ТУТ ДОЛЖЕН БЫТЬ ТИТУЛЬНЫЙ ЛИСТ

**Data Encryption Standard (DES)**

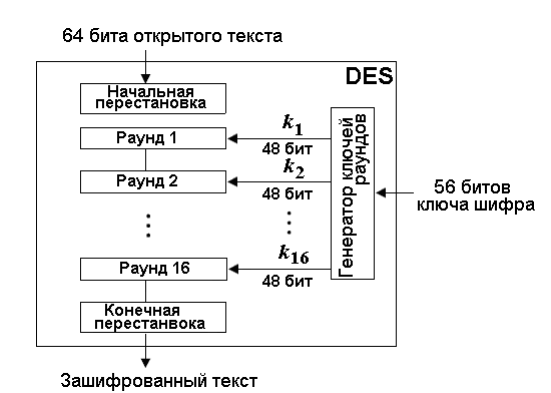
The Data Encryption Standard (DES) is a block cipher with symmetric keys, developed by the National Institute of Standards and Technology (NIST).

History: In 1973, NIST issued a request for the development of a proposal for a national cryptographic system with symmetric keys. The modification of the project proposed by IBM, named LUCIFER, was adopted as DES. In March 1975, the DES crypto algorithm was published in draft form in the Federal Register as the Federal Information Processing Standard (FIPS).

After publication, the algorithm was severely criticized for two reasons. The first was the critically small key length of 56 bits, which could make the cipher vulnerable to a brute force attack. The second reason was that critics were concerned about some hidden construction in the internal structure of DES. They suspected that the S-boxes had a backdoor that would allow the National Security Agency of the USA to decrypt messages without the key. Subsequently, IBM designers reported that the internal structure had been refined to prevent cryptanalysis. The Federal Register declared DES as the encryption standard, and it quickly became the most widely used block cipher. Later, NIST proposed a new standard, recommending the use of the DES cipher three times in succession. In 2000, the new AES standard replaced DES.

**General Provisions**

For encryption, DES takes a 64-bit plaintext and generates a 64-bit ciphertext and vice versa, receiving 64 bits of encrypted text, it outputs 64 bits of decrypted text. In both cases, the same 56-bit key is used for encryption and decryption.



**Figure 1.** DES structure

The encryption process consists of two permutations, known as the initial and final (or terminal) permutations, and 16 rounds of Feistel. Each round uses different generated 48-bit keys.

**Initial IP and final IP-1 permutations**

Both of these permutations take 64 bits as input, which are then rearranged according to predefined tables. These permutations are mutually inverse. In other words, the 58th bit at the input of the initial permutation moves to the 1st position at its output. And the final permutation will move the 1st input bit to the 58th position at the output.



**Figure 2.** Initial and final permutations

Both permutations have no cryptographic significance. The reason why they are included in DES is unclear and not specified by the designers of DES.

***Example:***

**Initial permutation:**

Input:

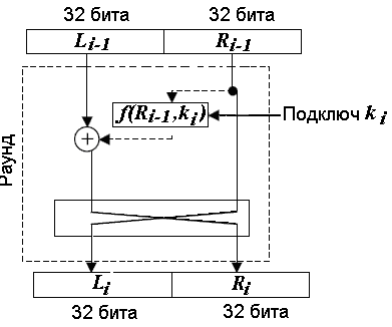
0000000000000010000000000000000000000000000000000000000000000001

Output:

0000000000000000000000001000000000000000000000000000000000000010

**DES Rounds**

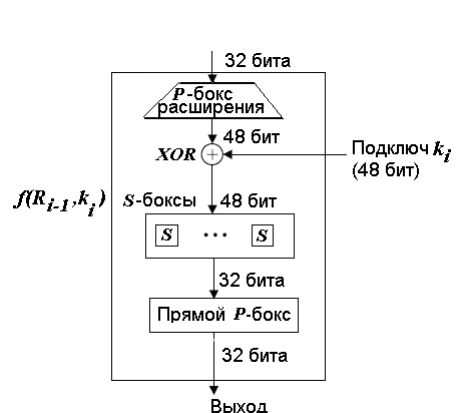
DES uses 16 rounds. Each round of DES applies the Feistel cipher, as shown in the figure. A round takes the half-blocks Li-1​ and Ri−1​ from the previous round (or the initial permutation block) and creates half-blocks Li​ and Ri​ for input into the next round (or the final permutation block). All irreversible elements are concentrated in the function *f(Ri-1, ki).*



**Figure 3.** DES Rounds

**DES Function**

The DES function encrypts the 32 rightmost bits Ri−1 ​using a 48-bit key to produce a 32-bit output. This function consists of four components, as shown: the XOR operation, the expansion P-box, a group of S-boxes, and the straight box.



