

Proses Replikasi Data Enkripsi Antara Server Utama dan Firewall

Kontribusi: Irvanizam Zamanhuri
Sunday, 17 September 2006

* Staf Jurusan Matematika, FMIPA, Universitas Syiah Kuala, Mahasiswa S2 di Computer Science, Fakultät für Informatik - Freie Universität Bozen, Italia. Perkembangan teknologi komputer dan sistem informasi mengakibatkan makin besarnya masalah keamanan data. Komputer tidak menciptakan masalah itu sendiri, tetapi keuntungannya yang efektif memperluas jangkauan pengumpulan informasi. Makin banyak jumlah informasi yang dikumpulkan, terutama informasi yang bernilai tinggi, makin banyak pula pihak tertentu yang ingin menarik keuntungan untuk memiliki/mengetahuinya. Kriptografi adalah salah satu teknik untuk menyelenggarakan sistem rahasia, sedangkan ilmu yang mempelajari tentang sistem rahasia disebut Kriptologi. Kriptografi digunakan untuk mengubah data ke dalam bentuk kode-kode tertentu, dengan tujuan data yang disimpan maupun yang ditransmisikan melalui jaringan, tidak dapat dibaca oleh siapapun kecuali oleh orang-orang yang berhak. Tulisan ini mendiskripsikan proses replikasi dan metode pengamanan data dengan algoritma enkripsi antara server perantara (firewall) dan server utama serta mengimplementasikan dengan bahasa pemrograman Pascal dan Assembly. Proses enkripsi dan dekripsi data diimplementasikan dengan menggunakan bahasa pemrograman assembly. Adapun teknik manipulasi data bit digunakan untuk mengimplementasikan algoritma enkripsi, dekripsi, dan checksum. Komposisi metode ini membentuk produk cipher heksadesimal dan dikirimkan melalui jaringan. Algoritma enkripsi dan checksum mengkonversikan masukan 64 bit yang disebut plaintext menjadi keluaran 64 bit yang disebut ciphertext dengan kunci tertentu. Proses replikasi data melibatkan checksum yang berfungsi sebagai pengontrol, pengecek kevalidan, dan pendeteksi data yang termanipulasi yang dilakukan oleh controller yang bereplikasi setiap detiknya. Sebagai pelayanan dibuat realisasi proses replikasi data antara dua buah server basis data dan algoritma enkripsi dalam bentuk perangkat lunak.

Teknik Manipulasi Data Bit Manipulasi data bit adalah pengolahan data dengan operasi logika tertentu untuk mengubah suatu bentuk data byte ke data byte yang lain. Operasi logika yang dilakukan oleh mikroprosesor adalah mengeset bit (bit set), mereset bit (bit reset), menginversi bit (bit invers), menggeser bit (bit shift), dan membandingkan byte (byte compare). Teknik untuk melakukan manipulasi adalah teknik bit masking yaitu suatu teknik menganalisa bit-bit tertentu dengan satu bit penyaring (masker byte). Byte penyaring adalah satu byte patokan untuk memeriksa atau menyaring byte data.

Mengeset Bit Operasi mengeset bit (bit set) adalah operasi mengubah bit-bit tertentu sehingga bernilai satu apapun kondisi sebelumnya, "0" atau "1". Operator logika yang dipakai adalah "OR". Operasi OR menghasilkan nol jika kedua operan nol dan akan menghasilkan satu jika salah satu operan adalah satu. Sesuai dengan persamaan aljabar Boole "A+1=1" maka bit masker untuk mengeset bit adalah "1".

