Studi Algoritma Stream Chiper HC-256

Simon H S – NIM: 13504056 [1)]

1) Program Studi Informatika ITB, Bandung 40135, email: if14056@students.if.itb.ac.id

Abstract *– Makalah ini membahas mengenai studi terhadap algoritma chiper aliran terbaru yang di-claim memiliki kecepatan yang sangat tinggi dan ketahanan terhadap serangan yang cukup besar.*

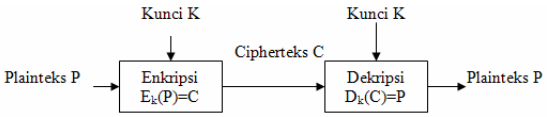
Algoritma ini menghasilkan keystream dari 256 bit initialization vector dan 256 bit key. Kedua hal ini kemudian diproses dan digabungkan di dalam 2 buah tabel berukuran 1024 elemen dengan masing-masing elemennya berukuran 32 bit.

Dalam makalah ini akan dibahas mengenai algoritma chiper aliran HC-256 secara mendalam, mencakup variable, fungsi dan operasi yang digunakan. Cara kerja algoritma HC-256, cara enkripsi dan dekripsi, analisis keamanan algoritma HC-256 terhadap beberapa serangan, serta penghitungan performansi algoritma tersebut juga akan dibahas.

Kata Kunci*: Chiper aliran, HC-256,*

1. PENDAHULUAN

Pengiriman dan penyimpanan pesan melalui media elektronik dewasa ini sudah sangat sering dilakukan. Terkadang pengiriman dan penyimpanan pesan melalui media elektronik perlu dirahasiakan untuk menjamin keamanan dan keutuhan data yang dikirimkan. Oleh sebab itu maka dibutuhkan metode penyandian pesan. Ilmu sekaligus seni untuk menyandikan pesan disebut juga sebagai kriptografi.



Gambar 1 Skema Enkripsi dan Dekripsi

Dilihat dari kunci yang digunakan, kriptografi dapat dibagi menjadi 2 jenis. Jika kunci enkripsi sama dengan kunci dekripsi, maka kriptografinya dinamakan dengan kriptografi kunci-simetri atau algoritma konvensional. Sedangkan bila kunci yang digunakan pada enkripsi berbeda dengan yang digunakan pada dekripsi, maka disebut kriptografi kunci-nirsimetri.

Kriptografi kunci-simetri juga terbagi atas dua jenis yaitu chiper blok (*block chiper*) dan chiper aliran (*stream chiper*). Kriptografi chiper blok beroperasi pada plainteks/chiperteks dalam bentuk blok bit, yang dalam hal ini rangkaian bit yang panjangnya telah ditentukan sebelumnya. Chiper aliran sebaliknya merupakan algoritma kriptografi yang beroperasi pada plainteks/chiperteks dalam bentuk bit tunggal, atau dengan kata lain rangkaian bit dienkripsikan/ didekripsikan secara bit per bit.

Ada banyak contoh algoritma chiper aliran yang telah dipublikasikan, salah satu yang terkenal dan paling banyak digunakan adalah RC4. Algoritma chiper aliran pada perangkat lunak modern umumnya dapat bekerja 4-5 kali lebih cepat daripada algoritma chiper blok. Namun sangat sedikit algoritma chiper aliran yang efisien dan aman yang telah dipublikasikan[2].

Untuk itu, dalam makalah ini akan dibahas mengenai salah satu algoritma chiper aliran baru yaitu HC-256. Algoritma ini merupakan algortima yang dibuat untuk menyediakan algoritma chiper aliran yang efisien dan aman.

2. Chiper Aliran HC-256[2]

HC-256 merupakan algoritma chiper aliran yang *software-efficient*. Algoritma ini membangkitkan bit-bit kunci (keystream) dari 256 bit kunci rahasia dan 256 bit *Initialitation Vector*. HC-256 juga mempunyai 2 buah tabel rahasia, yang masing-masingnya mempunyai 1024 elemen, dengan masing-masing elemen berukuran 32 bit.

2.1. Variabel, Fungsi dan Operasi

Algoritma HC-256 menggunakan operator sebagai berikut:

a. + : $x + y$ berarti $x + y \bmod 2^{32}$, dimana $0 \le x \le 2^{32}$
b. – : $x – y$ berarti $x – y \bmod 1024$
c. ^ : $x \wedge y$ berarti XOR
d. || : $x \,||\, y$ berarti konkatenansi
e. >> : $x >> n$ berarti x digeser ke kanan sejauh n
f. << : $x << n$ berarti x digeser ke kiri sejauh n
g. >>> : $x >>> n$ berarti $((x >> n) \wedge (x << (32 - n)))$ dimana $0 \le n < 32$, $0 \le x < 2^{32}$

Pada HC-256 digunakan 2 buah tabel. Notasi-notasi yang digunakan dalam pemrosesan tabel adalah :

a. P : tabel dengan 1024 32-bit elemen. Tiap elemen digambarkan dengan $P[i]$ dengan $0 \le i \le 1024$
b. Q : tabel dengan 1024 32-bit elemen. Tiap elemen digambarkan dengan $Q[i]$ dengan $0 \le i \le 1024$
c. K : kunci 256 bit yang digunakan HC-256
d. IV : Initialisation Vector 256 bit dalam HC256
e. s : keystream total yang dihasilkan HC256. 32 bit keystream yang dihasilkan pada langkah ke i disebut s_i.