**Figure 4.** DES Funtion

The expansion P-box is used to expand the 32-bit block Ri−1 to 48 bits to match the size of the round subkey. The block Ri−1 is divided into 8 sections of 4 bits each. Each section is expanded to 6 bits. (For the section, the values of the input bits in positions 1, 2, 3, and 4 are assigned to the bits in positions 2, 3, 4, and 5 respectively at the output. The 1st output bit is equal to the 4th input bit of the previous section; the 6th output bit is equal to the 1st bit of the following section. If sections 1 and 8 are considered adjacent sections, then the same rules apply to bits 1 and 32).

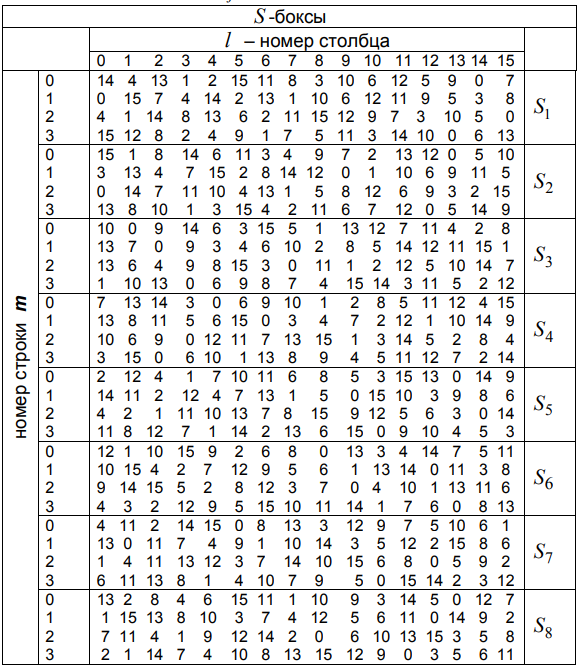
Although the relationships between input and output can be defined mathematically, the P-box is defined by a table.



**Figure 5.** P-expansion box

After expansion, DES uses the XOR operation on the expanded part of the right half-block and the round key. After the XOR with the key bits, the 48-bit block is divided into 8 consecutive 6-bit vectors, each of which is replaced by a 4-bit vector using S-boxes.

Each S-box is a non-linear substitution box that takes a 6-bit input and produces a 4-bit output. There are 8 S-boxes in total, one for each of the 6-bit vectors. The substitution provided by the S-boxes introduces confusion into the cipher, which is an essential aspect of the security of DES. The S-boxes are designed to be resistant to known cryptanalytic attacks, and each one has a predefined substitution pattern that is used during the encryption process.



**Figure 6.** S-box structure

Thus, after the S-boxes, we obtain 8 4-bit vectors, which are then combined back into a 32-bit block. Next, the bits of the block are shuffled in the straight P-box based on a predefined table (the rules for using the permutation table are the same: for example, the 7th input bit becomes the 2nd output bit).

The straight P-box, also known as the permutation function, rearranges the bits of the 32-bit output from the S-boxes to produce a permuted 32-bit output. This permutation is designed to spread the output bits of the S-boxes across the next round's input, which helps to ensure that the changes to one bit will affect many bits in subsequent rounds, contributing to the diffusion property of the cipher.

After the 16th round of DES, the right and left blocks are not swapped. Instead, they are combined into a block and subjected to the final permutation.

**Key Generation**

DES generates 16 round keys of 48 bits each from a 56-bit cipher key k. However, to set the cipher key, 8 parity check bits need to be inserted at positions 8, 16, ..., 64 among the 56 key bits, so that each byte contains an odd number of ones. This operation is used to detect errors in key exchange and storage.

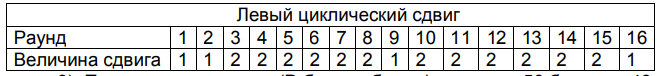
The key schedule consists of the following steps:

Compression permutation to remove parity bits – from the 64-bit key, bits 8, 16, 24, 32, ..., 64 are removed, and the remaining bits are rearranged according to a table (the numbering of the bits of the expanded key is preserved during the permutation).



**Figure 7.** Key parity bit removal + permutation

After the permutation, the 56 bits of the key are divided into two blocks of 28 bits each. Then, to generate the round keys from blocks are built using a left circular shift operation of 1-2 bits. In rounds 1, 2, 9, and 16, the shift is by 1 bit; in other rounds, it is by 2 bits. After determining the blocks, the bits of these blocks are combined into one 56-bit key.

Left circular shift: 

**Figure 8.** Left circular shift

Compression permutation (P-box, table) changes the 56 bits to 48 bits, which form the round key.



