SYMMETRIC CRYPTOGRAPHY ALGORITHMS

The cryptographic strength of a symmetric key algorithm refers to its ability to resist attacks and protect the confidentiality of the information it is used to encrypt. The cryptographic strength of a symmetric key algorithm is determined by a variety of factors, including

Key size − The size of the key used in a symmetric key algorithm is a major determinant of its cryptographic strength. In general, the larger the key size, the stronger the algorithm.

- Block size − The block size of a symmetric key algorithm refers to the size of the blocks of data that are encrypted and decrypted using the algorithm. A larger block size can increase the cryptographic strength of the algorithm.
- Number of rounds − The number of rounds in a symmetric key algorithm refers to the number of times that the encryption and decryption process is repeated. A larger number of rounds can increase the cryptographic strength of the algorithm.
- Resistance to attacks − The resistance of a symmetric key algorithm to attacks, such as brute-force attacks or differential cryptanalysis, is another factor that determines its cryptographic strength. Algorithms that are resistant to these types of attacks are generally considered to be stronger.

Stronger algorithms are generally more resistant to attacks and more effective at protecting the confidentiality of the information they are used to encrypt.

TYPES OF SYMMETRIC KEY ALGORITHMS:

There are several different types of symmetric key algorithms, including

Block ciphers − Block ciphers are symmetric key algorithms that operate on fixed-size blocks of data and use a secret key to encrypt and decrypt the data. Examples of block ciphers include the Advanced Encryption Standard (AES) and Blowfish.

- Stream ciphers − Stream ciphers are symmetric key algorithms that operate on a stream of data and use a secret key to encrypt and decrypt the data. Stream ciphers are generally faster and more efficient than block ciphers, but they are also generally considered to be less secure.

- Feistel ciphers − Feistel ciphers are a type of block cipher that are based on a structure known as a Feistel network. They are widely used in symmetric key algorithms and are known for their efficiency and ease of implementation.

- Substitution-permutation ciphers − Substitution-permutation ciphers are a type of block cipher that use both substitution and permutation operations to encrypt and decrypt data. They are known for their strong cryptographic properties and are used in many modern symmetric key algorithms.

What is the DES Algorithm in Network Security?

The DES (Data Encryption Standard) algorithm is a symmetric-key block cipher created in the early 1970s by an IBM team and adopted by the National Institute of Standards and Technology (NIST). The algorithm takes the plain text in 64-bit blocks and converts them into ciphertext using 48-bit keys.

Since it's a symmetric-key algorithm, it employs the same key in both encrypting and decrypting the data. If it were an asymmetrical algorithm, it would use different keys for encryption and decryption.

Initial Permutation (IP)

The plain text is divided into smaller chunks of 64-bit size. The IP is performed before the first round. This phInitial Permutation (IP)

The plain text is divided into smaller chunks of 64-bit size. The IP is performed before the first round. This phase describes the implementation of the transposition process. For example, the 58th bit replaces the first bit, the 50th bit replaces the second bit, and so on. The resultant 64-bit text is split into two equal halves of 32-bit each called Left Plain Text (LPT) and Right Plain Text (RPT).

Step 1: Key Transformation

We already know that the DES process uses a 56-bit key, which is obtained by eliminating all the bits present in every 8th position in a 64-bit key. In this step, a 48-bit key is generated. The 56-bit key is split into two equal halves and depending upon the number of rounds the bits are shifted to the left in a circular fashion.

Due to this, all the bits in the key are rearranged again. We can observe that some of the bits get eliminated during the shifting process, producing a 48-bit key. This process is known as compression permutation.

Step 2: Expansion Permutation

Let's consider an RPT of the 32-bit size that is created in the IP stage. In this step, it is expanded from 32-bit to 48-bit. The RPT of 32-bit size is broken down into 8 chunks of 4 bits each and extra two bits are added to every chunk, later on, the bits are permutated among themselves leading to 48-bit data. An XOR function is applied in between the 48-bit key obtained from step 1 and the 48-bit expanded RPT.

ase describes the implementation of the transposition process. For example, the 58th bit replaces the first bit, the 50th bit replaces the second bit, and so on. The resultant 64-bit text is split into two equal halves of 32-bit each called Left Plain Text (LPT) and Right Plain Text (RPT).

Step 1: Key Transformation

We already know that the DES process uses a 56-bit key, which is obtained by eliminating all the bits present in every 8th position in a 64-bit key. In this step, a 48-bit key is generated. The 56-bit key is split into two equal halves and depending upon the number of rounds the bits are shifted to the left in a circular fashion.

Due to this, all the bits in the key are rearranged again. We can observe that some of the bits get eliminated during the shifting process, producing a 48-bit key. This process is known as compression permutation.

