersama datanya [8)].

4. Kesimpulan dan Saran Pengembangan

Teknik kriptografi dapat diterapkan untuk mengamankan basis data dari serangan pihak luar maupun pihak internal perusahaan. Terdapat beberapa langkah dalam mengimplementasikan pengamanan basis data menggunakan teknik kriptografi. Pertama adalah analisis lingkungan yang mencakup identifikasi data sensitif, aliran data perusahaan, identifikasi pengguna, dan identifikasi terhadap ancaman yang potensial terhadap basis data. Langkah kedua adalah pemilihan desain solusi, yaitu enkripsi di dalam basis data atau di luar basis data. Langkah terakhir adalah mengimplementasikan desain solusi yang dipilih berdasarkan analisis lingkungan yang diperoleh.

Beberapa persoalan terkait desain pengamanan basis data adalah pemilihan algoritma kriptografi, manajemen kunci, strategi otentikasi, dan pemilihan format penyimpanan data yang dienkripsi.

Saran pengembangan terhadap makalah ini adalah penambahan cakupan implementasi pengamanan basis data terhadap proses bisnis yang berkaitan dengan pertukaran data antara konsumen dan perusahaan.

Referensi

[1] Federal Information Processing Standards Publication. *The Keyed-Hash Message Authentication Code*. http://csrc.nist.gov/publications/fips/fips198/fips-198a.pdf, diakses tanggal 26 Desember 2005

[2] http://www.hipaa.org/, diakses tanggal 9 Desember 2005.

[3] Kerberos: The Network Authentication Protocol. http://web.mit.edu/kerberos/, diakses tanggal 26 Desember 2005

[4] K.Kenan, *Criptography in the Database : The Last Line of Defense*, Addison Wesley Proffesional, 2005.

[5] NetLib Encryptionizer, *Difference Between Whole Database and Column Encryption*, NetLib Encryptionizer Product Overview, http://www.netlib.com/column-vs-database.shtml, diakses tanggal 10 Desember 2005.

[6] PKI. http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci214299,00.html, diakses tanggal 26 Desember 2005

[7] RSA Security, *Securing Data at Rest : Developing a Database Encryption Strategy, A White Paper for Developers, e-Bussiness Managers and IT*, http://www.rsasecurity.com/products/bsafe/whitepapers/DDES_WP_0702.pdf, diakses tanggal 9 Desember 2005.

[8] U.T. Mattsson, CTO Protegrity Corp, *Transparent Encryption and Separation of Duties for Enterprise Databases, A Solution for Field Level Privacy in Database*, http://www.quest-pipelines.com/newsletter-v6/0105_A.htm, diakses tanggal 12 Desember 2005.