

The **Data Encryption Standard** (**DES**, [/ˌdiːˌiːˈɛs, dɛz/](https://en.wikipedia.org/wiki/Help:IPA/English)) is a [symmetric-key algorithm](https://en.wikipedia.org/wiki/Symmetric-key_algorithm) for the [encryption](https://en.wikipedia.org/wiki/Encryption) of electronic data. Although insecure, it was highly influential in the advancement of modern [cryptography](https://en.wikipedia.org/wiki/Cryptography).

History:

Developed in the early 1970s at [IBM](https://en.wikipedia.org/wiki/IBM) and based on an earlier design by [Horst Feistel](https://en.wikipedia.org/wiki/Horst_Feistel), the algorithm was submitted to the [National Bureau of Standards](https://en.wikipedia.org/wiki/National_Bureau_of_Standards) (NBS) following the agency's invitation to propose a candidate for the protection of sensitive, unclassified electronic government data. In 1976, after consultation with the [National Security Agency](https://en.wikipedia.org/wiki/National_Security_Agency) (NSA), the NBS eventually selected a slightly modified version (strengthened against [differential cryptanalysis](https://en.wikipedia.org/wiki/Differential_cryptanalysis), but weakened against [brute-force attacks](https://en.wikipedia.org/wiki/Brute-force_attack)), which was published as an official [Federal Information Processing Standard](https://en.wikipedia.org/wiki/Federal_Information_Processing_Standard) (FIPS) for the United States in 1977.

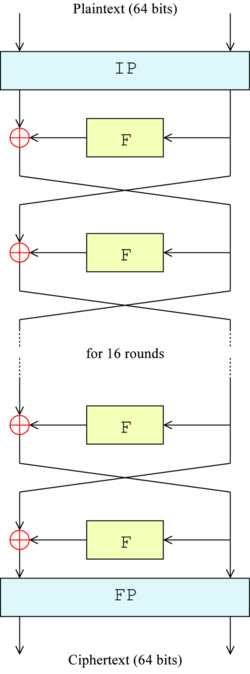
Overall Structure

The algorithm's overall structure is shown in Figure 1:

there are 16 identical stages of processing, termed *rounds*. There is also an initial and final [permutation](https://en.wikipedia.org/wiki/Permutation), termed *IP* and *FP*, which are [inverses](https://en.wikipedia.org/wiki/Inverse_function) (IP "undoes" the action of FP, and vice versa). IP and FP have no cryptographic significance, but were included in order to facilitate loading blocks in and out of mid-1970s 8-bit based hardware.[[25]](https://en.wikipedia.org/wiki/Data_Encryption_Standard" \l "cite_note-25)

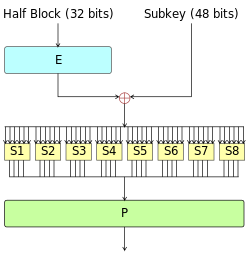
Before the main rounds, the block is divided into two 32-bit halves and processed alternately; this criss-crossing is known as the [Feistel scheme](https://en.wikipedia.org/wiki/Feistel_scheme). The Feistel structure ensures that decryption and encryption are very similar processes—the only difference is that the subkeys are applied in the reverse order when decrypting. The rest of the algorithm is identical. This greatly simplifies implementation, particularly in hardware, as there is no need for separate encryption and decryption algorithms.

The ⊕ symbol denotes the [exclusive-OR](https://en.wikipedia.org/wiki/XOR) (XOR) operation. The *F-function* scrambles half a block together with some of the key. The output from the F-function is then combined with the other half of the block, and the halves are swapped before the next round. After the final round, the halves are swapped; this is a feature of the Feistel structure which makes encryption and decryption similar processes.



### **The Feistel (F) function**

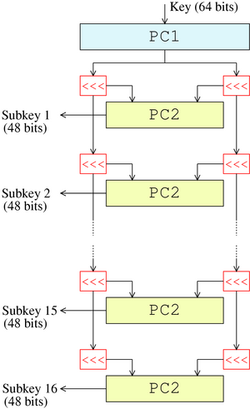
The F-function, depicted in Figure 2, operates on half a block (32 bits) at a time and consists of four stages:



[*Figure 2*](https://en.wikipedia.org/wiki/File:DES-f-function.png)—The Feistel function (F-function) of DES

1. *Expansion*: the 32-bit half-block is expanded to 48 bits using the *expansion permutation*, denoted *E* in the diagram, by duplicating half of the bits. The output consists of eight 6-bit (8 \* 6 = 48 bits) pieces, each containing a copy of 4 corresponding input bits, plus a copy of the immediately adjacent bit from each of the input pieces to either side.
2. *Key mixing*: the result is combined with a *subkey* using an XOR operation. Sixteen 48-bit subkeys—one for each round—are derived from the main key using the [*key schedule*](https://en.wikipedia.org/wiki/Key_schedule) (described below).
3. *Substitution*: after mixing in the subkey, the block is divided into eight 6-bit pieces before processing by the [*S-boxes*](https://en.wikipedia.org/wiki/Substitution_box), or *substitution boxes*. Each of the eight S-boxes replaces its six input bits with four output bits according to a non-linear transformation, provided in the form of a [lookup table](https://en.wikipedia.org/wiki/Lookup_table). The S-boxes provide the core of the security of DES—without them, the cipher would be linear, and trivially breakable.
4. *Permutation*: finally, the 32 outputs from the S-boxes are rearranged according to a fixed [permutation](https://en.wikipedia.org/wiki/Permutation), the *P-box*. This is designed so that, after permutation, the bits from the output of each S-box in this round are spread across four different S-boxes in the next round.

### **Key schedule**[[edit](https://en.wikipedia.org/w/index.php?title=Data_Encryption_Standard&action=edit&section=8)]



[*Figure 3*](https://en.wikipedia.org/wiki/File:DES-key-schedule.png)— The key-schedule of DES

Figure 3 illustrates the *key schedule* for encryption—the algorithm which generates the subkeys. Initially, 56 bits of the key are selected from the initial 64 by *Permuted Choice 1* (*PC-1*)—the remaining eight bits are either discarded or used as [parity](https://en.wikipedia.org/wiki/Parity_bit) check bits. The 56 bits are then divided into two 28-bit halves; each half is thereafter treated separately. In successive rounds, both halves are rotated left by one or two bits (specified for each round), and then 48 subkey bits are selected by *Permuted Choice 2* (*PC-2*)—24 bits from the left half, and 24 from the right. The rotations (denoted by "<<<" in the diagram) mean that a different set of bits is used in each subkey; each bit is used in approximately 14 out of the 16 subkeys.