Step 2: Expansion Permutation

Let's consider an RPT of the 32-bit size that is created in the IP stage. In this step, it is expanded from 32-bit to 48-bit. The RPT of 32-bit size is broken down into 8 chunks of 4 bits each and extra two bits are added to every chunk, later on, the bits are permutated among themselves leading to 48-bit data. An XOR function is applied in between the 48-bit key obtained from step 1 and the 48-bit expanded RPT.

DES Algorithm steps:

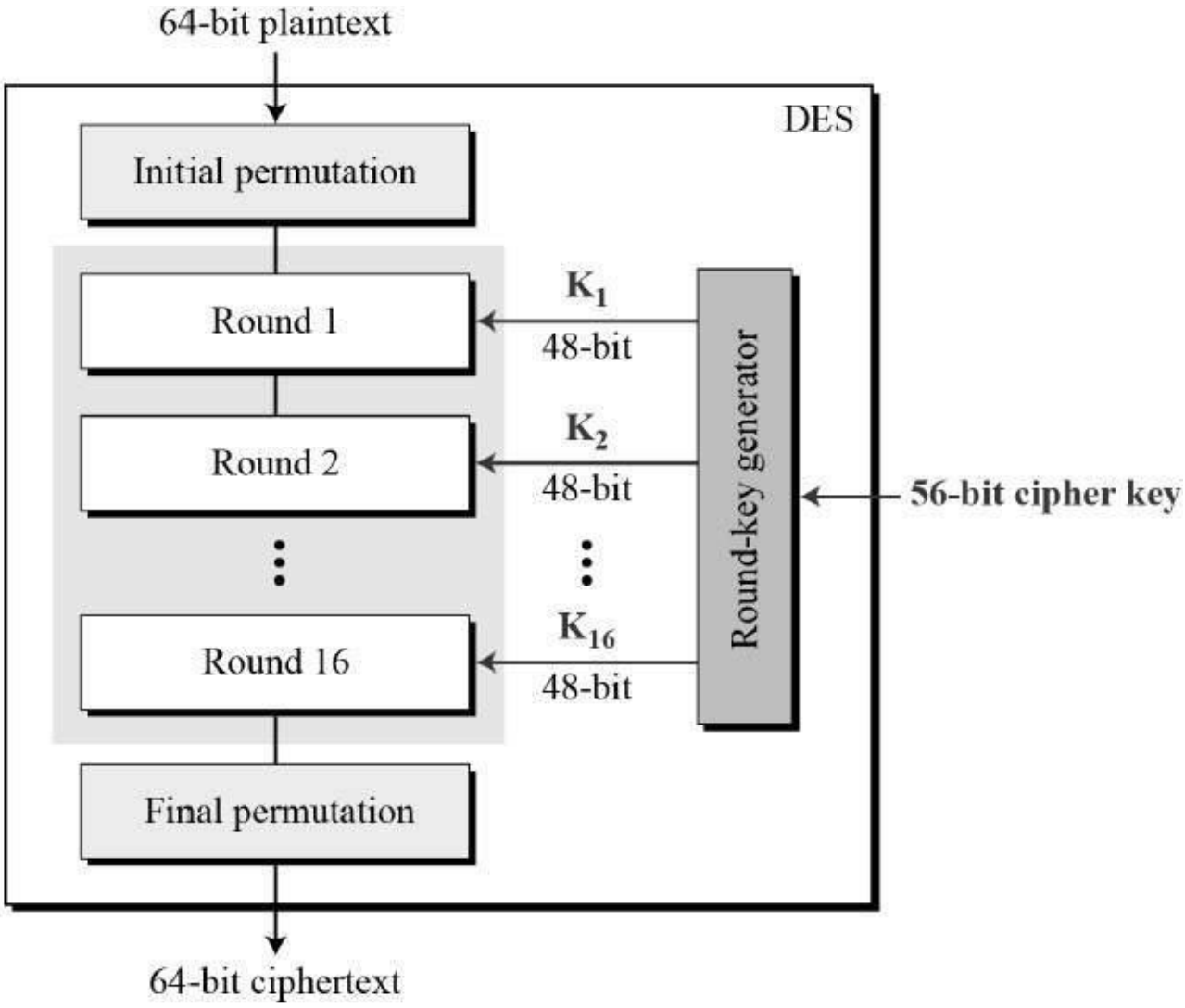
The algorithm process breaks down into the following steps:

1. The process begins with the 64-bit plain text block getting handed over to an initial permutation (IP) function.
2. The initial permutation (IP) is then performed on the plain text.
3. Next, the initial permutation (IP) creates two halves of the permuted block, referred to as Left Plain Text (LPT) and Right Plain Text (RPT).

4. Each LPT and RPT goes through 16 rounds of the encryption process.

5. Finally, the LPT and RPT are rejoined, and a Final Permutation (FP) is performed on the newly combined block.

6. The result of this process produces the desired 64-bit ciphertext.



Single Round of Execution of DES Algorithm: