Algoritma *Block Cipher* Pyramid

Kevin Erdiza Yogatama
13515016
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jl. Ganesha 10 Bandung 40132, Indonesia
13515016@std.stei.itb.ac.id

Marvin Jerremy Budiman
13515076
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jl. Ganesha 10 Bandung 40132, Indonesia
13515076@std.stei.itb.ac.id

***Abstrak*—Teknik kriptografi yang menggunakan prinsip *block cipher* merupakan teknik yang populer. Sudah banyak metode yang diusulkan dari tahun ke tahun, namun banyak juga yang sudah ditemukan kelemahannya. Di tulisan ini, disulkan sebuah algoritma *block cipher* baru yang bernama Pyramid. Algoritma ini menggunakan struktur Feistel yang dimodifikasi dan menggunakan S-box dengan struktur Pyramid. Keamanan algoritma ini juga didasari oleh komponen penting lain yang disebut Permutor dan pembangkit bilangan acak yang meningkatkan properti *diffusion* dan *confusion* dari algoritma ini.**

***Kata kunci—block cipher, confusion, diffusion,* jaringan Feistel.**

I. PENDAHULUAN

Penyembunyian pesan menjadi hal yang umum ditemui dalam kehidupan sehari-hari. Di era digital seperti sekarang ini, pesan digital dikirimkan melalui berbagai media, seperti saluran telepon dan Internet. Saluran-saluran tersebut merupakan saluran publik, sehingga rentan disadap oleh berbagai pihak. Penyembunyian pesan perlu dilakukan, agar menjaga kerahasiaan pesan yang dikirimkan melalui saluran-saluran tersebut.

Kriptografi merupakan ilmu yang mempelajari teknik-teknik menyembunyikan pesan, dengan cara menyandikan pesan tersebut, sehingga tidak dapat dimengerti oleh pihak-pihak lain, selain pihak pengirim dan penerima pesan yang sah [1, 2]. Sejak zaman Mesir kuno hingga sekarang, telah banyak algoritma kriptografi yang ditemukan. Seiring perkembangan zaman, orang-orang mulai menemukan kelemahan dari setiap algoritma yang telah ditemukan. Dengan adanya kelemahan-kelemahan tersebut, orang-orang berlomba-lomba untuk menciptakan algoritma kriptografi baru yang lebih aman, atau memperbaiki algoritma yang telah ada.

Kunci memegang peranan penting pada algoritma kriptografi. Berdasarkan kesamaan kunci, algoritma kriptografi dapat dikelompokkan menjadi 2 kelompok besar, yaitu algoritma kriptografi kunci simetri (kunci untuk enkripsi dan dekripsi sama) dan algoritma kunci nirsimetri (kunci untuk enkripsi dan dekripsi berbeda). Contoh algoritma kriptografi kunci simetri modern adalah DES dan 3DES, sedangkan contoh algoritma kriptografi kunci nirsimetri adalah RSA dan ElGamal.

Dua prinsip utama yang harus diterapkan pada algoritma kriptografi agar tidak mudah dipecahkan penyadap adalah prinsip *confusion* dan *diffusion*. Prinsip *confusion* menyatakan bahwa sebisa mungkin tidak boleh terdapat hubungan antara *plaintext, ciphertext,* dan kunci. *Substitution-box* (S-box) merupakan salah satu contoh realisasi prinsip *confusion*. S-box digunakan sebagai tabel substitusi untuk *plaintext*. Prinsip *diffusion* menyatakan bahwa perubahan pada satu bit informasi di *plaintext/ciphertext/*kunci menyebarkan pengaruh seluas mungkin kepada komponen yang lain.

Algoritma *block cipher* Pyramid merupakan algoritma *block cipher* yang memproses *block* dengan ukuran 64 bit. Kunci yang digunakan pada algoritma ini berukuran tetap, yaitu 64 bit. Algoritma ini dijalankan selama 20 *round*. Masing-masing *round* menerapkan permutasi, jaringan Feistel dan S-box.

