Algoritma RC4 sebagai Perkembangan Metode Kriptografi

Arie Pratama Sutiono/13509007
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
arie.pratama.s@students.itb.ac.id

Abstrak : Selama beberapa tahun terakhir, dunia telah menjadi saksi perkembangan teknologi informasi yang sangat pesat. Perkembangan pesat ini telah menghasilkan beberapa kemajuan di bidang kriptografi untuk menjaga integritas dan kerahasiaan data. Sistem keamanan tersebut dibuat dengan menggunakan metode-metode kriptografi. Salah satu metode kriptografi modern yang dikembangkan adalah algoritma RC4. RC4 yang juga dikenal dengan ARCFOUR (Alleged RC4) adalah software stream cipher yang paling banyak digunakan. RC4 juga digunakan dalam protokol yang populer seperti SSL (Secure Sockets Layer) yang digunakan untuk melindungi lalu lintas internet dan WEP yang digunakan untuk mengamankan jaringan wireless.

***Kata Kunci* : Kriptografi, algoritma, RC4. Sistem keamanan**

I. PENDAHULUAN

Dewasa ini, sistem keamanan memainkan peran penting dalam perancangan sistem teknologi informasi. Area dan jangkauan aplikasi yang membutuhkan sistem keamanan seperti tanpa ada akhirnya, contohnya : komunikasi internet, perangkat-perangkat nirkabel (misalnya : handphone dan wireless LAN), komunikasi antar mobil, e-commerce, e-Banking, proteksi hak cipta untuk media digital, elektronik bar code pada produk-produk, stampel elektronik, dll. Maka rancangan dan implementasi untuk solusi sistem keamanan di masa mendatang akan sangat menantang untuk dilakukan, karena *penyerang* biasanya akan menyerang link yang lemah dalam sistem keamanan. Penyerangan ini dapat menggunakan berbagai opsi, contohnya menerobos crypto-algorithm. Dalam kebanyakan kasus, crypto-algorithm adalah perangkat inti dalam aplikasi sistem keamanan dan karena itu crypto-algorithm harus didesain secara hati-hati, kemudian dipilih, dan diimplementasikan untuk menghindari inti kriptografi tersebut menjadi link yang terlemah apapun solusi keamanannya.

Teknik kriptografi modern adalah transformasi matematika (algoritma). Teknik ini memperlakukan suatu pesan sebagai angka atau elemen aljabar dalam sebuah ruang dan mengubahnya menjadi sebuah pesan lain yang mempunyai arti, atau pesan yang dapat dimengerti (chipertext). Untuk menerjemahkan pesan – pesan tersebut, tentu harus ada proses yang mengembalikan informasi asli yang terkandung dalam pesan – pesan tersebut. Kita harus membalikan proses transformasi tersebut, proses pembalikan transformasi disebut juga dengan *dekripsi*. Biasanya algoritma enkripsi dan dekripsi diberikan parameter berupa *cryptographic keys* (kunci kriptografi). Sebuah algoritma enkirpsi dan sebuah algoritma dekripsi, ditambah dengan deskripsi format sebuah pesan dan kunci, akan membentuk sebuat sistem kriptografi atau disebut juga sebagai *cryptosystem.*

Sekarang ini ada 2 macam protokol kriptografi yang populer, yaitu : *symmetric-key* dan *asymmetric-key*. Pada protokol *symmetric-key*, sebuah kunci (kunci rahasia atau kunci privat) digunakan oleh kedua pihak yang berkomunikasi untuk mengenkripsi dan mendekripsi pesan. Protokol jenis ini sudah lama digunakan. Penemuan protokol ini diperkirakan sekitar tahun 1900 BC, yang meliputi kode kriptografi dan teknik cipher (contohnya Caesar Cipher, Hill Cipher, Vernam Cipher). Sedangkan pada protokol kriptografi dengan *asymmetric key*, satu kunci di sebarluaskan ke publik dan menahan yang lainnya sebagai *private key*. *Private key* tersebut didesain agar sulit untuk dipecahkan, walaupun telah mengetahui kunci yang disebarluaskan ke publik.

II. DASAR TEORI

II.1. Definisi Kriptografi

Kriptografi berasal dari bahasa Yunani yaitu *kryptos* yang berarti “rahasia” dan *graph* yang berarti “menulis” atau “belajar”. Jadi kriptografi adalah ilmu dan seni yang mempelajari cara-cara untuk menyembunyikan informasi dengan cara menyamarkannya menjadi sandi yang sulit ditemukan maknanya.

