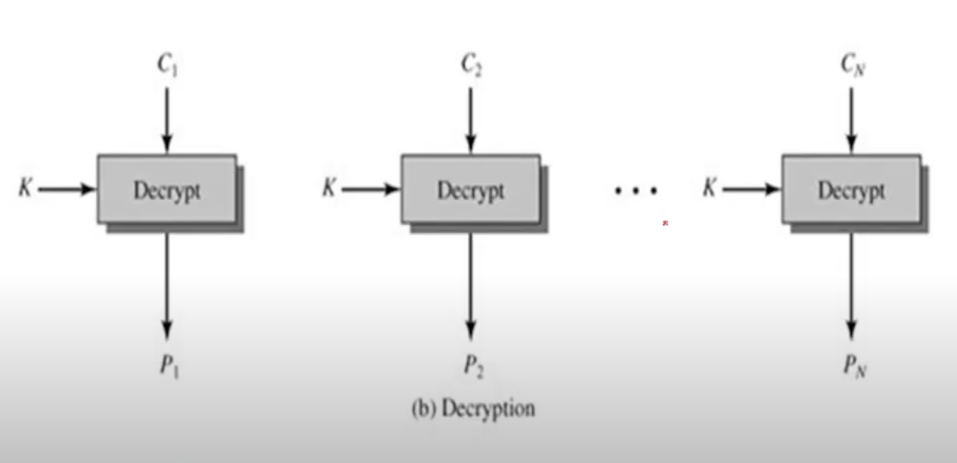
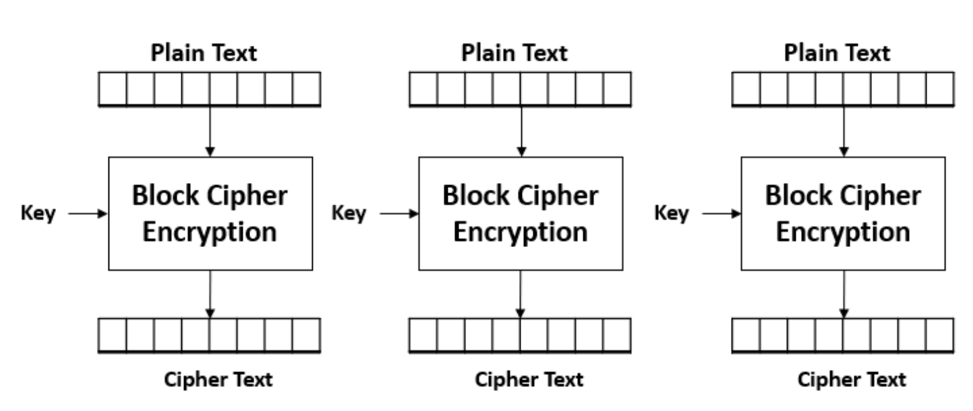
**Data Encryption Standard (DES) Algorithm**

**Overview**

The DES algorithm is a block cipher algorithm that takes 64 bits of data at a time and encrypts them. It is a symmetric algorithm wherein it utilizes the same key for both encryption and decryption. Its original usage utilized the Electronic Code Book (ECB) scheme.

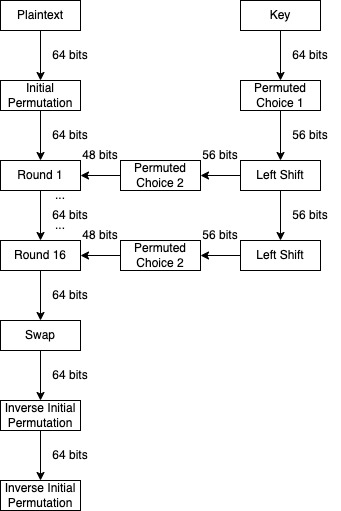
**Electronic Code Book Scheme**

****

*Figure 1. ECB from [][]*

The ECB scheme encrypts/decrypts plaintext in parallel. However, it requires for each block cipher to be equal in length of 64 bits. As such, block padding is sometimes necessary to ensure that the plaintext to be encrypted follows this requirement. This is the process of adding in trailing zeroes to fill in the remaining bits.

**DES Algorithm**



*Figure 2. DES Algorithm*

The algorithm begins with taking plaintext and applying an initial permutation. Afterwards, it will, go through 16 rounds of various operations that will be further expounded later. Then, the 32-bit halves of the data will be swapped. Finally, an inverse initial permutation will be applied to get the resulting ciphertext.

**Key Scheduling Algorithm**

The DES algorithm utilizes a 64-bit key, which is then converted to 56 bits. The left and right half of the key is then obtained and circularly left shifted appropriately. The shift value depends on the rounds in particular. For rounds 1, 2, 9, 16, you shift one (1) time. For the remaining rounds, you shift twice. After shifting, the bits are permuted to 48 bits, which then is referred to a round key.

**Rounds**

In the rounds, both halves of the data are obtained and denoted by L and R. For each round, the above equation gets their corresponding values. The left half is simply the value of the previous round’s right half. For the right half, it is the result of the XOR operation for the previous round’s left half and the result from the mangler function given the previous round’s right half, and the current round key.

**Mangler Function**

The mangler function consists of a few operations done on the bits of the data. First, the data is expanded through permutation and then an XOR operation is done with the key. Afterwards, substitution is done, and finally transposition is applied.