Tiga komponen penting dalam algoritma ini adalah Pyramid, Permutor, dan pembangkit bilangan acak. Pyramid terdiri dari 10 S-box yang disusun menjadi piramida dengan 4 tingkat. Pada tingkat pertama terdapat 1 S-box, pada tingkat kedua terdapat 2 S-box, dan seterusnya. Substitusi pesan akan dilakukan pada setiap tingkat dari piramida. Permutor merupakan komponen yang digunakan untuk melakukan permutasi terhadap pesan. Permutasi dilakukan berdasarkan urutan tertentu yang telah dibuat sebelumnya. Pembangkit bilangan acak digunakan untuk membangkitkan susunan permutasi dan susunan piramida. Pada setiap *round* akan dibangkitkan kunci internal (*round key*) berdasarkan kunci eksternal. Kunci internal akan menjadi *seed* dari pembangkit bilangan acak, sehingga setiap round akan memiliki *seed* yang berbeda.

II. DASAR TEORI

A. *Confusion dan Diffusion*

Confusion dan *Diffusion* merupakan 2 properti yang dinilai untuk mengukur keamanan sebuah *cipher*. Properti tersebut diidentifikasikan oleh Claude Shannon pada laporannya yang berjudul *A Mathematical Theory of Cryptography* [4]. Dijelaskan bahwa properti *confusion* adalah dimana relasi antara hasil enkripsi dengan kunci sangat kompleks sampai terlihat seperti tidak ada relasi sama sekali. sedangkan properti *diffusion* adalah dimana jika ada perubahan sedikit pada

plainteks, cipherteks yang dihasilkan akan sangat jauh berbeda dan sebaliknya.

B. *Block Cipher*

Block cipher melakukan proses enkripsi dan dekripsi pada blok bit dengan ukuran tertentu. Sebagai contoh, pada algoritma DES, ukuran blok yang dioperasikan adalah 64 bit. Untuk menangani ukuran pesan yang bukan merupakan kelipatan dari ukuran blok, maka *padding* dilakukan pada pesan tersebut. *Padding* dapat dilakukan dengan cara menambahkan bit 0 pada blok terakhir, hingga ukuran pesan adalah kelipatan dari ukuran blok.

Block cipher dapat dioperasikan dalam berbagai mode. Terdapat 5 mode yang direkomendasikan oleh NIST[5]. Berikut adalah kelima mode tersebut:

1. *Electronic Codebook Mode*

   Pada mode ini, setiap blok dienkripsi secara terpisah dan tidak mempengaruhi satu sama lain. akibatnya, mode ini memiliki properti *diffusion* yang lemah. karena perubahan pada satu bagian plainteks hanya mempengaruhi di bagian tersebut saja pada cipherteks yang dihasilkan.

   Secara matematis, mode ini didefiniskan sebagai berikut:

   untuk enkripsi

   $$C_j = CIPH_K(P_j) \qquad \text{for } j = 1 \ldots n.$$

   untuk dekripsi

   $$P_j = CIPH^{-1}_K(C_j) \qquad \text{for } j = 1 \ldots n.$$

2. *Cipher Block Chaining Mode*

   Pada mode ini, setiap blok palinteks akan terlebih dahulu dioperasikan XOR dengan hasil cipherteks yang dihasilkan proses enkripsi blok sebelumnya, yang kemudian hasilnya dienkripsi untuk menghasilkan cipherteks. Blok pertama yang dienkripsi perlu melakukan XOR dengan nilai masukan yang disebut *Initialization Vector* (IV). IV yang diberikan membuat setiap pesan yang dienkripsi bisa unik.

   Secara matematis, mode ini didefiniskan sebagai berikut:

   untuk enkripsi

   $$C_1 = CIPH_K(P_1 \oplus IV);$$
   $$C_j = CIPH_K(P_j \oplus C_{j-1}) \qquad \text{for } j = 2 \ldots n.$$

   untuk dekripsi

   $$P_1 = CIPH^{-1}_K(C_1) \oplus IV;$$
   $$P_j = CIPH^{-1}_K(C_j) \oplus C_{j-1} \qquad \text{for } j = 2 \ldots n.$$

3. *Cipher Feedback Mode*

   Pada mode ini, yang dilakukan adalah memjadikan hasil cipherteks pada mode sebelumnya sebagai input di setiap proses enkripsi bloknya. cipherteks dihasilkan dengan operasi XOR hasil enkripsi cipherteks sebelumnnya dengan plainteks. pada mode ini juga dibutuhkan nilai IV sebagai input pertama untuk proses enkripsi pertama.