Ada empat tujuan mendasari dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi, yaitu :

- Kerahasiaan, adalah layanan yang digunakan untuk menjaga informasi dari pihak lain, yang dapat membuka informasi tersebut adalah pihak yang memiliki kunci rahasia untuk mendekripsi data tersebut.
- Integritas data. Hal ini berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk medeteksi

manipulasi data oleh pihak-pihak yang tidak mempunyai hak, antara lai penyisipan, penghapusan, dan penggantian data lain ke dalam data sebenarnya.

- Autentikasi. Hal ini berhubungan dengan identifikasi/pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan harus diautentikasi keasliannya, isi datanya, waktu pengiriman, dan lain lain.
- Non Repudiasi, atau dapat disebut juga nirpenyangkalan adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/ terciptanya suatu informasi oleh yang mengirimkan/membuatnya.

Algoritma yang berfungsi untuk melakukan tujuan kriptografis disebut sebagai **algoritma sandi**. Algoritma tersebut harus memiliki kekuatan untuk melakukan :

- Konfusi/pembingungan, mempersulit pembaca biasa untuk memecahkan pesan yang sudah dienkripsi menjadi sandi-sandi tanpa memakai algoritma pendekripsinya.
- Difusi/peleburan, yaitu dengan cara menghilangkan karakteristik dari informasi yang dienkripsi.

Jika algoritma sandi memiliki kekuatan tersebut maka algoritma tersebut dapat digunakan untuk mengamankan informasi. Pada implementasinya sebuah algoritma sandi harus memperhatikan kualitas layanan/ QoS (Quality of Service) dari keseluruhan sistem dimana dia diimplementasikan. Algoritma sandi yang handal adalah algoritma sandi yang kekuatannya terletak pada kunci dan bukan kerahasiaan algoritma itu sendiri, maksudnya adalah kehandalan diukur dari tingkat kesulitan kunci pendekripsian dari sandi, bukan seberapa rahasia algoritma sandi tersebut. Teknik dan metode untuk menguji kehandalan algoritma sandi adalah kriptanalisa.

Dasar matematis yang mendasari proses enkripsi dan dekripsi adalah relasi antar dua himpunan, dimana satu himpunan berisi elemen teks terang (plaintext) dan yang lain berisi elemen teks sandi (ciphertext).

Secara umum berdasarkan kesamaan kuncinya, algoritma sandi dibedakan menjadi :

- Kunci Simetris/*symmetric-key*, sering disebut juga dengan kunci pribadi/*private key*.
- Kunci Asimetri/*asymmetric-key*, sering disebut juga dengan kunci publik/*public key*.

Berdasarkan arah implementasi dan pembabakan zamannya :

- Algoritma sandi klasik
- Algoritma sandi modern

Berdasarkan kerahasiaan kuncinya dibedakan menjadi :

- Algoritma sandi kunci rahasia (*secret key*).
- Algoritma sandi kunci publik (*puclic key*).

Ada juga kriteria-kriteria yang harus dimiliki suatu algoritma kriptografi menurut *National Bureau Of Standards* (NBS), yang sekarang diganti namanya dengan *National Institute of Standards and Technology* (NIST). Kriteria-kriteria tersebut antara lain adalah

- Algortima harus memiliki tingkat keamanan yang tinggi
- Algoritma harus spesifik dan mudah dimengerti
- Algoritma harus dapat diadaptasi pada aplikasi yang beragam
- Algoritma harus ekonomis dalam pemgimplementasiaanya pada perangkat keras
- Algoritma harus efisien dalam penggunaannya
- Algoritma harus dapat berlaku secara umum
- Algoritma dapat dipisahkan

II.2. Sejarah Kriptografi

Sebelum era modern, kriptografi hanya bersangkutan dengan keterpercayaan pesan, dengan mengubah pesan yang dapat dimengerti menjadi tidak dapat dimengerti kemudian proses tersebut dibalikan untuk dapat dibaca, menyebabkan pesan tersebut tidak dapat dibaca oleh penyusup dan mata-mata tanpa pengetahuan khusus yang menyangkut dengan kunci-kunci dekripsi. Enkripsi digunakan untuk menjaga kerahasiaan dalam berkomunikasi, seperti mata-mata, pemimpin militer, dan diplomat.