Tabel 1. Operasi OR

B	A	OR	B	0
0	0	0	1	1
0	1	1	0	1
1	0	1	1	1
1	1	1	1	1

Mereset Bit Operasi mereset bit (bit resetting) adalah operasi bernilai nol apapun kondisi sebelumnya, "0" atau "1". Operator logika yang dipakai adalah "AND". Operasi AND menghasilkan satu jika kedua operan satu dan akan menghasilkan nol jika salah satu operan adalah nol. Sesuai dengan persamaan aljabar Boole "A * 0 = 0" maka bit masker untuk mereset bit adalah nol. **Tabel 2 Operasi AND**

Tabel 2 Operasi AND

B	A	AND	B	0
0	0	0	1	0
0	1	0	0	0
1	0	0	1	1
1	1	1	1	1

Menginversi Bit Operasi menginversi bit (bit invers) adalah operasi membalik nilai bit-bit tertentu sehingga mempunyai nilai yang berlawanan dengan nilai sebelumnya. Logika yang dipakai adalah "XOR". Jika XOR menghasilkan nol jika nilai kedua operan sama dan akan menghasilkan satu jika nilai kedua operan berbeda. Sesuai dengan persamaan Boole dari fungsi logika XOR "Q=A B = A*~B+~A*B" maka bit masker untuk menginversi bit adalah satu.

Penjumlahan modulo k (k adalah bilangan bulat) mempunyai rumus umum sebagai berikut:

Sehingga operasi modulo 2 dapat ditulis sebagai berikut: $0 + 0 = 0 + 0 = 0$ $0 + 1 = 0 + 1 = 1$ $1 + 0 = 1 + 0 = 1$ $1 + 1 = (1+1) - 2 = 2 - 2 = 0$

Tabel 2.3 Operasi XOR

B	A	XOR	B	0
0	0	0	1	1
0	1	1	0	1
1	0	1	1	0
1	1	0	1	1

Menggeser Bit Operasi menggeser bit (bit shifting) adalah operasi suatu bit-bit berurutan ke posisi sebelum atau sesudahnya.

Proses Replikasi Data. Keamanan merupakan faktor yang sangat penting dan harus ada di dalam sistem informasi. Proses replikasi tidak menjamin keamanan data di dalam server basis data. Hal ini diperlukan adanya firewall untuk memfilter data yang keluar/masuk dari sistem dalam upaya menghindari penduplikasian dan perusakan data. Proses replikasi data akademik mempunyai dua langkah. Langkah pertama, mereplikasikan data-data ciphertext (data yg tidak dapat dibaca) dari basis data firewall ke pengguna sistem (user interface) melalui proses dekripsi datanya. Langkah kedua, mereplikasikan data-data plaintext (data yang dapat dibaca) dari user interface ke basis data firewall dan server utama, melalui proses enkripsi data. Tahapan proses replikasi data dijelaskan melalui gambar.

Algoritma Enkripsi
Metode enkripsi adalah suatu metode manipulasi data dengan mengkodekan/ menyembunyikan data aslinya, sehingga data yang bisa dibaca dan dimengerti oleh siapapun (plaintext/cleartext) menjadi data yang tidak bisa dibaca dan dimengerti dengan jelas. Perancangan program algoritma enkripsi dirancang dengan memperhatikan kerumitan pemecahan untuk mendeteksi alur programnya. Paket data masuk harus bertipe string. Jika data yang diinputkan bertipe integer atau date, maka data tersebut dikonversikan menjadi data bertipe string. Pemilihan kata kunci sangat dibutuhkan pada implementasi algoritma enkripsi. Kunci yang rumit akan menjadikan algoritma enkripsi yang handal. Proses enkripsi data dilakukan melalui perkarakturnya. Setiap karakter yang telah terenkripsi dan dilakukan manipulasi data bitnya lagi

menjadi kata kunci selanjutnya. Kata kunci yang baru akan mendukung proses enkripsi karakter selanjutnya. Berikut potongan program algoritma enkripsi dengan menggunakan metode manipulasi data bit.

```
asm push eax mov al,key1
xor mxor,al //melakukan xor dengan key1 pop eax push eax
ror mxor,3 //melakukan geser ke kanan 3 x push
mov al,key2 xor mxor,al //melakukan xor dengan key2 pop eax
push eax pop eax rol mxor,2 //melaku
ke kiri 2 x ror mxor,3 //melakukan geser ke kiri 3 x
add mxor,1 //Menambahkan 1 xor mxor,3 //melakukan xor de
3 end;
```

