# RC4 Stream Cipher

Endang, Vantonny, dan Reza

*Departemen Teknik Informatika*
*Institut Teknologi Bandung*
*Jalan Ganesha 10 Bandung 40132*

*E-mail : if10010@students.if.itb.ac.id*
*if10073@students.if.itb.ac.id*
*if11059@students.if.itb.ac.id*

**Abstrak**

Sistem kriptografi yaitu suatu fasilitas untuk mengkonversikan plaintext ke ciphertext atau menkonversikan chiperteks ke plainteks. Secara umum dalam proses enkripsi dan dekripsi dikenal dua macam cipher berdasarkan cara kerja penyandiannya, yaitu Stream Cipher dan block chiper. Stream cipher adalah suatu sistem dimana proses enkripsi dan dekripsinya dengan cara bit per bit. Sedangkan Sistem block cipher mengkodekan data dengan cara memebagi plaintext menjadi per blok dengan ukuran yang sama dan tetap.

Salah satu jenis stream chiper adalah RC4. RC4 stream cipher yang merupakan salah satu jenis stream cipher kriptografi yang didesain oleh Ron Rivest di laboratorium RSA ( RSA Data Security inc ) pada tahun 1987. RC4 sendiri merupakan ini merupakan teknik enkripsi yang dapat dijalankan dengan panjang kunci yang variabel dan beroperasi dengan orientasi byte.

***Kata kunci**: RC4, stream chiper*, block chiper

## 1. Pendahuluan

RC4 Stream Chiper merupakan salah satu jenis algoritma yang mempunyai sebuah S-Box, S0,S1,……,S255, yang berisi permutasi dari bilangan 0 sampai 255. Pada algoritma enkripsi ini akan membangkitkan pseudorandom byte dari key yang akan dikenakan operasi Xor terhadap plaintext untuk menghasilkan ciphertext. Untuk menghasilkan plaintext semula, maka ciphertext nya akan dikenakan operasi Xor terhadap pseudorandom bytenya. Pada RC4 Menggunakan dua buah indeks yaitu i dan j di dalam algoritmanya. Indeks i digunakan untuk memastikan bahwa suatu elemen berubah, sedangkan indeks j akan memastikan bahwa suatu elemen berubah secara random.

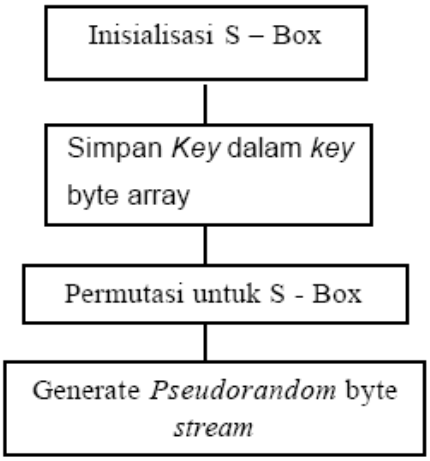
Secara garis besar algoritma dari metode RC4 Stream Cipher ini terbagi menjadi dua bagian, yaitu : key setup dan stream

generation. Pada. Key Setup terdapat tiga tahapan proses di dalamnya, yaitu Inisialisasi S-Box, Menyimpan key dalam Key Byte Array, Permutasi pada S-Box. Pada Stream Generation akan menghasilkan nilai pseudorandom yang akan dikenakan operasi XOR untuk menghasilkan ciphertext ataupun sebaliknya yaitu untuk menghasilkan plaintext. Algoritmanya Stream Generation adalah sebagai berikut:

- Isi indeks i dan j dengan nilai 0
- Untuk i = 0 sampai i = panjang plaintext
- Isi nilai i dengan hasil operasi ( i + 1 ) mod 256
- Isi nilai j dengan hasil operasi ( j + S( i ) ) mod 256
- Swap S( i ) dan S( j )
- Isi nilai t dengan hasil operasi ( S( I ) + (S( j ) mod 256)) mod 256
- Isi nilai y dengan nilai S( t )
- Nilai y dikenakan operasi XOR terhadap plaintext
- Tambahkan i dengan 1, kembali ke 2.

**2. Enkripsi dan Dekripsi RC4**

Proses enkripsi dan dekripsi mempunyai proses yang sama sehingga hanya ada satu fungsi yang dijalankan untuk menjalankan kedua proses tersebut. Berikut ini akan diberikan sebuah bagan yang menggambarkan rangkaian proses yang dijalankan untuk mengenkripsi atau mendekripsi:




Untuk lebi jelasnya Tahapan-tahapan enkripsi dan deskripsi adalah sebagai berikut:

a. Pengguna memasukkan secret key
b. Inisialisasi awal S-Box berdasarkan indeks
c. Simpan secret key yang telah dimasukkan user ke dalam array 256 byte.
d. Bangkitkan nilai pseudorandom berdasarkan nilai key sequence.
e. Proses permutasi nilai dalam S-Box selama 256 kali.

f. Bangkitkan nilai pseudorandom key byte stream berdasarkan indeks dan nilai S-Box.
g. Lakukan operasi XOR antara plaintext /ciphertext dan pseudorandom key