   Secara matematis, mode ini didefiniskan sebagai berikut:

   untuk enkripsi

   $$I_1 = IV;$$
   $$I_j = LSB_{b-s}(I_{j-1}) \mid C^{\#}_{j-1} \qquad \text{for } j = 2 \ldots n;$$
   $$O_j = CIPH_K(I_j) \qquad \text{for } j = 1, 2 \ldots n;$$
   $$C^{\#}_j = P^{\#}_j \oplus MSB_s(O_j) \qquad \text{for } j = 1, 2 \ldots n.$$

   untuk dekripsi

   $$I_1 = IV;$$
   $$I_j = LSB_{b-s}(I_{j-1}) \mid C^{\#}_{j-1} \qquad \text{for } j = 2 \ldots n;$$
   $$O_j = CIPH_K(I_j) \qquad \text{for } j = 1, 2 \ldots n;$$
   $$P^{\#}_j = C^{\#}_j \oplus MSB_s(O_j) \qquad \text{for } j = 1, 2 \ldots n.$$

4. *Output Feedback Mode*

   Pada mode ini, yang dilakukan setiap bloknya adalah melakukan proses enkripsi pada nilai yang hasil enkripsi yang diawali dengan nilai IV. setiap nilai yang dihasilkan di setiap bloknya, yang disebut *output*, akan dioperasikan XOR dengan blok plainteks yang akan menghasilkan cipherteks

   Secara matematis, mode ini didefiniskan sebagai berikut:

   untuk enkripsi

   $$I_1 = IV;$$
   $$I_j = O_{j-1} \qquad \text{for } j = 2 \ldots n;$$
   $$O_j = CIPH_K(I_j) \qquad \text{for } j = 1, 2 \ldots n;$$
   $$C_j = P_j \oplus O_j \qquad \text{for } j = 1, 2 \ldots n\text{-}1;$$
   $$C^*_n = P^*_n \oplus MSB_u(O_n).$$

   untuk dekripsi

   $$I_1 = IV;$$
   $$I_j = O_{j-1} \qquad \text{for } j = 2 \ldots n;$$
   $$O_j = CIPH_K(I_j) \qquad \text{for } j = 1, 2 \ldots n;$$
   $$P_j = C_j \oplus O_j \qquad \text{for } j = 1, 2 \ldots n\text{-}1;$$
   $$P^*_n = C^*_n \oplus MSB_u(O_n).$$

5. *Counter Mode*

   Pada mode ini, di setiap bloknya, melakukan proses enkripsi nilai dari hasil *counter*. hasilnya akan dioperasikan XOR dengan blok plainteks yang akan menghasilkan blok cipherteks. Untuk mencapai properti setiap blok sangat berbeda dengan setiap blok lainnya (properti *confusion*), *counter* yang digunakan harus menghasilkan nilai yang berbeda untuk setiap bloknya. banyak fungsi matematis yang bisa digunakan sebagai *counter*, salah satunya adalah *increment-by-one counter*, yang merupakan *counter*

yang paling populer digukanan walaupun merupakan yang paling sederhana.

Secara matematis, mode ini didefiniskan sebagai berikut:

untuk enkripsi

$O_j = CIPH_K(T_j)$ for $j = 1, 2 \ldots n$;
$C_j = P_j \oplus O_j$ for $j = 1, 2 \ldots n\text{-}1$;
$C^*_n = P^*_n \oplus MSB_u(O_n)$.

untuk dekripsi

$O_j = CIPH_K(T_j)$ for $j = 1, 2 \ldots n$;
$P_j = C_j \oplus O_j$ for $j = 1, 2 \ldots n\text{-}1$;
$P^*_n = C^*_n \oplus MSB_u(O_n)$.

