Algoritma Blok Cipher Confidentia

Dimpos A.G. Sitorus (13513083)
Sekolah Teknik Elektro dan Informatika
Insititut Teknologi Bandung
Bandung, Indonesia
dimpossitorus@gmail.com

Mahesa Gandakusuma (13513091)
Sekolah Teknik Elektro dan Informatika
Insititut Teknologi Bandung
Bandung, Indonesia
aseham.krazzy@gmail.com

***Abstract*—Makalah ini adalah proposal untuk mengajukan sebuah algoritma baru untuk blok cipher. Dalam makalah ini akan diberikan penjelasan mengenai algoritma dan implementasinya.**

Keywords—Kriptografi, Algoritma Blok Cipher, Blok Cipher, Confidentia;

I. PENDAHULUAN

Pada makalah ini kami akan mengajukan sebuah algoritma baru untuk kriptografi *block cipher.*. Algoritma ini kami beri nama *Confidentia.* Pada algoritma baru ini akan terdapat proses proses tertentu, yaitu operasi substitusi dan transposisi, memanfaatkan prinsip jaringan feistel dan juga penerapan *cipher* berulang. Penggunaan proses tersebut diharapkan akan memenuhi prinsip *diffusion* dan *confusion*[1]. Dengan memenuhi prinsip tersebut diharapkan juga algoritma *block cipher* ini akan sukar untuk dipecahkan/didekripsi. Algoritma ini dirancang dengan prasyarat ukuran minimal blok data adalah 64 bit, dan panjang blok diketahui dari panjang kunci yang dimasukkan.

II. DASAR TEORI

A. Block Cipher

Block Cipher adalah sebuah fungsi yang memetakan n-bit teks input ke n-bit teks cipher.[3] Hal ini hampir sama dengan cipher substitusi namun dengan ukuran yang lebih besar, misalnya blok ukuran 64 bit (setara dengan 8 karakter)

B. Prinsip *Diffusion dan Confusion*

Prinsip *diffusion* dan *confusion* adalah salah satu cara untuk mengacaukan pola statistik yang dihasilkan dari enkripsi.

Metode *diffusion* bertujuan untuk mengacaukan pola statistik dari pesan dengan cara membuat hasil enkripsi menjadi panjang dan dan banyak data redundan

Metode confusion bertujuan untuk membuat hubungan antara statistik dari teks cipher dan kunci K menjadi sangat kompleks.

C. Jaringan Feistel

Jaringan feistel adalah salah satu cara untuk membuat hubungan antara teks cipher dan kunci K menjadi kompleks. Cara kerja jaringan feistel adalah membagi dua input blok plaintext menjadi blok kiri L dan blok kanan R. Lalu dilakukan enkripsi yang saling bergantung atara blok kiri L dan blok kanan R.

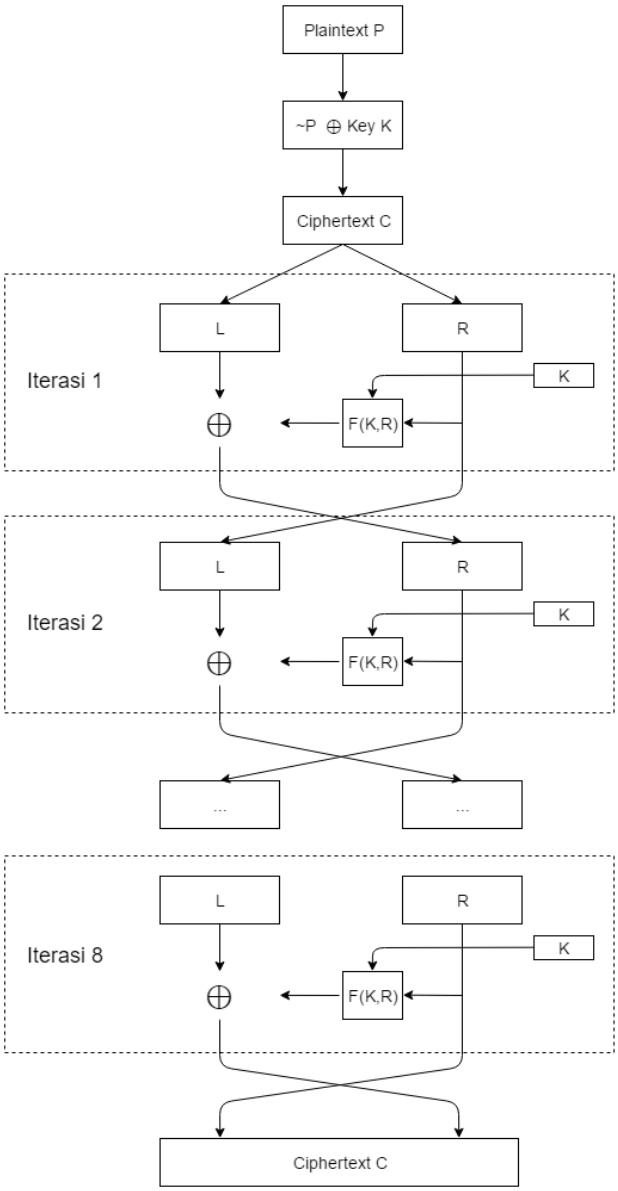
III. RANCANGAN *BLOCK CIPHER*

A. Skema dan Fungsi

Fungsi di bawah ini digunakan dalam jaringan feistel :