Listing 1. Program Manipulasi Data Bit Pada Algoritma Enkripsi

Berikut program algoritma pembangkit key dengan menggunakan metode manipulasi data bit.

```
procedure createKey (mxor,m:byte;count:integer):string;
Var Key1,key2:byte;
Begin Key1:=m+6*mxor+47; Key2:=mxor+3*m+55;
{melakukan operasi biner terhadap key1 & key2
untuk karakter ke 2 dst...}
if count mod 2 =0 then
asm ror key1,3 //geser key1 ke kanan 3x
push ebx m
//memindahkan hasil rotate ke register bl
xor key2,2 //xor key2 gengan 2
push eax mov al,key2 mov key1
//memindahkan nilai register al ke key1
mov key2,bl //memindahkan nilai tegister bl ke key2
pop eax pop ebx
else
asm rol key1,3 //geser key1 ke kiri 3x
xor key2,3 //xor key2 gengan 3
end; end;
```

Listing 2. Program Ma

Data Bit Pada Algoritma Pembangkit Key

Algoritma enkripsi pada tulisan ini adalah mengenkripsikan data plaintext per karakter dengan melibatkan perubahan key dari hasil data karakter enkripsi sebelumnya. Hasil enkripsi dalam bentuk data heksadesimal, yang dikonversikan dari data bertipe byte dari hasil teknik manipulasi data bit.

```
function encrypt(mfield :string);
string; var mtext:string; ret_str:string;
has:array[1..100] of byte; count,max_text :integer;
m,mxor,key,key1 :byte;
begin mtext:=Trim(mfield);
max_text:=length(mtext); count:=1; key1:=3; key2:=15;
while count <= max_text do
begin m:=ord(mtext[count]);
{ untuk menampung nilai mxor dalam assembler}
mxor:=m; // Listing program 1
ditampilkan disini { mxor menjadi variabel tambahan untuk key1 }
has[count]:=mxor; createKey(mxor,m,count);
// function pemanggil pembangkit key
inc(count); end;
ret_str:=""; count:=1;
{ prosedur mengkonversikan data byte ke heksadesimal }
while count <= max_text do
begin m:=has[count];
ret_str:=ret_str+inttohex(m,2);
inc(count); end;
encrypt:=ret_str; end;
```

Listing 3. Program Algoritma Enkripsi

Algoritma Dekripsi

Algoritma dekripsi merupakan inversi dari algoritma enkripsi. Jika pada algoritma enkripsi dipakai manipulasi data bit dengan metode menggeserkan semua bit sebanyak dua kali ke kiri, maka proses dekripsi harus menggunakan manipulasi data bit dengan metode menggeserkan bit sebanyak dua kali ke kanan. Parameter yang diparsingkan ke dalam algoritma dekripsi bertipe string. Data masukkan adalah data dalam bentuk string dalam format heksadesimal. Maka langkah pertama yang dilakukan algoritma dekripsi adalah mengkonversikan kembali data format heksadesimal kedalam bentuk string. Dua buah karakter data enkripsi akan menjadi satu data string.

```
function hexadesimalToStr (data:string):string;
Var mhigh_h,mlow_h:byte;
j,maxLengthData,mhigh,mlow:integer;
dataStr:string;
begin maxLengthData:=length(data) div 2;
dataStr:=""; j:=0;
while j < maxLengthData do
begin mhigh:=data_encrypt[2*j+1];
case mhigh of 'A': mhigh_h:=160; 'B': mhigh_h:=176;
'C': mhigh_h:=192; 'D': mhigh_h:=208; 'E': mhigh_h:=224; 'F': mhigh_h:=240;
else val(mhigh,mhigh_h,code);
mhigh_h:=mhigh_h*16; end;
mlow:=data_encrypt[2*j+2];
case mlow of 'A': mlow_h:=10; 'B': mlow_h:=11; 'C':
mlow_h:=12; 'D': mlow_h:=13; 'E': mlow_h:=14; 'F': mlow_h:=15;
else val(mlow,mlow_h,code); end;
dataStr:=dataStr+chr(mhigh_h+mlow_h);
inc(j); end;
```

