Listen

# The Data Encryption Standard (DES)

An earlier module investigated the Hill cipher, a block cipher simple enough to not require a computer. Other widely used block ciphers are the International Data Encryption Algorithm (IDEA) and the Data Encryption Standard (DES). The DES is the primary focus on this module. The concepts of diffusion and confusion, which were defined in the previous module regarding block ciphers, are the key quantifications of the randomness in the cipher text. Feistel functions are the main practical implementations of these two concepts in the DES.

DES is a variant of the Lucifer cipher, which was developed by Horst Feistel while he was an IBM employee in the 1970s. This cipher variant was submitted to the National Bureau of Standards (now the National Institute for Standards and Technology) in response to a solicitation for a nonmilitary cryptographic algorithm that would become the national standard. After the National Security Agency suggested modifications, various organizations adopted the algorithm. Most importantly, the major banks and the U.S. Treasury adopted it for encrypting unclassified material, and DES quickly became the industry standard. Over time, as computing power advanced, DES was no longer secure; by 1997, it could be attacked by brute force. DES was replaced with the Advanced Encryption Standard (AES) in 2000, although DES continues to be a widely used algorithm.

Feistel functions play a crucial role in the process of encrypting and decrypting data in DES. The Feistel network is a cryptographic structure that provides both confusion and diffusion, key principles in achieving strong encryption. DES operates on 64-bit blocks of data and uses 16 rounds of Feistel network transformations. Each round of DES consists of two main components: the Feistel function and the key schedule.

The Feistel function takes a 32-bit half-block of data as input and a 48-bit round subkey. It applies several operations, including expansion, substitution, permutation, and XOR, to the input data and the round subkey. First, the 32-bit input is expanded to 48 bits, making it the same size as the round subkey. Then, a bitwise XOR operation is performed between the expanded data and the subkey. The result is passed through an S-box substitution layer, where 48 bits are compressed to 32 bits. Next, a fixed permutation, known as the P-box, is applied to the data. Finally, the result is XORed with the other half of the 64-bit data block. This half-swapping and manipulation provide confusion and diffusion, making it extremely challenging for an attacker to reverse the encryption process without knowledge of the secret key.

Overall, the Feistel function in DES ensures that the data undergoes a complex series of transformations in each round, combining both the input data and the round subkey in a way that enhances the security of the encryption process.