C. *Jaringan Feistel*

Jaringan Fesitel merupakan sebuah struktur simetri operasi pembentukan *block cipher*. Struktur ini didesain oleh Horst Feistel yang pertama kali digunakan pada sebuah cipher bernama Lucifer. Secara matematis, struktur jaringan Feistel bisa didefiniskan sebagai berikut:

Untuk enkripsi

$$L_{i+1} = R_i$$
$$R_{i+1} = L_i \oplus \mathrm{F}(R_i, K_i).$$

Untuk dekripsi

$$R_i = L_{i+1}$$
$$L_i = R_{i+1} \oplus \mathrm{F}(L_{i+1}, K_i).$$

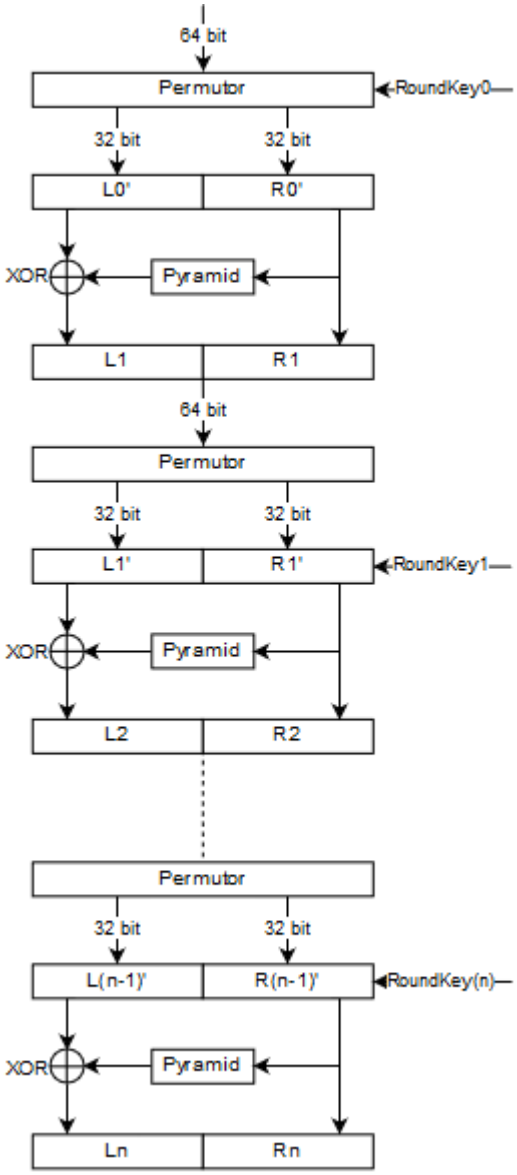
D. *S-Box*

S-Box adalah komponen dasar pada proses kriptografi simetris yang melakukan substitusi. *S-box* di desain sedemikian rupa untuk mencapai properti *confusion* yang bagus.

III. Proposal Algoritma

A. *Algoritma Secara Umum*

Algoritma Pyramid dirancang untuk melakukan *cipher* berulang sebanyak 20 kali (20 *round*). Setiap iterasi terdiri dari beberapa tahap. Tahap pertama adalah permutasi pesan menggunakan Permutor. Pesan yang telah mengalami permutasi dimasukkan ke tahap berikutnya, yaitu jaringan Feistel, yang didalamnya terdapat Pyramid. Skema dari algoritma Pyramid ditunjukkan oleh Gambar 1.



Gambar 1. Skema Algoritma Pyramid

Algoritma diimplementasikan sebagai sebuah fungsi yang menerima input berupa *array of bytes* berukuran 8 (64 bit). Untuk enkripsi terdapat beberapa operasi yang dilakukan pada setiap round. Operasi tersebut akan diulangi sesuai jumlah *round*. Berikut ini adalah urutan operasi enkripsi dalam *round* 0 pada algoritma Pyramid:

1. Permutor membangkitkan tabel permutasi menggunakan RoundKey0.
2. Permutor akan melakukan permutasi terhadap nilai masukan, sehingga urutan byte menjadi acak.
3. Nilai yang telah dipermutasi kemudian dibagi menjadi 2 bagian, masing-masing berukuran 4 byte.
4. Pyramid dipermutasi menggunakan RoundKey0.