Tulisan rahasia yang pertama kali memerlukan tidak lebih dari sekedar pena dan kertas, karena kebanyakan orang pada masa itu tidak dapat membaca. Ketika sudah lebih banyak yang dapat membaca, atau dalam menghadapi musuh yang dapat membaca tentu dibutuhkan kriptografi yang sebenarnya. Cipher klasik yang utama berbentuk cipher transposisi. Chiper transposisi mengubah urutan huruf yang dituliskan dalam sebuah pesan.

Gambar 1: alat pengenkripsi Yunani kuno

Sedangkan chiper substitusi mengubah huruf ke dalam huruf lain (misalnya kalimat “fly at once” menjadi “gmz bu podf”, dengan mengubah setiap huruf menjadi huruf Latin). Contoh chiper substitusi kuno adalah Caesar cipher. Caesar chiper menggunakan prinsip mengganti huruf-huruf dalam pesan menjadi beberapa huruf setelah huruf asli. Metode ini dinamai berdasarkan laporan bahwa yang menggunakannya pertama kali adalah Julius Caesar, ia menggunakannya dengan mengganti huruf pesan asli dengan 3 huruf setelah huruf asli (Caesar menggunakannya untuk berkomunikasi dengan jendral-jendralnya saat ekpedisi militer). Ada juga beberapa Hewbrew chiper versi kuno. Penggunaan kriptografi yang paling awal diketahui adalah ciphertext yang di pahat pada batu di Mesir (sekitar 1900BC), tapi hal ini mungkin dilakukan hanya untuk kesenangan oleh kaum terpelajar. Penggunaan yang tertua berikutnya adalah resep roti dari

Mesopotamia. Kriptografi digunakan dalam Kama Sutra sebagai cara pasangan untuk berkomunikasi tanpa diketahui pihak lain.

Orang-orang Yunani kuno sudah mengenal chiper, misalnya chiper transposisi yang digunakan oleh militer bangsa Sparta. Steganography pertama dikembangkan sejak zaman kuno. Steganography pada zaman modern dilakukan dengan misalnya : tinta yang tidak kelihatan, microdots, dan digital watermark.

Cipherteks yang dibuat oleh cipher kuno, maupun sebagian cipher modern selalu mengungkap informasi statistik mengenai teks yang sebenarnya, sehingga hal ini dapat dimanfaatkan untuk memecahkannya. Pada abad ke 9, setelah penemuan analisis frekuensi yang kemungkinan ditemukan oleh matematikawan Arab yaitu Al-Kindi hampir semua cipher kuno menjadi sulit dipecahkan dengan hanya membaca saja. Metode chiper yang seperti itu masih digunakan sampai sekarang ini, dan kebanyakan berupa puzzle. Al-Kindi menulis buku tentang kriptografi yang berjudul *Risalah fi Istikhraj al-Mu'amma* (Naskah untuk Menguraikan Pesan Kriptografi) yang mendeskripsikan teknik kriptanalisis pertama dan meliputi cipher polialphabetik.

Pada dasarnya, semua cipher dapat dipecahkan oleh kriptanalisis dengan menggunakan teknik frekuensi analisis sampai pengembangan cipher polialphabetik. Cipher polialphabetik dikembangkan oleh Leon Battista Alberti sekitar tahun 1467. Inovasi Alberti menggunakan cipher yang berbeda (contohnya dengan penggantian alphabet) untuk bermacam-macam bagian yang berbeda. Ia juga mengembangkan alat cipher yang pertama yang berupa sebuah roda.

Gambar 2 mesin cipher Perancis pada abad ke 16

Gambar 3 Surat yang sudah dicipher dari Gabriel de Luetz d'Aramon, duta besar Perancis

Walaupun analisis frekuensi merupakan sebuah teknik yang sudah umum dan efektif untuk berbagai chiper namun enkripsi juga masih efektif dalam praktiknya, dan tidak jarang kriptanalisa tidak menyadari penggunaan teknik tersebut. Memecahkan sebuah pesan tanpa menggunakan frekuensi analisis pada umumnya membutuhkan pengetahuan tentang cipher yang digunakan dan kunci yang terlibat. Hal ini membuat kasus-kasus seperti penyuapan, pencurian,dll menjadi pendekatan yang menarik untuk pemecahan kriptanalisis. Pada abad 19 ditemukan bahwa kerahasian algoritma cipher tidak efektif lagi untuk diterapkan untuk menjaga kerahasiaan sebuah pesan, faktanya skema kriptografi seharusnya sudah cukup aman, walaupun musuh sangat mengerti tentang algoritma cipher itu sendiri. Keamanan dari kunci yang digunakan seharusnya juga sudah cukup untuk untuk sebuah cipher yang bagus untuk menjaga keamanan pesan. Prinsip fundamental ini dikemukakan oleh Aguste Kerckhoffs dan prinsip ini pada umumnya disebut prinsip Kerckhoffs. Prinsip tersebut dikemukakan ulang oleh Claude Shannon, yang merupakan pencipta teori informasi dan teori dasar kriptografi.