Dalam HC-256 digunakan 6 buah fungsi yaitu :

a. $f1(x) = (x >>> 7) \wedge (x >>> 18) \wedge (x >> 3)$
b. $f2(x) = (x >>> 17) \wedge (x >>> 19) \wedge (x >> 10)$
c. $g1(x; y) = ((x >>> 10) \wedge (y >>> 23)) + Q[(x \wedge y) \bmod 1024]$
d. $g2(x; y) = ((x >>> 10) \wedge (y >>> 23)) + P[(x \wedge y) \bmod 1024]$
e. $h1(x) = Q[x0] + Q[256 + x1] + Q[512 + x2] + Q[768 + x3]$
f. $h2(x) = P[x0] + P[256 + x1] + P[512 + x2] + P[768 + x3]$

dimana $x = x_3||x_2||x_1||x_0$.x adalah karakter 32 bit dengan x_0,x_1,x_2 dan x_3 masing 4 bit. x_0 dan x_3 merupakan MSB dan LSB dari x.

2.2. Inisialiasi HC-256 (*IV* dan *key*)

Dalam proses inisialiasi, yang pertama kali dilakukan adalah mengekspan *key* dan *IV* ke dalam *P* & *Q*, dan melakukan pembangkitan keystream sebanyak 4096 kali. Secara detail dapat dilihat:

i. Misalnya $K = K_1||\ldots||K_7$ dan $IV = IV_1||\ldots||IV_7$, dimana K_i dan IV_i adalah bilangan 32 bit. K dan IV ini kemudian dimasukkan ke dalam array $W_i(0 \le i \le 2559)$ sebagai berikut :

$W_{0..7} = K_{1..7}$
$W_{8..16} = IV_{1..8}$
$W_{16..2559} = f2(Wi_{i}2) + Wi_{i}7 + f1(Wi_{i}15) + Wi_{i}16 + i$

ii. Update tabel P dan Q dengan array W sebagai berikut:

$P_{[i]} = W_i + 512$ untuk $0 \le i \le 1023$
$Q_{[i]} = W_i + 1536$ untuk $0 \le i \le 1023$

iii. Bangkitkan keystream 4096 kali tanpa menghasilkan keluaran.

2.3. Algoritma pembangkitkan keystream

Algoritma ini bertugas untuk menghasilkan keystream dimana pada tiap langkahnya, 1 buah elemen dari tabel *P* atau *Q* akan diupdate nilainya, dan akan membangkitkan 1 buah elemen 32 bit sebagai keystream keluaran.

Algoritma pembangkitan keystream ini dapat dilihat sebagai berikut:

```
procedure keystreamGen
{
i = 0;
repeat until enough keystream bits are
generated
{
j = i mod 1024;
if (i mod 2048) < 1024{
P[j] = P[j]+P[j-10] + g1(P[j-3],P[j-1023]);
si = h1(P[j-12]) ^ P[j];
}
else{
Q[j] = Q[j]+Q[j-10] + g2(Q[j-3],Q[j-1023] );
si = h2(Q[j - 12] ) ^ Q[j];
}
end-if
i = i + 1;
}
end-repeat
}
```

2.3. Enkripsi dan Dekripsi

Untuk enkripsi, dapat dilakukan dengan melakukan XOR antara pesan yang ingin dienkripsi dengan keystream yang telah dihasilkan.

Untuk dekripsi, dilakukan dengan melakukan XOR antara chipertext yang ingin didekripsi dengan keystream yang dihasilkan.

3. Analisis Keamanan HC-256

Chiper aliran biasanya dirancang dengan menggunakan teknik Linear Feedback Shift Registers (LFSR) atau register geser untuk membangkitkan aliran bit kunci acak[1]. Namun, algoritma yang dirancang berdasarkan LFSR bisa dikatakan cukup rentan. Telah banyak teknik-teknik kriptanalis yang telah dipublikasikan yang disebut mampu memecahkan chiper aliran yang berdasarkan pada LFSR tersebut. Teknik korelasi antar chipertext, kriptanalisis linier, dan distinguishing attack merupakan contoh-contoh kriptanalis pada chiper aliran tersebut. Beberapa teknik itu juga diyakini dapat diterapkan pada algoritma chiper aliran lain yang tidak berdasarkan LFSR.

Algoritma HC-256 diyakini mampu bertahan dari serangan-serangan yang disebutkan di atas. Sekarang mari kita lihat keamanan dari setiap komponen dari HC-256.