5. Kelompok 4 byte paling kanan (R0') akan disubstitusikan ke dalam Pyramid.
6. Operasi XOR dilakukan antara 4 byte paling kiri (L0') dengan hasil substitusi.
7. Hasil XOR dan R0' akan digabungkan kembali menjadi *array of bytes* berukuran 8 (L1 dan R1).
8. Lakukan operasi 1 sampai 6, hingga mencapai *round* n.

Untuk dekripsi terdapat beberapa operasi yang diubah urutannya. RoundKey yang digunakan dimulai dari RoundKey(n) hingga RoundKey0. Berikut ini adalah urutan operasi dekripsi dalam *round* n pada algoritma Pyramid:

1. Nilai masukan dibagi menjadi 2 bagian, masing-masing berukuran 4 byte.
2. Pyramid dipermutasi menggunakan RoundKey(n).
3. Kelompok 4 byte paling kanan (R(n)) akan disubstitusikan ke dalam Pyramid.
4. Operasi XOR dilakukan antara 4 byte paling kiri (L(n)) dengan hasil substitusi.
5. Hasil XOR dan R(n) akan digabungkan kembali menjadi *array of bytes* berukuran 8 (L(n-1)' dan R(n-1)').
6. Permutor membangkitkan tabel permutasi menggunakan RoundKey(n).
7. Permutor akan melakukan invers permutasi terhadap nilai masukan, sehingga urutan byte berubah.
8. Lakukan operasi 1 sampai 6, hingga mencapai *round* 0.

B. *Permutor*

Permutor akan membangkitan tabel permutasi yang akan digunakan untuk mengubah urutan elemen dalam *array of bytes* yang menjadi masukan algoritma. Isi dari tabel permutasi dibangkitkan oleh pembangkit bilangan acak, menggunakan *round key* sebagai *seed* untuk pembangkit bilangan acak. Berikut ini merupakan contoh isi dari tabel permutasi.

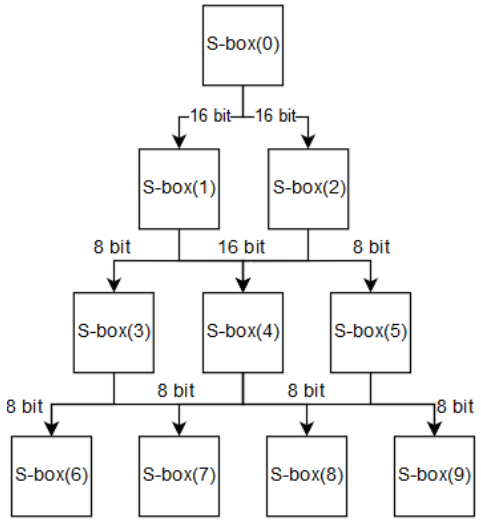
Tabel 1. Contoh Isi Tabel Permutasi

Indeks	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Isi	2	4	6	0	3	7	1	5

Makna dari Tabel 1 adalah sebagai berikut. Byte pada urutan 0 di *array* lama, akan ditempatkan pada urutan 2 di *array* yang baru. Byte pada urutan 1 di *array* lama, akan ditempatkan pada urutan 4 di *array* yang baru, dan seterusnya.

C. *Pyramid*

Pyramid merupakan struktur yang terdiri dari 10 S-box. S-box tersebut disusun menjadi 4 tingkat. Pada tingkat pertama terdapat S-box(0), pada tingkat kedua terdapat S-box(1) dan S-box(2), pada tingkat ketiga terdapat S-box(3), S-box(4), dan S-box(5), dan pada tingkat keempat terdapat S-box(6), S-box(7), S-box(8), dan S-box(9). Gambar 2 menunjukkan susunan dari Pyramid.



Gambar 2. Susunan Pyramid

Pada tingkat pertama, S-box(0) menerima masukan berupa *array of bytes* berukuran 4 (4 bytes = 32 bit). Kemudian hasil substitusi akan dibagi 2, masing-masing 2 bytes (16 bit). Pada tingkat kedua, 2 bytes pertama disubstitusi dengan S-box(1) dan 2 bytes sisanya disubstitusi dengan S-box(2). Pada tingkat ketiga *array of bytes* dibagi menjadi 3 kelompok, masing-masing 1 byte (masuk S-box(3)), 2 bytes (masuk S-box(4)), dan 1 byte (masuk S-box(5)). Pada tingkat keempat *array of bytes* dibagi menjadi 4 kelompok, masing-masing berukuran 1 byte (8 bit), dan masing-masing menjadi masukan S-box(6), S-box(7), S-box(8), dan S-box(9).

