Data Encryption Standard (DES)

Sources:

- *Applied Cryptography* by B. Schneier
- *Cryptography and Network Security* by W. Stallings
- *Standing the Test of Time: The Data Encryption Standard* by S. Landau, Notices of the AMS, March 2000

DES has been a worldwide standard for more than 20 years.

It was designed by IBM, and it was adopted by the National Bureau of Standards (NBS) in 1976 for use in unclassified government communications.

Subsequent official publications described in detail how to implement it.

The American National Standards Institute (ANSI) approved DES as a private-sector standard in 1981.

The National Institute of Standards and Technology (NIST) validates implementations of DES.

Substitution and Permutation

Claude Shannon observed that there are two fundamental techniques for encryption:

- *Confusion* – obscuring the relationship between the plaintext and the ciphertext

- *Diffusion* – Spreading the change throughout the ciphertext

The simplest form of confusion is *substitution*: replacing one symbol by another.

The simplest form of diffusion is *permutation*: moving the symbols of a block around.

Frequency analysis can be used to break both.

Nevertheless, combinations of these operations form the *backbone* of modern cryptosystems.

Description of DES

(Single) DES is a block cipher; it encrypts data in 64-bit blocks.

The same algorithm and key are used for encryption and decryption (except for minor differences in key schedule).

The key length is 56 bits.

(The key is usually expressed as a 64-bit number, but every eight bit is used for parity checking and ignored.)

The key can be any 56-bit number.

A handful of numbers are considered weak keys, e.g.,

$$E_k(E_k(M)) = M$$

$$E_{k1}(E_{k2}(M)) = M$$

The fundamental building block of DES is called a *round*.

A round consists of a substitution followed by a permutation.

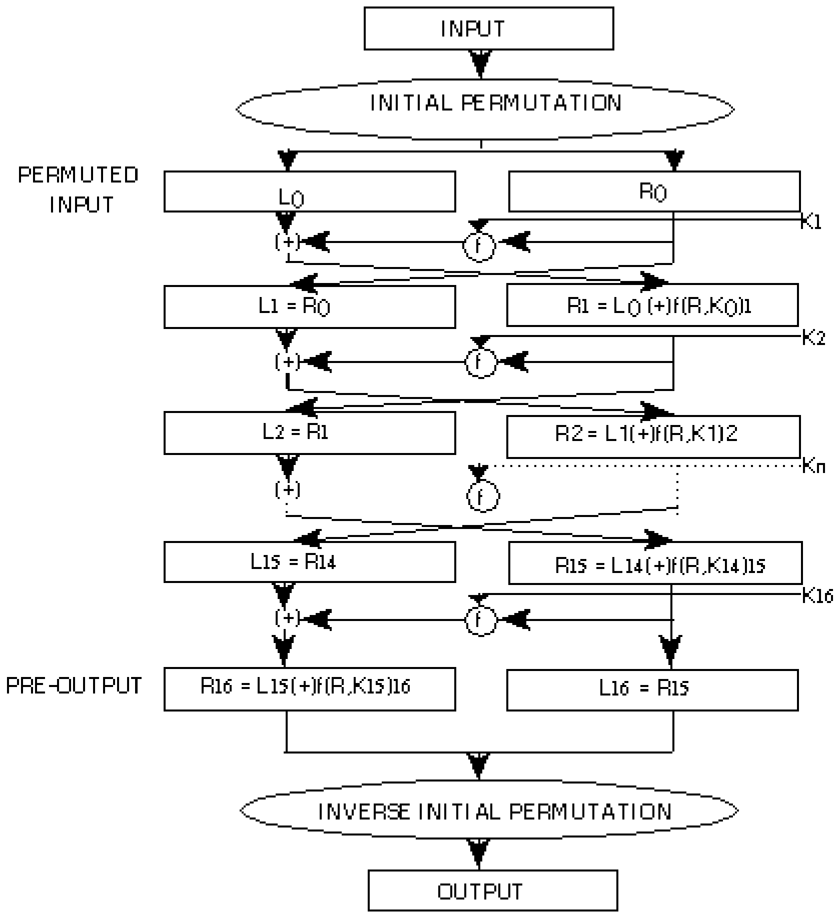
It is based on the key.