Peralatan dan penanganan yang berbeda telah digunakan untuk bekerja dengan cipher. Salah satu yang terdahulu digunakan oleh bangsa Yunani kuno adalah sebuah tongkat, dimana tongkat tersebut digunakan bangsa Sparta untuk mentransposisi cipher (gambar 1). Pada abad pertengahan, penanganan lain telah ditemukan, contohnya adalah cipher grille, yang digunakan juga untuk steganography. Seiiring dengan pengembangan cipher polialphabetik, semakin banyak pula bentuk-bentuk alat dan penanganan cipher, misalnya adalah cipher disk buatan Alberti, skema Johannes Trithemius tabula recta, dan multi silinder buatan Thomas Jefferson. Banyak mesin pengenkripsi dan pendekripsi diciptakan dan dikembangkan pada abad ke 20 dan beberapa yang dipatenkan.

Perkembangan komputer digital dan elektronika setelah perang dunia kedua memungkinkan untuk membuat chiper yang lebih kompleks lagi. Berbeda dengan chiper kuno yang mengenkripsi dengan tulisan dan bahasa, komputer memungkinkan enkripsi bermacam-macam data dalam format biner. Cipher komputer dapat dikategorikan berdasarkan operasi bilangan biner yang dilakukannya. Komputer juga membantu kriptanalisis yang dikompensasikan sampai pada batas tertentu untuk meningkatkan kompleksitas cipher. Cipher yang modern telah melewati ilmu kriptanalisis. Jenis cipher ini pada umumnya paling efisien dan sangat sulit untuk memecahkannya.

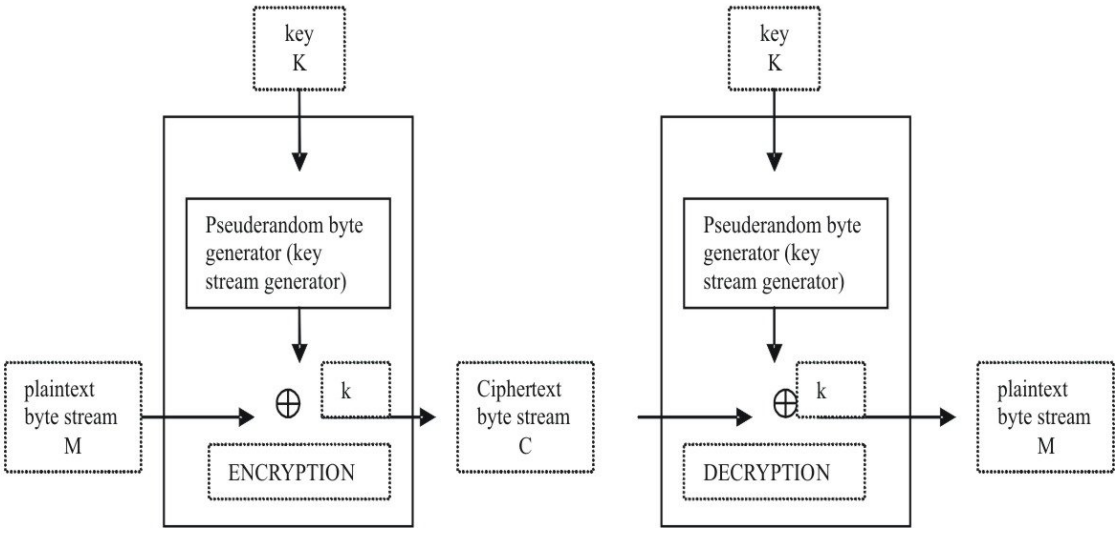
Riset ekstensif yang diadakan secara terbuka terhadap kriptografi adalah sesuatu yang baru. Riset ini baru diadakan pada tahun 1970an. Dewasa ini, personel IBM mendesain algoritma yang menjadi Data Encryption Standard (DES) yang dibuat oleh Whitfield Diffie dan Martin Hellman. Algoritma RSA dipublikasikan oleh Martin Gardner pada kolom Scientific American. Sejak itu, kriptografi menjadi alat yang umum untuk digunakan dalam bidang komunikasi, jaringan komputer, dan sistem keamanan komputer pada umunya. Sebagian dari teknik kriptografi modern tidak dapat dipecahkan bila persoalan matematika tertentu tidak dapat diatangani. Contoh