Permutasi dilakukan pada susunan Pyramid di setiap *round*, sehingga susunan Pyramid di setiap *round* menjadi berbeda. RoundKey menjadi *seed* dari permutasi Pyramid pada setiap *round*.

D. *Kunci Eksternal dan Pembangkitan Kunci Internal*

Pada algoritma ini digunakan 2 macam kunci, yaitu kunci eksternal dan kunci internal. Kunci eksternal berfungsi untuk membangkitkan isi dari setiap S-box di dalam Pyramid. Kunci eksternal yang berukuran 64 bit dikonversi menjadi bilangan bulat, kemudian menjadi *seed* untuk membangkitkan isi dari S-box dalam Pyramid.

Kunci internal (*round key*) untuk setiap *round* berupa bilangan bulat, yang dibangkitkan menggunakan pembangkit bilangan acak. Total kunci internal seluruhnya adalah 20. *Seed* yang digunakan untuk membangkitkan kunci internal adalah nilai bilangan bulat dari kunci eksternal. Fungsi kunci internal adalah untuk membangkitkan tabel permutasi pesan dan tabel permutasi Pyramid.

IV. SIMULASI DAN HASIL

Simulasi dilakukan untuk mengetahui hasil enkripsi dan dekripsi yang dilakukan menggunakan algoritma Pyramid. Simulasi dilakukan dengan berbagai mode *block cipher*, yaitu ECB, CBC, CFB, OFB, dan mode *counter*. Kunci eksternal

yang digunakan untuk simulasi adalah “namasaya”. *Plaintext* yang akan dienkripsi adalah sebagai berikut.

```
   While the first category is impossible to eradicate
completely, we can create both laws and code to
minimize this behaviour, just as we have always done
offline. The second category requires us to redesign
systems in a way that changes incentives. And the final
category calls for research to understand existing
systems and model possible new ones or tweak those we
already have.

   You can’t just blame one government, one social
network or the human spirit. Simplistic narratives risk
exhausting our energy as we chase the symptoms of these
problems instead of focusing on their root causes. To
get this right, we will need to come together as a
global web community.

   At pivotal moments, generations before us have
stepped up to work together for a better future. With
the Universal Declaration of Human Rights, diverse
groups of people have been able to agree on essential
principles. With the Law of the Sea and the Outer Space
Treaty, we have preserved new frontiers for the common
good. Now too, as the web reshapes our world, we have a
responsibility to make sure it is recognised as a human
right and built for the public good. This is why the
Web Foundation is working with governments, companies
and citizens to build a new Contract for the Web.
```

Selain itu akan dilakukan juga pengujian berupa pengubahan 1 bit pada *plaintext* (huruf pertama *plaintext* akan diubah menjadi “V”) dan dan 1 bit pada kunci (kunci akan diubah menjadi “mamasaya”).

A. Mode ECB

Ciphertext yang dihasilkan menggunakan algoritma Pyramid dengan mode ECB adalah sebagai berikut.

```
924e 2f12 3804 1244 fdfe 4648 8b94 a24f
6ed8 5365 cd46 8e0c b5e0 64a3 9e17 dae4 ...
```

Dekripsi memberikan hasil yang sama dengan *plaintext*.

Hasil pengujian dengan mengubah 1 bit *plaintext* adalah sebagai berikut.

```
ecdc afe7 377e 642b fdfe 4648 8b94 a24f
6ed8 5365 cd46 8e0c b5e0 64a3 9e17 dae4 ...
```

Hasil pengujian dengan mengubah 1 bit kunci adalah sebagai berikut.

```
d5f4 a6d2 9a8b 750f b942 5872 d554 52c3
fdcc 5fb3 1570 c298 e5f2 d42f 9fff 37ff ...
```

Dari hasil pengujian tersebut didapati bahwa perubahan 1 bit di *plaintext*, mengubah 64 bit di *ciphertext* dan perubahan 1 bit di kunci, mengubah keseluruhan *ciphertext*. Prinsif *diffusion* direalisasikan dengan baik pada mode ECB.