DES has 16 rounds.

The algorithm uses only standard arithmetic and logical operations on numbers of at most 64 bits.

The repetitive nature of the algorithm makes it ideal for hardware implementation.

DES flowchart:



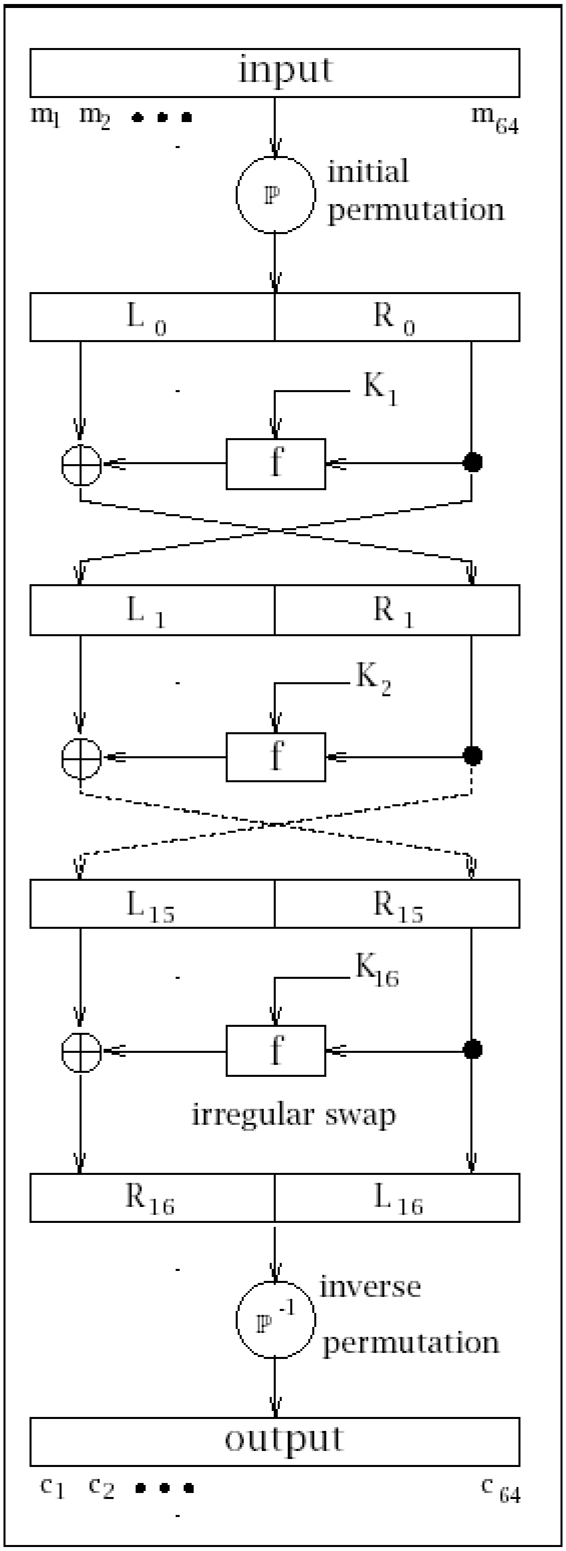


Figure 1. The Data Encryption Standard.

After an initial permutation of the plaintext block, it is broken into a right half and left half.

There are 16 rounds of identical operations, called *Function f,* in which the data are combined with the key.

After the 16th round, the right and left halves are joined.

A final permutation completes the algorithm.

The initial and final permutations don’t affect DES’s security.

Round of DES

In each round, the key bits are shifted, then 48 bits are selected from the 56 bits of the key.

The right half of the data is expanded to 48 bits via an *expansion permutation*.

It is then combined with 48 bits of a shifted and permuted key via an XOR.

It is then sent through 8 “S-boxes” producing 32 new bits and permuted again.

These operations make up Function f.

The output of Function f is then combined with the left half via another XOR.

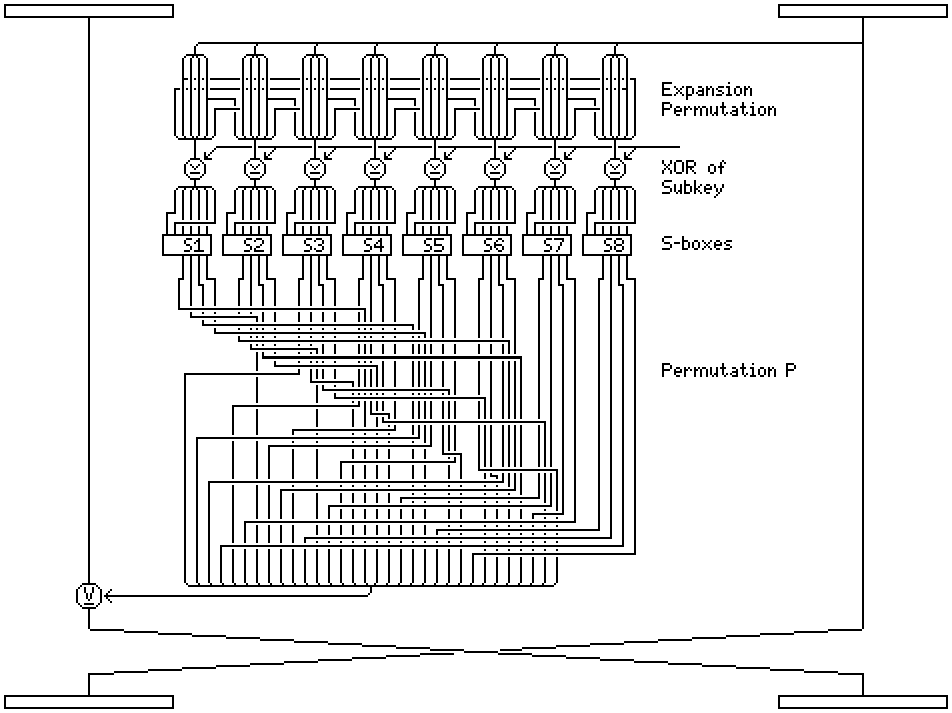
The result of these operations becomes the new right half; the old right half becomes the new left half.

A round looks like this:

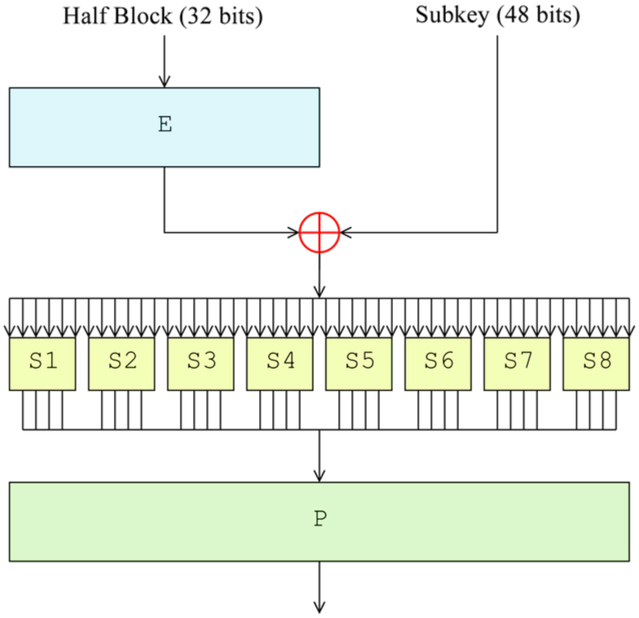
$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus \mathrm{f}(R_{i-1}, K_i)$$

Round of DES:



:



[http://en.wikipedia.org/wiki/File:Data_Encryption_Standard_InfoBox_Diagram.png]

The Key Transformation

Initially, the 64-bit DES key is reduced to a 56-bit key by ignoring every eighth bit.

