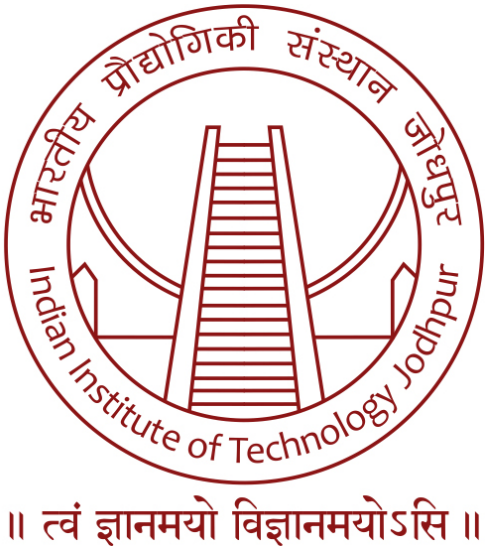
Quantum Cryptography and Coding

Report On Feistel Cipher- DES(Data Encryption Standard)- Implementation.



Name: **SUMIT KUMAR**
Roll Number: **(M23IQT007)**
Program: **MTech(QUANTUM TECHNOLOGIES)**

The Feistel Cipher Structure

The Feistel cipher is a specific structure and approach for implementing block ciphers, which are encryption algorithms that operate on fixed-size blocks of plaintext data. The fundamental characteristics of the Feistel cipher design include:

- The plaintext is divided into two equal-length halves
- Multiple rounds of processing are performed, typically 16 rounds
- In each round, one half of the data undergoes a substitution cipher based on the round key
- The other half is circularly shifted or permuted
- The two halves are then swapped before the next round

This Feistel structure introduces confusion and diffusion into the cipher, desirable properties that help increase the complexity and resistance to cryptanalysis. The multiple rounds and mechanisms like substitution and permutation thoroughly obscure the relationship between the plaintext and ciphertext.

The Data Encryption Standard (DES)

DES is a widely used symmetric-key block cipher that was adopted as a standard in 1977. It implements the Feistel cipher structure while incorporating specific functions for key scheduling, substitutions (S-boxes), permutations, and other transformations. Some key points about DES:

- 64-bit plaintext input with 56-bit key (8 bits for parity)
- 16 rounds of processing based on the Feistel structure
- Uses 8 different S-boxes (substitution boxes) to introduce non-linearity
- Expansion permutation to expand 32-bit half to 48 bits before XOR with round key
- Key schedule generates 16 48-bit round keys from the 56-bit master key

The high-level process is:

1. Initial permutation of plaintext
2. 16 rounds of:
    - Expansion permutation on right half
    - XOR with round key
    - Substitution via S-boxes
    - Permutation
    - XOR with left half
    - Swap left and right halves
3. Final permutation to produce ciphertext

While revolutionary at the time, advances in computing power and cryptanalysis techniques like differential and linear cryptanalysis have made DES vulnerable to brute-force attacks due to its limited 56-bit key size. It has since been replaced by the Advanced Encryption Standard (AES).

Implementation

Steps for Single-Round DES

Here are some additional details on the steps involved in a single round of DES encryption:

In each round, the 64-bit data is divided into two 32-bit halves, left (L) and right (R). The rounds process these halves as follows:

1. The 32-bit right half (R) is first expanded from 32 bits to 48 bits using an expansion permutation function that duplicates and permutes the bits.

2. The 48-bit expanded value is XORed with the 48-bit round key for the current round.

3. The result is then divided into eight 6-bit portions which are used as inputs to eight different substitution boxes or S-boxes.

4. Each 6-bit input to an S-box is substituted with a 4-bit output value according to a predefined substitution mapping in the S-box lookup table.

5. The eight 4-bit S-box outputs are then concatenated into a 32-bit value.

6. This 32-bit value undergoes a permutation based on a fixed permutation mapping.

7. The permuted 32-bit value is XORed with the left half (L) from the start of the round.

8. The two halves (L, R) are swapped, so the new right half is what was just calculated, and the new left half is the original right half.

This completes one round, and the process repeats for 16 rounds total, using a different 48-bit round key for each round generated from the initial 56-bit key.

The substitution via S-boxes and the complex permutations/expansions introduce the critical confusion and diffusion needed to make the cipher cryptographically strong.

With this additional context on the inner workings of a DES round, the full report covers the overarching Feistel structure, the DES algorithm itself, cryptanalysis techniques like differential and linear cryptanalysis that exposed some weaknesses, and the importance of these pioneering works in the evolution of modern block cipher design.

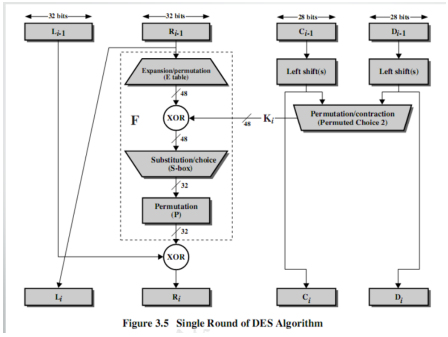


Figure 3.5 Single Round of DES Algorithm

Cryptanalysis of DES

Two important cryptanalytic techniques against DES are differential and linear cryptanalysis:

1. **Differential Cryptanalysis:** This analyzes how differences in plaintext pairs propagate through multiple rounds, looking for patterns that reveal information about the secret key. By carefully selecting pairs of plaintexts and tracking the evolution of differences, cryptanalysts can derive the round keys.

2. **Linear Cryptanalysis:** This exploits linear approximations of the nonlinear components in DES like the S-boxes. By combining multiple linear approximations, cryptanalysts can derive a linear expression involving key bits that holds with a known probability. This can then be used to recover the key more efficiently than brute force.

   While groundbreaking, these attacks are still not as efficient as a full brute-force attack against the 56-bit DES key. However, they demonstrated vulnerabilities in the cipher's design and paved the way for the development of more secure ciphers like AES. The principles behind the Feistel cipher and analyses like differential and linear cryptanalysis have been instrumental in advancing the understanding and design of robust block ciphers used ubiquitously today. e key more efficiently than brute force.

Reference:- : Study and Analysis of Symmetric Key-Cryptograph DES, Data Encryption Standard- Bhargavi Goswami