3.1. Periode perulangan

HC-256 dapat dikatakan memiliki rentang perulangan yang sangat besar. Hal ini disebabkan oleh 65547 state yang dapat dibentuk oleh algoritma tersebut. Namun, angka periode perulangan yang pasti pada algoritma HC-256 sukar untuk dihitung. Diyakini periode perulangan yang terjadi adalah sekitar 2^{65546}[2]

(dengan asumsi setiap pengubahan nilai elemen P dan Q adalah random). Angka yang sangat besar ini juga menjamin keamanan algoritma HC-256 dari serangan yang dilakukan dengan waktu yang lama.

3.2. Keamanan kunci

Kunci yang terdapat pada HC-256 diproses dengan cara yang sangat tidak linear. HC -256 menggunakan 2 buah fungsi (fungsi keluaran dan fungsi *feedback*) tidak linear yang menjamin sedikitnya informasi yang bocor dan juga menjamin keamanan kunci.

HC-256 menggunakan 2 buah tabel berukuran 1024 dengan masing-masing elemennya berukuran 32 bit. Korelasi kedua tabel tersebut diperumit dengan menggunakan salah satu tabel sebagai S-Box untuk tabel lainnya, dan bila ditambahkan fungsi feedback akan sangat sulit untuk mendapatkan kunci kembali.

3.3. Keamanan Initialitation Process

Pembentukan tabel inisialiasi dalam algoritma HC-256 dilakukan dengan 2 tahap yaitu mengekspan key dan *IV* ke dalam tabel P dan Q. Pada tahap ini, setiap bit yang diperoleh dari *IV* dan key telah mempengaruhi seluruh tabel P dan Q. Efeknya, akan tercipta perbedaan di antara kedua tabel tersebut yang tidak berkolerasi sama sekali. Setelah proses ekspansi tersebut selesai, algoritma tersebut juga akan memproses kedua buah tabel dengan fungsi pembuatan keystream tanpa menghasilkan output sebanyak 4096 kali. Hal ini bertujuan untuk sangat mengacak nilai yang terdapat pada tabel P dan Q, dan diharapkan bila terdapat perbedaan kunci dan IV dari yang seharusnya, akan sangat merusak aliran data yang ada.

4. Perhitungan Performansi Implementasi HC-256[3]

Subbab ini bertujuan untuk menghitung performansi dari implementasi algoritma HC-256 dengan menggunakan bahasa Java.

4.1. Parameter Evaluasi

Adapun parameter evaluasi yang dilakukan adalah dengan menghitung:

a. Penghitungan proses inisiasi
   Dilakukan dengan pseudo code sebagai berikut:

```
Read 10 keys from file
For each key
   Start timer
   Repeat 70 times
  Initialize cipher with key
       Stop timer
```

b. Penghitungan pembangkitan stream
   Dilakukan dengan pseudo code sebagai berikut:

```
Initialize cipher with key
Repeat 128 times
   Start timer
      Generate key stream (B blocks of N
bytes each)
   Stop timer
```

c. Penggunaan Memori
   Penghitungan penggunaan memori dilakukan dengan estimasi dengan memperhatikan ukuran file kelas dan ukuran data yang digunakan (misalnya ukuran S-Box yang digunakan)

Dalam penghitungan ini, terdapat beberapa kesulitan antara lain adalah:

- Perhitungan yang dilakukan tidak didasarkan pada penghitungan apapun. Hal ini dilakukan secara subjektif
- Untuk dapat mengimplementasikan algoritma tersebut secara penuh, perlu pemahaman yang mendalam mengenai algoritma yang dipakai, terutama dalam hal ini HC-256 yang mempunyai banyak optimisasi. Hal ini sangat penting untuk diperhatikan agar esensi dari algoritma ini terpenuhi

4.2. Hasil

Adapun hasil yang diperoleh adalah sebagai berikut:

a. Perhitungan proses inisiasi
   Hasil perhitungannya adalah sebagai berikut:

Output Size	Speed
2 blocks of 4096 bytes	3430.5137 keys/s

b. Perhitungan pembangkitan stream
   Hasil perhitungannya adalah sebagai berikut:

Output Size	Speed
11 blocks of 4096 bytes	433.8398 Mbps

c. Penggunaan memori
   Hasil perhitungannya adalah sebagai berikut:

Chiper	Size
HC-256	2714 bytes

5. Kesimpulan

1. Algoritma HC-256 merupakan algoritma chiper aliran yang sangat baik. Hal ini dapat dilihat dari ketahanan algoritma ini dalam menghadapi serangan dan sampai saat ini masih belum ditemukan adanya celah yang bisa ditembus dengan sempurna
2. HC-256 merupakan algoritma yang cukup cepat dalam pembentukan stream, namun cukup lambat dalam pembentukan key. Hal ini kemungkinan disebabkan penggunaan lookup tabel yang cukup besar dalam proses awal yang jarang digunakan dalam algoritma lain.

DAFTAR REFERENSI

[1] Munir, Rinaldi, *Diktat Kuliah IF5054 Kriptografi*, Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung, 2007

[2] Wu, Hongjun, *A New Stream Chiper HC-256*, Institute for Infocomm Research, Singapure

[3] Anand, Chandana, Kartika Bhimavarapu, Sirija PuramShetty, *Performance of selected eSTREAM candidates in Java & Assembly Language*,