After the 56-bit key is extracted, a different 48-bit *subkey* is generated for each of the 16 rounds of DES.

First, the 56-bit key is divided into two 28-bit halves.

Then, the halves are circularly-shifted left by either one or two bits, depending on the round.

After being shifted, 48 bits out of the 56 bits are selected.

Because this operation permutes the bits as well as selects a subset of them, it is called a *compression permutation*.

Each key bit is used in approximately 14 of the 16 subkeys.

The Expansion Permutation

This operation expands the right half of the data, *Ri*, from 32 bits to 48 bits.

It is called an *expansion permutation* because it changes the order of the bits and repeats certain bits.

It has three purposes:

1. It makes the right half the same size as the key for the XOR operation.

2. It provides a longer result that can be compressed during the substitution operation.

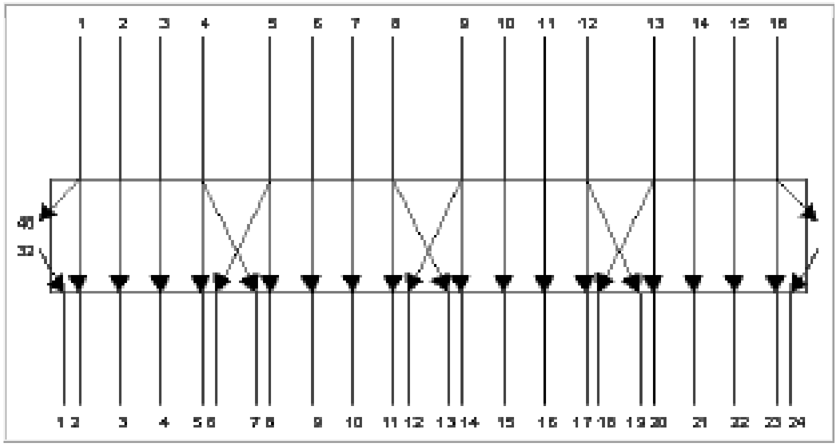
3. It helps to make every bit of the ciphertext depend on every bit of the plaintext and every bit of the key as quickly as possible.

The condition in (3) is called the *avalanche effect*.

The 1st and 4th bits of each 4-bit input block each represent two bits of the output block.

The 2nd and 3rd bits each represent one bit of the output block.

DES expansion permutation:



The S-Box Substitution

After the compressed key is XORed with the expanded block, the 48-bit result is sent through a substitution operation.

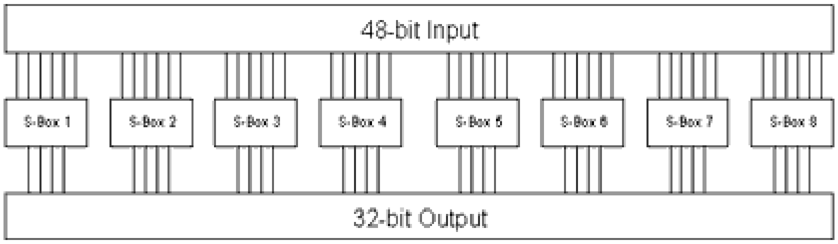
Substitutions are performed in parallel by 8 *substitution boxes* or *S-boxes*.

Each S-box has a 6-bit input and a 4-bit output.

The 48 bits are divided into eight 6-bit sub-blocks.

Each sub-block is operated on by a separate S-box, each containing a lookup table.

S-box substitution is the critical step in DES.



The P-Box Permutation

The 32-bit output of the S-box substitution is permuted according to a *P-box* permutation.

Decrypting DES

With DES it is possible to use the same function to encrypt and decrypt a block.

However, the sub-keys must be used in the reverse order.