$$.F(K,P) = ((P << 8) * \text{S-Box}) \oplus K$$

Kotak-S (S-Box) yang digunakan adalah S-Box AES



Untuk fungsi enkripsi E, blok plaintext pertama kali di bitwise NOT dan kemudian di XOR dengan keys. Lalu kemudian blok dibagi dua, blok kiri (Li) dan blok kanan (Ri). Blok kanan diproses dengan menggunakan kunci F lalu di XOR dengan blok kiri. Lalu Iterasi berikutnya, blok kanan Ri menjadi blok kiri L(i+1) dan blok kiri Li menjadi blok kanan R(i+1) dan proses sebelumnya diulang sampai 8 kali

Untuk fungsi dekripsi D, kita hanya perlu membalikkan proses diatas. Pada skema kita memulai dari bawah.

B. Pseudocode

Berikut adalah pseudo-code untuk aplikasi enkripsi block cipher *Confidentia.* Fungsi akan menerima 2 masukan berupa string plaintext dan string kunci dan memberikan keluaran string hasil berupa ciphertext.

```
function Encrypt(s,k : string) : string {
foreach (blok_plaintext as p) {
  c ← ~p ⊕ k
  L ← (c,0,length(c)/2); R ← (c,length(c)/2,length(c)/2)
  for (i=0) to 8 {
    tmp ← R;
    L ← (R << 8 + s_box(R)) ⊕ k
    R ← tmp
  }
  result ← result + L + R
}
return result;
}
```

IV. Implementasi dan Uji Coba

A. Implementasi

Berikut adalah implementasi aplikasi enkripsi dalam bahasa C++. Parameter s adalah plaintext dan parameter key adalah kunci yang berukuran 2 byte (16-bit).

```
char[] Confidentia(char[] s, char[] key){
char[65536] result = “”;
for (int k=0; k<strlen(s)/4+ (strlen(s)%4 != 0); k++) {
    for (int i=0; i<4; i++) {
      c[i] = (~s[i+k*4]^key[i % strlen(key)]);
    }
    int sep = k*4 + 2;
     //separate block
    for (int i=0; i<4; i++) {
      if (i+k*4<sep) {
        LB[i] = c[i];
      } else {
        RB[i-sep] = c[i];
      }
    }
    for (int i=0; i<8; i++) {  // iterasi 8x
      char tmp[65536] = "";
```

```
        strcpy(tmp,RB);
        RB[0] = RB[1];
        unsigned char lookup = RB[0];
        RB[1] = s_box[lookup];
        for (int j = 0; j < strlen(RB); j++) {
           RB[j] = RB[j]^key[i % strlen(key)]^LB[j];
           LB[j] = tmp[j];
        }
      }
      strcat(result,LB);
      strcat(result,RB);
    }
  return result;
  }
```

B. Uji Coba Algoritma Enkripsi

Mula-mula program diuji dengan plainteks sebesar 8 byte dan kunci sepanjang 2 byte. Plainteks yang dimasukkan adalah “abcdefgh” dan kunci “12”.

Hasil yang didapat adalah sebuah teks yang tidak dapat dibaca, yaitu “ｳk6pn密k”.

Kemudian kunci dimodifikasi sedikit menjadi “13”, hasil yang didapat adalah “・%・u”. Terdapat perbedaan yang cukup berarti.

Uji coba berikutnya adalah dengan plainteks “abcdefghijkl” dan kunci “12”. Hasil yang didapat adalah “ｳk6pn密k寶2・”. Terlihat bahwa bagian plainteks abcdefgh dienkripsi dengan hasil yang sama dengan sebelumnya.

Uji coba berikutnya adalah mengganti 1 karakter pada plainteks “abcdefgh” menjadi “abcdefgi” Hasil yang didapat adalah “ｳk6pn密k”. Selain itu, jika plainteks masukan menjadi “accdefgh”, hasil yang didapat adalah “・P蕭7・”.

V. Analisis Data

Berdasarkan hasil uji coba yang dilakukan, terdapat bahwa algoritma ini bersifat blok independen, yang artinya enkripsi satu blok tidak mempengaruhi blok yang lain. Selain itu, penggantian 1 karakter pada plainteks dapat menyebabkan hasil yang berbeda jauh atau tidak berbeda jauh. Apabila kunci yang digunakan berbeda, hasil yang didapatkan akan berbeda jauh.

Panjang string cipher sama dengan string plainteks jika direpresentasikan dalam bit, sehingga tidak banyak menghasilkan tambahan ukuran data.

VI. Simpulan dan Saran

Algoritma block cipher Confidentia lebih tepat digunakan untuk data yang tidak terlalu rahasia, karena proses dekripsi yang mudah.

Jika iterasi dalam algoritma ini diperbanyak, maka algoritma ini akan menjadi lebih sulit dikriptanalisis.

References

[1] Slide Kuliah Algoritma Kriptografi Modern