**3. Permasalahan RC4 Stream Chiper**

Permasalahan metode RC4 Stream Cipher ini adalah sebagai berikut:

a. Terlalu tingginya kemungkinan terjadi S-Box yang sama karena nilai pseudorandom yang sama seringkali dibangkitkan berulang, hal ini terjadi karena kunci user diulang-ulang untuk mengisi 256 bytes array. Meskipun metode ini memungkinkan penggunaan variabel yang panjangnya dapat mencapai 256 karakter/byte namun pada kenyataannya jarang sekali ada yang menggunakan kunci sepanjang itu, selain karena sulit mencari kombinasinya juga sulit untuk mengingatnya. Sehingga jika kunci yang digunakan sebanyak 8 byte misalnya, maka kunci ini akan diulang sebanyak 32 kali untuk mengisi key byte array sampai penuh.
b. Enkripsi RC4 adalah XOR antara data bytes dan pseudorandom byte stream yang dihasilkan dari kunci, maka penyerang akan mungkin untuk menentukan beberapa byte pesan orisinal dengan meng-XOR dua set cipher byte, bila beberapa byte plaintext diketahui (atau mudah ditebak). Diasumsikan A berhasil menyadap dua buah message berbeda yang dienkripsi menggunakan algoritma stream cipher dengan menggunakan kunci yang sama. A kemudian meng-XOR-kan kedua ciphertext yang berhasil disadapnya untuk menghilangkan pengaruh rangkaian kunci. Jika A berhasil mengetahui plaintext dari salah satu message terenkripsi tersebut maka A akan dengan mudah menemukan plaintext message yang lain tanpa mengetahui rangkaian kuncinya.

**4. Pemecahan Masalah**

Untuk mengatasi permasalahan di atas yang terdapat pada stream chipper RC4 maka ada beberapa yang dapat dilakukan yaitu:

a. Gunakanlah kunci yang panjang (minimal panjang kunci • 3 karakter dan maksimal • 255 karakter) agar kemungkinan kunci dimasukkan berulang dalam key byte array semakin kecil dan gunakan kombinasi yang berlainan.
b. Usahakan untuk tidak menggunakan kunci yang sama untuk mengenkripsi file yang berbeda.
c. Jika kita akan menggunakan kunci yang sama untuk setiap kali mengenkripsi file, maka diperlukan Initialization Vector ( IV ) pada secret key. Jika IV yang digunakan untuk setiap kali proses enkripsi dijalankan tidak pernah sama maka akan dihasilkan ciphertext yang berbeda meskipun dienkrip plaintext yang sama.
d. Mengacak (mengubah susunan) plaintext sebelum diubah ke dalam cipher, sehingga jika seorang pengganggu memperoleh 1 byte data dari plaintext maka ia tidak dapat memperoleh data yang lainnya dengan cara meng- XOR-kan dua buah ciphertext dan byte data yang ia ketahui.
e. Mengubah metode pengisian key ke dalam key array. Caranya adalah key cukup diisikan sekali dalam array kemudian sisa variabel array key yang lainnya akan diisi dengan nilai yang dibangkitkan secara

**5. Pengembangan RC4 Stream Cipher**

Pengembangan algoritma dari RC4 Stream cipher di sini akan dilakukan dengan cara membangkitkan nilai random untuk pengisian key byte array sehingga pengisian key ke dalam array tidak berulang dan dalam penerapannya akan dipadukan dengan teknik dasar enkripsi blocking untuk mengacak susunan plainteks sebelum diubah ke dalam ciphertext.

Proses enkripsi terdiri atas tahapan, antara lain :

1) Random Number Generator

Akan diambil sebuah nilai yang akan dijadikan “seed”. Kemudian dibangkitkan nilai random berdasarkan nilai “seed” ini. Algoritmanya adalah sebagai berikut :

a) Kerjakan fungsi Rnd -1
b) Kerjakan fungsi Randomize dengan nilai seed yang diambil.

2) Inisialisasi S-Box

Pada tahapan ini, S-Box akan diisi dengan nilai sesuai indeksnya untuk mendapatkan S-

Box awal. Algoritmanya adalah sebagai berikut :

a) Untuk i = 0 sampai i = 255 lakukan
b) Isikan S ke i dengan nilai i
c) Tambahkan i dengan 1 , kembali ke 1.

cara membangkitkan nilai random untuk pengisian key byte array

**6. Kesimpulan**

RC4 stream cipher yang merupakan salah satu jenis teknik stream cipher kriptografi yang dapat dijalankan dengan panjang kunci yang variabel dan beroperasi dengan orientasi byte.

Seperti halnya teknik-teknik yang lain, RC4 mempunyai beberapa kelemahan yaitu Terlalu tingginya kemungkinan terjadi S-Box yang sama karena nilai pseudorandom yang sama seringkali dibangkitkan berulang**.**

Untuk mengatasi permasalahan dengan menggunakan kunci yang panjang, Usahakan untuk tidak menggunakan kunci yang sama untuk mengenkripsi file yang berbeda, Mengacak (mengubah susunan) plaintext sebelum diubah ke dalam cipher, dan lain-lain. Selain itu juga dilakukan Pengembangan algoritma dari RC4 Stream cipher dengan

[1] http://www.achtung.com/crypto/rc4.html, diakses tanggal 2 Januari 2006 pukul 11:00
[2] http://security.devx.com/bestdefense/2001/mh0201/mh0201-4.asp, diakses tanggal 2 Januari 2002 pukul 11:20
[3] Ir. Rinaldi Munir, M.T, Diktat Kuliah IF5054 Kriptografi, Departemen Teknik Informatika