**Figure 9.** Key compression permutation

**Implementation**

Output:

ENCRYPTION 123456ABCD132536  
  
After initial permutation: 14A7D67818CA18AD  
  
Round 1: LPT = 18CA18AD, RPT = 5A78E394, KEY = 194CD072DE8C  
Round 2: LPT = 5A78E394, RPT = 4A1210F6, KEY = 4568581ABCCE  
Round 3: LPT = 4A1210F6, RPT = B8089591, KEY = 06EDA4ACF5B5  
Round 4: LPT = B8089591, RPT = 236779C2, KEY = DA2D032B6EE3  
Round 5: LPT = 236779C2, RPT = A15A4B87, KEY = 69A629FEC913  
Round 6: LPT = A15A4B87, RPT = 2E8F9C65, KEY = C1948E87475E  
Round 7: LPT = 2E8F9C65, RPT = A9FC20A3, KEY = 708AD2DDB3C0  
Round 8: LPT = A9FC20A3, RPT = 308BEE97, KEY = 34F822F0C66D  
Round 9: LPT = 308BEE97, RPT = 10AF9D37, KEY = 84BB4473DCCC  
Round 10: LPT = 10AF9D37, RPT = 6CA6CB20, KEY = 02765708B5BF  
Round 11: LPT = 6CA6CB20, RPT = FF3C485F, KEY = 6D5560AF7CA5  
Round 12: LPT = FF3C485F, RPT = 22A5963B, KEY = C2C1E96A4BF3  
Round 13: LPT = 22A5963B, RPT = 387CCDAA, KEY = 99C31397C91F  
Round 14: LPT = 387CCDAA, RPT = BD2DD2AB, KEY = 251B8BC717D0  
Round 15: LPT = BD2DD2AB, RPT = CF26B472, KEY = 3330C5D9A36D  
Round 16: LPT = 19BA9212, RPT = CF26B472, KEY = 181C5D75C66D  
Cipher Text : C0B7A8D05F3A829C  
======================================================================  
  
DECRYPTION C0B7A8D05F3A829C  
  
After initial permutation: 19BA9212CF26B472  
  
Round 1: LPT = CF26B472, RPT = BD2DD2AB, KEY = 194CD072DE8C  
Round 2: LPT = BD2DD2AB, RPT = 387CCDAA, KEY = 4568581ABCCE  
Round 3: LPT = 387CCDAA, RPT = 22A5963B, KEY = 06EDA4ACF5B5  
Round 4: LPT = 22A5963B, RPT = FF3C485F, KEY = DA2D032B6EE3  
Round 5: LPT = FF3C485F, RPT = 6CA6CB20, KEY = 69A629FEC913  
Round 6: LPT = 6CA6CB20, RPT = 10AF9D37, KEY = C1948E87475E  
Round 7: LPT = 10AF9D37, RPT = 308BEE97, KEY = 708AD2DDB3C0  
Round 8: LPT = 308BEE97, RPT = A9FC20A3, KEY = 34F822F0C66D  
Round 9: LPT = A9FC20A3, RPT = 2E8F9C65, KEY = 84BB4473DCCC  
Round 10: LPT = 2E8F9C65, RPT = A15A4B87, KEY = 02765708B5BF  
Round 11: LPT = A15A4B87, RPT = 236779C2, KEY = 6D5560AF7CA5  
Round 12: LPT = 236779C2, RPT = B8089591, KEY = C2C1E96A4BF3  
Round 13: LPT = B8089591, RPT = 4A1210F6, KEY = 99C31397C91F  
Round 14: LPT = 4A1210F6, RPT = 5A78E394, KEY = 251B8BC717D0  
Round 15: LPT = 5A78E394, RPT = 18CA18AD, KEY = 3330C5D9A36D  
Round 16: LPT = 14A7D678, RPT = 18CA18AD, KEY = 181C5D75C66D  
Plain Text : 123456ABCD132536

**Conclusion**

In conclusion, the Data Encryption Standard (DES) is a symmetric-key algorithm for the encryption of digital data. Although it was widely used and considered secure for many years, the relatively short 56-bit key has been its Achilles' heel, leading to its eventual replacement by the Advanced Encryption Standard (AES). DES's encryption process involves an initial permutation, followed by a series of 16 rounds using the Feistel network structure, and ends with a final permutation.

Each round of DES includes key-dependent transformations using a round key derived from the original key through a complex key schedule. This schedule involves permuting and selecting subsets of the key bits to generate 16 distinct 48-bit round keys. The Feistel structure ensures that encryption and decryption are very similar processes, with the only difference being the order in which the round keys are applied.

The heart of the DES algorithm lies in its S-boxes, which provide non-linearity and complexity essential for security. The S-boxes, combined with the P-box permutations, create a diffusion effect that spreads the influence of a single plaintext bit over many ciphertext bits, making the cipher resistant to statistical analysis.

Despite its historical significance, DES's vulnerability to brute-force attacks due to its short key length has rendered it obsolete for protecting sensitive data in the modern era. The DES story is a testament to the evolution of cryptography from an art to a science, demonstrating the need for constant adaptation to the ever-increasing computational power and sophistication of attackers. The transition from DES to AES reflects the ongoing pursuit of cryptographic algorithms that balance security, efficiency, and practicality in the face of advancing technology and emerging threats.

**Link to GitHub repository:**https://github.com/Ricigeroi/CS\_Lab\_4