B. Mode CBC

Ciphertext yang dihasilkan menggunakan algoritma Pyramid dengan mode CBC adalah sebagai berikut.

```
40c7 0d19 9956 c497 6bee bea3 7216 e7c5
cf56 91b1 6f16 0393 a0fd baed 1635 e98b ...
```

Dekripsi memberikan hasil yang sama dengan *plaintext*.

Hasil pengujian dengan mengubah 1 bit *plaintext* adalah sebagai berikut.

```
1de4 e2db 2ac8 3520 1aff e176 2b31 ae3e
3b20 1cf4 405a 8fc8 3263 7cc8 5319 2e6f ...
```

Hasil pengujian dengan mengubah 1 bit kunci adalah sebagai berikut.

```
f8ae c8fd d40d 27f1 92a5 c829 e255 8df6
276e ac52 ed7e 8fee a47d afd6 5b9a 8f1b ...
```

Dari hasil pengujian tersebut didapati bahwa perubahan 1 bit di *plaintext*, mengubah keseluruhan *ciphertext* dan perubahan 1 bit di kunci, mengubah keseluruhan *ciphertext*. Prinsif *diffusion* direalisasikan dengan baik pada mode CBC.

C. Mode CFB

Ciphertext yang dihasilkan menggunakan algoritma Pyramid dengan mode CFB adalah sebagai berikut.

```
5e31 3ddf eee2 70ef 4c71 6d2b 3707 aab2
b480 8f3d c0ff 0af3 0372 b9e0 3088 b47d ...
```

Dekripsi memberikan hasil yang sama dengan *plaintext*.

Hasil pengujian dengan mengubah 1 bit *plaintext* adalah sebagai berikut.

```
5f31 3ddf eee2 70ef cba5 edcd ada1 60a2
764a 641d b3ea c04e 8ec3 3178 cf6c e8cf ...
```

Hasil pengujian dengan mengubah 1 bit kunci adalah sebagai berikut.

```
a1d1 22a3 ebdd 2ed7 f6a3 3203 fcfb 9361
2a30 9284 f826 c9d5 6834 2549 204b bcbb ...
```

Dari hasil pengujian tersebut didapati bahwa perubahan 1 bit di *plaintext*, mengubah hampir keseluruhan *ciphertext* dan perubahan 1 bit di kunci, mengubah keseluruhan *ciphertext*. Prinsif *diffusion* direalisasikan dengan baik pada mode CFB.

D. Mode OFB

Ciphertext yang dihasilkan menggunakan algoritma Pyramid dengan mode OFB adalah sebagai berikut.

```
5e31 3ddf eee2 70ef 8291 a809 ee33 6cb0
57e8 4bee 42cd a708 c916 1744 f4af 74f5 ...
```

Dekripsi memberikan hasil yang sama dengan *plaintext*.

Hasil pengujian dengan mengubah 1 bit *plaintext* adalah sebagai berikut.

```
5f31 3ddf eee2 70ef 8291 a809 ee33 6cb0
57e8 4bee 42cd a708 c916 1744 f4af 74f5 ...
```

Hasil pengujian dengan mengubah 1 bit kunci adalah sebagai berikut.

```
a1d1 22a3 ebdd 2ed7 173a 4e79 acfc b12d
2e39 2400 c2a3 60f5 0cf0 503a ca91 a9ba ...
```

Dari hasil pengujian tersebut didapati bahwa perubahan 1 bit di *plaintext*, mengubah hanya 1 byte *ciphertext* dan perubahan 1 bit di kunci, mengubah keseluruhan *ciphertext*. Prinsif *diffusion* tidak bisa direalisasikan dengan baik pada mode OFB.