Listing 4. Program Konversi Data

Heksadesimal ke Data String

```
end;
```

Berikut program manipulasi data bit untuk algoritma dekripsi.

```
asm xor mxor,3 //melakukan xor dengan 3
dec mxor //
Mengurangkan dengan 1
rol mxor,3 //melakukan geser ke kiri 3 x
rol mxor,2 //melakukan geser ke kanan 2 x
push
eax mov al,key2 xor mxor,al //melakukan xor dengan key2
pop eax push eax pop eax rol mxor,3 //m
geser ke kiri 3 x
push eax mov al,key1 xor mxor,al //melakukan xor dengan key1
pop eax push eax e
```

5. Program Manipulasi Data Bit Pada Algoritma Dekripsi

Prosedur pembangkitan key untuk algoritma dekripsi adalah sama halnya pada algoritma enkripsi. Pemanggilan fungsi key dengan menampilkan prototype `createKey(mxor,m:byte;count:integer)` dalam prosedur dekripsi. Fungsi `createKey` mempunyai tiga parameter masing-masing parameter hasil proses manipulasi data bit, nilai konversi karakter dekripsi ke tipe byte, dan urutan karakter enkripsinya.

```
function decrypt(mfield :string): string;
var mtext:string; ret_str:string;
has:array[1..100] of byte; count,max_text :integer;
m,mxor,key,key1 :byte;
begin hexadesimalToStr(mfield);
max_text:=length(hexadesimalToStr(mfield));
count:=1; key1:=3; key2:=15;
while count <= max_text do
begin charDecrypt:=ord(mtext[count]);
{ untuk menampung nilai mxor dalam assembler}
mxor:=charDecrypt; // Listing
program 4
ditampilkan disini decryptText:= decryptText+chr(charDecrypt);
{ function pemanggil pembangkit key }
createKey(mxor,charDecrypt,count);
inc(count); end;
decrypt:= decryptText; end;
```

Hasil Enkripsi

Setelah salah satu data KRS dikirimkan, semua data tersebut menjadi data enkripsi dalam tabel `insert_nilai` di firewall. Algoritma mengeluarkan data enkripsi yang sulit dimengerti oleh pengguna sistem. Hasil enkripsi data diperlihatkan pada tabel berikut.

Tabel 1. Hasil Enkripsi Salah Satu Data KRS

Data Plaintext

Chipertext

Kode Matakuliah

MMT402EDC36C114BD2

Nilai A

5F

NIM

98811102550319CF

FAF8AEA458DE25B

Kelas

Jumlah SKS

6A6

Bobot Nilai

066

Status

Semester

2001126A637EE8C

No. Pengguna

User

106

Hasil Chechsum

5416F92604A477507BB2

Penutup

Hasil enkripsi data menunjukkan bahwa pengkodean data menjadi data enkripsi dapat menghindari pemanipulasian dan perusakan data melalui jaringan dalam upaya melindungi basis data server utama. Selain itu, Teknik manipulasi data bit bisa juga digunakan untuk mengimplementasikan algoritma enkripsi, dekripsi, dan checksum.

Referensi Bruce Schneier, Applied Cryptography – Second Edition, John Wiley & Sons, Inc, 1996

Budi Rahardjo.Keamanan Sistem Informasi Berbasis Internet, PTInsanIndonesia, 1998
Illik W., Elang, Keamanan Data 2 (Mikrodata Pilihan), PT. Elex Media Komputindo, Jakarta, 1995. Setiawan, G.C,
Mikro Komputer LS-Z80 Sebagai Pengendali Model Robot. Skripsi, Yokyakarta, 2000.