KUIS 3 KRIPTOGRAFI DAN KEAMAAN SIBER
IMPLEMENTASI DAN ANALISIS ENKRIPSI DATA ENCRYPTION STANDARD (DES)

Disusun Oleh:

Bambang Istijab 105222007

FAKULTAS SAINS DAN ILMU KOMPUTER
PROGRAM STUDI ILMU KOMPUTER
UNIVERSITAS PERTAMINA
2025

I. Definisi

Data Encryption Standard (DES) adalah cipher blok simetris. Yang kami maksud dengan 'simetris' adalah ukuran teks masukan dan teks keluaran (ciphertext) sama (64-bit). 'Blok' di sini berarti ia mengambil sekelompok bit bersama-sama sebagai masukan alih-alih mengenkripsi teks sedikit demi sedikit. Data encryption standard (DES) telah ditemukan rentan terhadap serangan yang sangat kuat dan oleh karena itu, digantikan oleh Advanced Encryption Standard (AES) (GeekforGeeks, 2021)

II. Tahapan Enkripsi

a. Konversi Plainteks dan Kunci ke Bit

Plainteks (dalam program ini yaitu ”COMPUTER”) dikonversi ke representasi biner 64-bit. Setiap karakter ASCII diubah menjadi 8-bit biner, contoh output pada program:

```
Plaintext: COMPUTER
Binary Plaintext:
0100001101001111010011010101000001010101010101000100010101010010
```

Begitu juga dengan kunci 64-bit dikonversi menjadi bit array

b. Initial Permutation (IP)

Blok plaintext 64-bit mengalami permutasi awal menggunakan **tabel IP** standar DES.

```
IP = [58, 50, 42, 34, 26, 18, 10, 2,
      60, 52, 44, 36, 28, 20, 12, 4,
      62, 54, 46, 38, 30, 22, 14, 6,
      64, 56, 48, 40, 32, 24, 16, 8,
      57, 49, 41, 33, 25, 17, 9, 1,
      59, 51, 43, 35, 27, 19, 11, 3,
      61, 53, 45, 37, 29, 21, 13, 5,
      63, 55, 47, 39, 31, 23, 15, 7]
```

Gambar 1 tabel IP standar DES 1

Permutasi ini menyebarkan bit asli ke posisi yang telah ditentukan agar memperbesar efek difusi (spreading bit influence). Contoh output pada program:

```
After Initial Permutation:
1111111110111000011101100101011100000000000000000000011010000011
```

c. Pembagian menjadi L dan R

Setelah permutasi awal, data dibagi menjadi dua bagian:

1. **L0 (Left)**: 32 bit kiri
2. **R0 (Right)**: 32 bit kanan

DES menggunakan struktur **Feistel**, sehingga hanya R yang dimodifikasi pada setiap ronde dan hasilnya di-XOR dengan L.

d. 16 Ronde Feistel (Iteratif)

DES menggunakan **16 ronde proses enkripsi**, di mana setiap ronde melibatkan:

1. Ekspansi (E): R diperluas dari 32 menjadi 48 bit menggunakan tabel E.

```
E = [32, 1, 2, 3, 4, 5, 4, 5,
     6, 7, 8, 9, 8, 9, 10, 11,
     12, 13, 12, 13, 14, 15, 16, 17,
     16, 17, 18, 19, 20, 21, 20, 21,
     22, 23, 24, 25, 24, 25, 26, 27,
     28, 29, 28, 29, 30, 31, 32, 1]
```

Gambar 2 Tabel E

2. XOR dengan Kunci: R hasil ekspansi di-XOR dengan subkey 48-bit yang unik untuk tiap ronde.
3. S-Box Substitution: Hasil XOR dibagi menjadi 8 blok 6-bit, masing-masing masuk ke 8 S-box yang mengubah 6-bit → 4-bit.
4. Permutasi P: Gabungan 32-bit hasil S-box dipermutasi ulang untuk penyebaran bit lebih lanjut.
5. XOR dengan L: Hasil permutasi di-XOR dengan L sebelumnya untuk membentuk R baru.
6. Swap: L baru menjadi R lama, dan R baru hasil dari proses di atas.

Setiap langkah ini ditampilkan dioutput, contoh output pada program:

```
-- Round 1 --
  Feistel input R: 00000000000000000000011010000011
  After Expansion: 100000000000000000000000000000011010100000000110
  After XOR with Key: 101100010011001000110011001101001110000100110000
  After S-Box Substitution: 01101111111010011011000111010011
  After P Permutation: 11100011000110111110101111010101
  L: 00000000000000000000011010000011
  R: 00011100101000111001110110000010

  :
  :
  :
  :
  :

-- Round 16 --
  Feistel input R: 11111111010011111000110010100111
  After Expansion: 111111111110101001011111110001011001010100001111
```

```
After XOR with Key: 1100111011011000011011001111000110100000000111001
After S-Box Substitution: 10010000010101100110010011100101
After P Permutation: 00000100110101010001110010101011
L: 11111111010011111000110010100111
R: 01101011110011000001001101001111
```

e. Swap Final dan Final Permutation (FP)
Setelah 16 ronde selesai:

1. Bagian kiri dan kanan digabungkan (namun di-swap lebih dulu).
2. Gabungan 64-bit akhir ini diproses menggunakan **Final Permutation (FP)** berdasarkan tabel FP standar DES.

Hasilnya adalah **ciphertext 64-bit**, contoh output pada program:

```
Before Final Permutation:
0110101111001100000100110100111111111111010011111000110010100111

After Final Permutation:
1110011111100111101110111111100110000100110000101111000110011010
```

f. Output Hasil Enkripsi

Output dari program akan menampilkan ciphertext dalam tiga bentuk yaitu ASCII, Hexadecimal dan Binary(64-bit), berikut contoh output pada program:

```
Ciphertext (ASCII): çç»ù„ Âñš
Ciphertext (hex): c3a7c3a7c2bbc3b9c284c382c3b1c29a
Ciphertext (binary): 1110011111100111101110111111100110000100110000101111000110011010
```

III. Kesimpulan

Melalui implementasi DES secara manual ini, diperoleh pemahaman mendalam mengenai cara kerja algoritma kriptografi blok klasik. Walaupun kompleks, setiap tahapan seperti permutasi, substitusi, dan transformasi bit dapat dijelaskan dan divisualisasikan dengan jelas. Adapun beberapa kesimpulan penting, sebagai berikut:

A. DES merupakan pondasi penting dalam dunia kriptografi simetris.
B. Struktur Feistel memungkinkan proses dekripsi dilakukan dengan algoritma yang sama seperti enkripsi (dengan urutan kunci dibalik).
C. Implementasi manual memberikan wawasan nyata terhadap konsep *bit-level transformation* yang tidak terlihat ketika menggunakan library siap pakai.

D. DES saat ini sudah dianggap tidak aman untuk keperluan industri, namun tetap sangat bermanfaat untuk pembelajaran.

IV. Referensi

GeeksforGeeks. (2021, June 10). Data Encryption Standard (DES) | Set 1. https://www.geeksforgeeks.org/data-encryption-standard-des-set-1/

V. Source Code

https://github.com/bmbng09/Kriptografi-dan-Keamanan-Siber.git