E. Mode Counter

Ciphertext yang dihasilkan menggunakan algoritma Pyramid dengan mode *counter* adalah sebagai berikut.

```
3c4a 96b5 29f4 892a f7a2 5357 cff2 b8c3
b2f2 025f 2702 0bcd 70b6 5a19 8639 9068 ...
```

Dekripsi memberikan hasil yang sama dengan *plaintext*.

Hasil pengujian dengan mengubah 1 bit *plaintext* adalah sebagai berikut.

```
3d4a 96b5 29f4 892a f7a2 5357 cff2 b8c3
b2f2 025f 2702 0bcd 70b6 5a19 8639 9068 ...
```

Hasil pengujian dengan mengubah 1 bit kunci adalah sebagai berikut.

```
d006 5d69 1f91 4273 116b 3e4d 31d3 a8c5
38f0 0283 20c2 bd81 2abc 12a6 30e5 c821 ...
```

Dari hasil pengujian tersebut didapati bahwa perubahan 1 bit di *plaintext*, mengubah hanya 1 byte *ciphertext* dan perubahan 1 bit di kunci, mengubah keseluruhan *ciphertext*. Prinsif *diffusion* tidak bisa direalisasikan dengan baik pada mode *counter*.

V. Analisis Keamanan

A. Serangan Brute Force

Keamanan sebuah *cipher* terhadapa *brute-force attack* bergantung pada jumlah kemungkinan kunci yang mungkin digunakan. Kunci yang digunakan pada *block cipher* Pyramid terdiri dari 64 bit nilai biner yang menghasilkan 2^{64} kemungkinan kunci. Jika sebuah percobaan *brute-force* menggunakan mesin yang memiliki kemampuan melakukan 10^{12} operasi per detik, maka waktu yang dibutuhkan untuk mencoba semua kemungkinan adalah 18446744 detik atau sekitar 213 hari.

B. Analisis Confusion dan Diffusion

Prinsip *confusion* berhasil diterapkan oleh algoritma Pyramid. Hal ini dibuktikan dari *ciphertext* yang dihasilkan oleh proses enkripsi, selalu berbeda dan tidak berhubungan dengan *plaintext*. Sedangkan prinsip *diffusion* diterapkan dengan baik ketika 1 bit kunci diubah. Hal ini dibuktikan dengan hasil *ciphertext* yang seluruhnya berbeda, jika dibandingkan dengan *ciphertext* sebelumnya. Lain halnya dengan perubahan pada 1 bit *plaintext*. Pada mode OFB dan mode *counter*, perubahan ini hanya mengubah beberapa byte dari *ciphertext*. Karena itu prinsip *diffusion* tidak tercapai ketika 1 bit *plaintext* diubah.

VI. Kesimpulan dan Saran

Hasil simulasi menunjukkan bahwa algoritma Pyramid memberikan hasil enkripsi yang baik pada beberapa mode tertentu (ECB, CBC, dan CFB). Pada mode tersebut, prinsip *confusion* dan *diffusion* berhasil dicapai. Dari hasil analisis serangan *brute force*, didapati pula bahwa untuk memecahkan *ciphertext* hasil algoritma Pyramid dibutuhkan waktu yang cukup lama. Untuk pengembangan selanjutnya, algoritma perlu diperbaiki agar mode OFB dan mode counter juga memberikan hasil enkripsi yang baik.

Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa, karena atas bantuan, rahmat, dan berkat-Nya, makalah ini dapat selesai. Penulis ingin menyampaikan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir selaku dosen mata kuliah IF4020 Kriptografi yang telah membagikan ilmunya kepada penulis. Selain itu, penulis juga ingin menyampaikan terima kasih kepada kedua orang tua yang selalu mendukung penulis.

Referensi

[1] R. Munir, *Diktat Kuliah IF4020 Kriptografi*, Departemen Teknik Informatika Insttitut Teknologi Bandung, 2005.

[2] R. Munir, *Slide Kuliah IF4020 Kriptografi*, Pengantar Kriptografi, 2019.

[3] R. Munir, *Slide Kuliah IF4020 Kriptografi*, Kriptografi Modern, 2019.

[4] C. Shannon, A Mathematical Theory of Cryptography, 1945

[5] M. Dworkin, Recommendation for Block Cipher Modes of Operation. Methods and Techniques, 2001.