persoalan matematika yang dipakai dalam kriptografi adalah faktorisasi bilangan bulat atau logaritma diskret, dengan kata lain teknik kriptografi tersebut mempunyai hubungan yang erat dengan matematika. Namun tidak ada jaminan bahwa suatu teknik kriptografi aman, yang ada hanya jaminan bahwa rahasia algoritma tersebut tetap aman jika sangat sulit dipecahkan.

Intinya pada awal abad ke 20, kriptografi banyak bersangkutan dengan pola linguistik dan lexicografik. Sejak itu fokus kriptografi mulai bergeser, dan sekarang kriptografi menggunakan ilmu matematika yang meliputi teori informasi, komplesitas komputasi, statistik, kombinatorik, aljabar abstrak, teori bilangan, dan matematika umum yang mempunyai batas.

II.3. Definisi *Stream Chiper*

Dalam kriptografi, stream chiper adalah sebuah kunci simetrik dimana bit sebuah plainteks digabungkan dengan urutan pseudorandom bit stream, yaitu dengan operator Xor. Dalam sebuah stream chiper, digit dari plainteks dienkripsi satu per satu dan transformasi digit setelahnya berbeda-beda selama proses enkripsi berlangsung, dengan kata lain stream chiper akan mengenkripsi sebuah data per-satuan data contohnya adalah : bit, byte, nible, atau 5 bit dan setiap mengenkripsi satu satuan data, digunakan kunci yang merupakan hasil pembangkitan dari kunci sebelumnya. Nama lain dari stream chiper adalah state chiper.



Gambar 4 :Diagram struktur stream chiper

Dalam gambar diatas, sebuah kunci adalah masukan yang dapat membangkitkan rangkaian kunci yang menghasilkan sebuah stream number sebanyak 8 bit dan acak. Output dari pembangkit kunci tersebut disebut juga *key stream*. Output ini dikombinasikan bit per bit per satuan waktu dan menggunakan operasi Xor. Sebagai contoh : 01101100 sebagai byte yang dibangkitkan oleh pembangkit dan 11001100 adalah plainteks selanjutnya, maka hasil cipherteksnya adalah :

1001100 plainteks

1101100 keystream

0100000 cipherteks

Selain stream chiper, juga ada *Block Chiper*. Kedua jenis kunci ini sebenarnya mempunyai karakteristik yang sama. Stream chiper dapat diimplementasikan sebagai block chiper dan begitu juga sebaliknya. Block cipher melakukan prosesnya pada data dengan transformasi yang tentu pada suatu ukuran besar blok data dari plainteks, sedangkan stream cipher bekerja berdasarkan pengolahan digit plainteks tersendiri dengan transformasi yang berbeda-beda setiap waktunya.

III. RC4

III.1. Sejarah Algoritma RC4

RC4 didesain oleh Ron Rivest yang berasal dari RSA Security pada tahun 1987. RC sendiri mempunyai singkatan resmi yaitu “Rivest Chiper”, namun juga dikenal sebagai “Ron’s Code” RC4 sebenarnya dirahasiakan dan tidak dipublikasikan kepada khalayak ramai, namun ternyata ada orang yang tidak dikenal menyebarkan RC4 ke mailing list *Cypherpunks*. Kemudian berita ini dengan cepat diposkan ke sci.crypt newsgroup, dan dari newsgroup ini kemudian menyebar luas di internet. Kode yang dibocorkan tersebut dipastikan keasliannya karena output yang dikeluarkan sama dengan software-software yang menggunakan RC4 yang berlisensi. Nama RC4 sudah dipatenkan, sehingga RC4 sering disebut juga ARCFOUR atau ARC4 (Alleged RC4) untuk menghindari masalah pematenan. RSA Security tidak pernah secara resmi merilis algoritma tersebut,

namun Rivest secara pribadilah yang merilisnya tersebut dengan menhubunkan Wikipedia Inggris ke catatan-catatan yang ia punya. RC4 telah menjadi bagian dari protokol enkripsi yang standard dan sering digunakan, termasuk WEP dan WPA untuk wireless card, serta TLS. Faktor utama yang menjadi kesuksesan dari RC4 adalah kecepatannya dan kesederhanaannya dalam menangani banyak applikasi, sehingga mudah untuk mengembangkan implementasi yang effisien ke software dan hardware.