That is, if the encryption keys for successive rounds are $K_1, K_2, \ldots, K_{16}$, then the decryption keys are $K_{16}, K_{15}, \ldots, K_1$.

Differential Cryptanalysis

Eli Biham and Adi Shamir introduced *differential cryptanalysis* in 1990.

Using it, they found a chosen-plaintext attack against DES that was more efficient than brute force.

Differential cryptanalysis looks at *ciphertext pairs*: pairs of ciphertext whose plaintexts have particular differences.

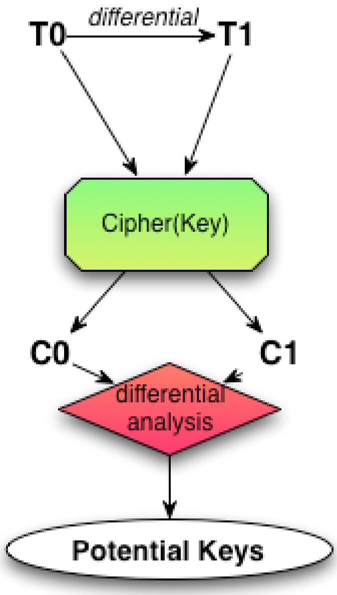
It analyzes the evolution of these differences as the plaintexts propagate through the rounds of DES when they are encrypted with the same key:

1. Choose pairs of plaintexts with a fixed difference.

2. Using the differences in the resulting ciphertexts, assign different probabilities to different keys.

3. As more and more ciphertext pairs are analyzed, one key will emerge as the most probable.

Differential cryptanalysis can be targeted at a *single stage* of the cipher.



Differential Cryptanalysis

[scienceblogs.com/goodmath/2008/10/02/differential-cryptanalysis/]

In 1998, a coalition of Cryptography Research, Advanced Wireless Technologies, and the Electronic Frontier Foundation used a key search machine called DES Cracker to find the key of the RSA's DES Challenge after searching 56 hours.

This demonstrated that 56 bit keys are too short for current and future cryptographic applications.

As of 2008, DES could be broken is less than a day with specialized hardware.

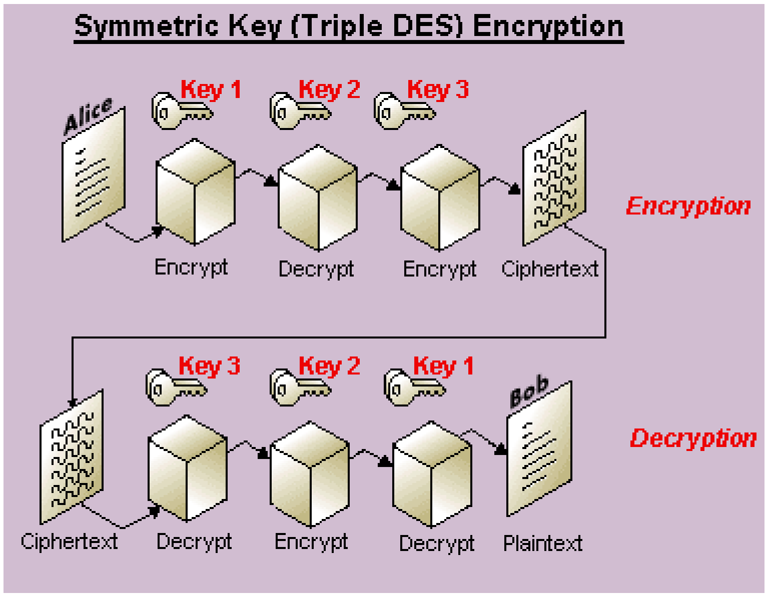
Triple DES addresses this issue by running the DES algorithm three times.

Encryption is described by

$$c \leftarrow E_{k1}(D_{k2}(E_{k1}(m)))$$

and decryption is described by

$$m \leftarrow D_{k1}(E_{k2}(D_{k1}(c)))$$



[http://www.smartcardbasics.com/smart_card_images/panel7_3des.gif]