III.2. Deskripsi algoritma RC4

RC4 menghasilkan pseudorandom stream bit. Seperti halnya stream cipher lainnya, algoritma RC4 ini dapat digunakan untuk mengenkripsi dengan mengombinasikannya dengan plainteks dengan menggunakan bit-wise Xor (Exclusive-or). Proses dekripsinya dilakukan dengan cara yang sama (karena Xor merupakan fungsi simetrik). Untuk menghasilkan keystream, cipher menggunakan state internal yang meliputi dua bagian :

1. Sebuah permutasi dari 256 kemungkinan byte.
2. 2 Indeks-pointer 8-bit.

Permutasi di inisialisasi dengan sebuah variabel panjang kunci, biasanya antara 40 sampai 256 bit dengan menggunakan algoritma *key-scheduling* (KSA). Setelah proses ini selesai, stream yang terdiri dari sekumpulan bit tersebut terbentuk dengan menggunakan *Pseudo-Random Generation Algorithm* (PRGA). Berikut ini akan dijelaskan tentang kedua algoritma tersebut.

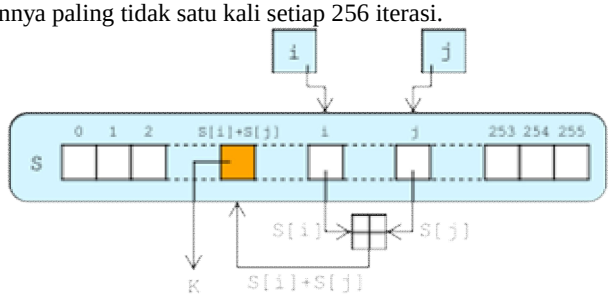
III.3. Key-Scheduling Algorithm (KSA)

Algoritma key scheduling digunakan untuk menginisialisasi permutasi di array “S”. panjang kunci didefinisikan sebagai jumlah byte di kunci dan mempunyai rentang panjang kunci dari 1 sampai 256, khususnya antara 5-16 tergantung dari panjang kunci 40-128bit. Pertama-tama array “S” diinisialisasi untuk identitas permutasi. S kemudian diproses ke 256 iterasi dengan cara yang sama dengan PRGA utama, tapi juga dikombinasikan dalam byte dari kunci dalam waktu yang bersamaan. Berikut adalah algoritma KSA :

```
for i from 0 to 255
    S[i] := i
endfor
j := 0
for i from 0 to 255
    j := (j + S[i] + key[i mod
keylength]) mod 256
    swap values of S[i] and S[j]
endfor
```

III.4. Pseudo-Random Generation Algortihm

PRGA (Pseudo-Random Generation Algortihm) memodifikasi state dan output sebuah byte dari keystream. Hal ini penting karena banyaknya dibutuhkan iterasi. Dalam setiap iterasi, PRGA menginkremen **i**, menambahkan nilai S yang ditunjuk oleh **i** sampai **j,** kemudian menukar nilai S[i] dan S[j], lalu mengembalikan elemen dari S di lokasi S[i] + S[j] (modulo 256). Setiap elemen S ditukar dengan elemen lainnya paling tidak satu kali setiap 256 iterasi.



Gambar 5 Tahap pencarian dari RC4. Byte output dipilih dengan mencari nilai S[i] dan S[j], menambahkannya dengan modulo 256, dan mencari hasil penjumlahannya di S

Realisasi algoritma PRGA dapat dilihat sebagai berikut :

```
i := 0
j := 0
while GeneratingOutput:
    i := (i + 1) mod 256
    j := (j + S[i]) mod 256
    swap values of S[i] and S[j]
    K := S[(S[i] + S[j]) mod 256]
    output K
endwhile
```

III.5. Implementasi dari Algoritma RC4

Banyak stream cipher dibuat berdasarkan Linear Feedback Shift Registers (LFSRs) yang efisien di perangkat keras namun kurang efisien dalam perangkat lunak. Desain dari RC4 menghindari penggunaan LFSRs, dan algoritma ini ideal untuk implementasi perangkat lunak karena hanya menggunakan manipulasi byte. Algoritma ini menggunakan 256 byte memori untuk kunci, yaitu key[0] sampai key[k-1], dan variabel integer **i, j,** dan **y**. Reduksi modular dari beberapa nilai modulo 256 dapat dilakukan dengan sebuah **bitwise AND** dengan 255 (yang ekuivalen untuk mengambil byte dengan orde rendah dari nilai di pertanyaan).
Berikut ini adalah implementasinya dalam bahasa C

```
unsigned char S[256];
unsigned int i, j;

void swap(unsigned char *s, unsigned
int i, unsigned int j) {
    unsigned char temp = s[i];
    s[i] = s[j];
    s[j] = temp;
}
```

```
/* KSA */
void rc4_init(unsigned char *key,
unsigned int key_length) {
    for (i = 0; i < 256; i++)
        S[i] = i;

    for (i = j = 0; i < 256; i++) {
        j = (j + key[i % key_length] +
S[i]) & 255;
        swap(S, i, j);
    }

    i = j = 0;
}

/* PRGA */
unsigned char rc4_output() {
    i = (i + 1) & 255;
    j = (j + S[i]) & 255;

    swap(S, i, j);

    return S[(S[i] + S[j]) & 255];
}

#include <stdio.h>

int main() {
    int k, output_length;
    unsigned char key[] = "Secret";
// key hardcoded to "Secret"

    output_length = 10;
// number of bytes of output desired

    rc4_init(key, 6);
// length of key is 6 in this case

    k = 0;
    while (k < output_length) {
      printf("%c", rc4_output());
      k++;
    }
}
```

III.6. Tentang Keamanan dari Algoritma RC4

Tidak seperti stream cipher modern, RC4 tidak mengambil *nonce yang* terpisah bersamaan dengan kunci. Hal ini berarti jika kunci *single long-term* digunakan untuk mengenkripsi beberapa stream, kriptosistemnya harus menentukan bagaimana cara mengombinasikan *nonce* tersebut dan kunci *long-term* untuk menghasilkan kunci stream untuk RC4. Sebuah pendekatan untuk menangani hal tersebut adalah dengan membuat sebuah kunci RC4 dengan menggunakan fungsi *hash*. Enkripsi dengan menggunakan RC4 dapat diterobos dan rentan terhadap *bit-flipping attack.* Untuk menanggulangi hal ini, skema enkripsi harus dikombinasikan dengan *message authentication code* yang kuat.

Pada tahun 2001 Fluhrer, Martin, dan Shamir membuat suatu penemuan yang mengejutkan. Dari semua kemungkinan kunci RC4, statistik untuk beberapa byte yang pertama dari keluaran keystream sangat tidak random, dan membocorkan informasi untuk kunci ini. Jika kunci *long-term* dikonkat dengan nonce untuk menghasilkan kunci RC4, kunci *long-term* ini dapat ditemukan dengan menganalisis sejumlah besar pesan yang dienkripsi dengan kunci ini. Efek ini dugunakan untuk menerobos enkripsi WEP dengan jaringan wireless 802.11. Hal ini menyebabkan perebutan untuk pengganti standard WEP.

Pada tahun 2005, Andreas Klein mempresentasikan sebuah analisis terhadap RC4 stream cipher yang menunjukan lebih banyak korelasi antara keystream RC4 dan kunci tersebut. Erik Tews, Ralf-Philip Weinmann, dan Andrei Pychkine menggunakan analisis ini untuk membuat aircrack-ptw, yaitu sebuah perangkat untuk memecahkan 104-bit RC4 yang digunakan di 128-bit WEP dalam waktu 1 menit.

REFERENSI

[1] *docsdrive.com/pdfs/ansinet/itj/2005/307-325.pdf,* 11 Desember 2010.
[2] http://en.wikipedia.org/wiki/RC4 11 Desember 2010.
[3] http://en.wikipedia.org/wiki/Cryptography 11 Desember 2010.
[4] Munir,Rinaldi, "Matematika Diskret Edisi Keempat", Institut Teknologi Bandung, 2006.
[5] http://id.wikipedia.org/wiki/Kriptografi 11 Desember 2010.
[6] http://kriptologi.wordpress.com/ 14 Desember 2010.

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, 29 April 2010

ttd

Arie Pratama Sutiono/13509007