Implementing DES

Based on *Applied Cryptography* by Schneier, Chapter 12

Lecture notes of Alexander Wood
awood@jjay.cuny.edu

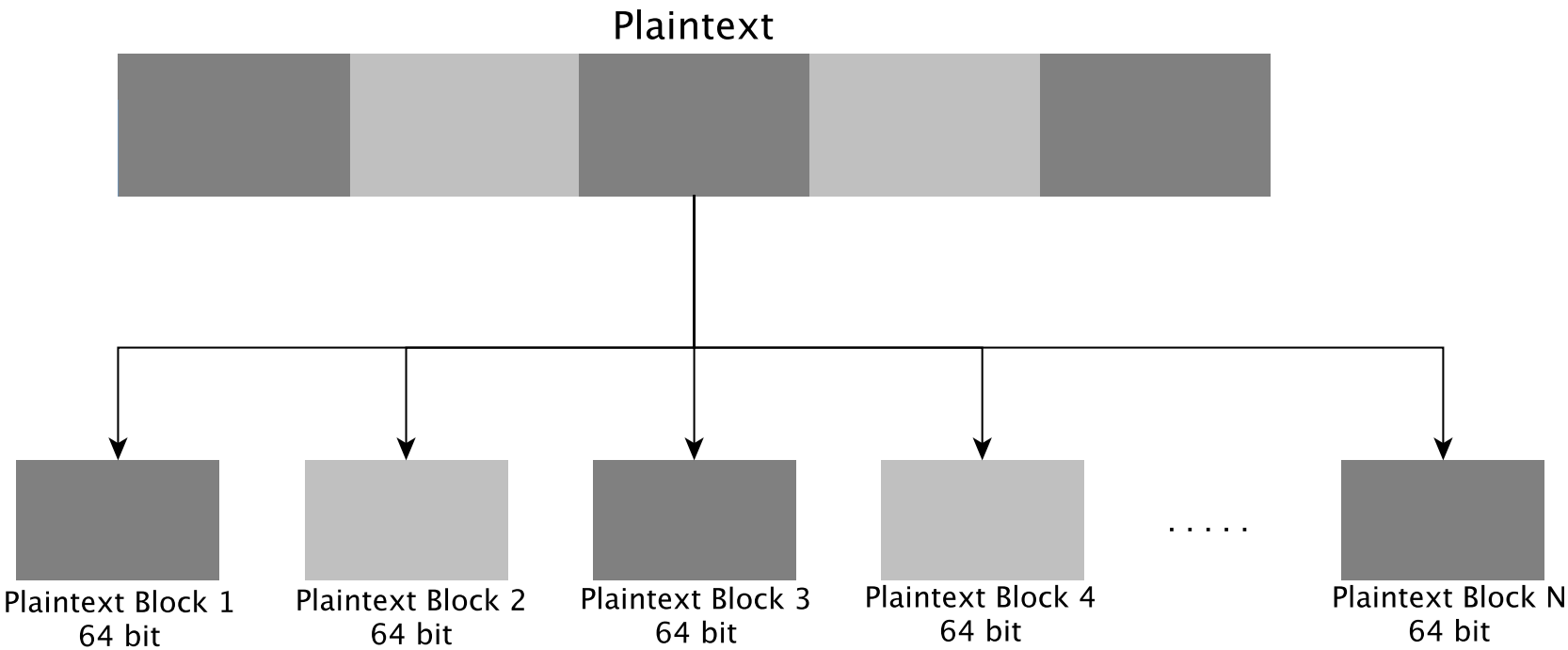
John Jay College of Criminal Justice

Overview of DES

- Block Cipher
- Uses 64-bit blocks
- Symmetric (aka, Private-key), meaning the decryption key equals the encryption key

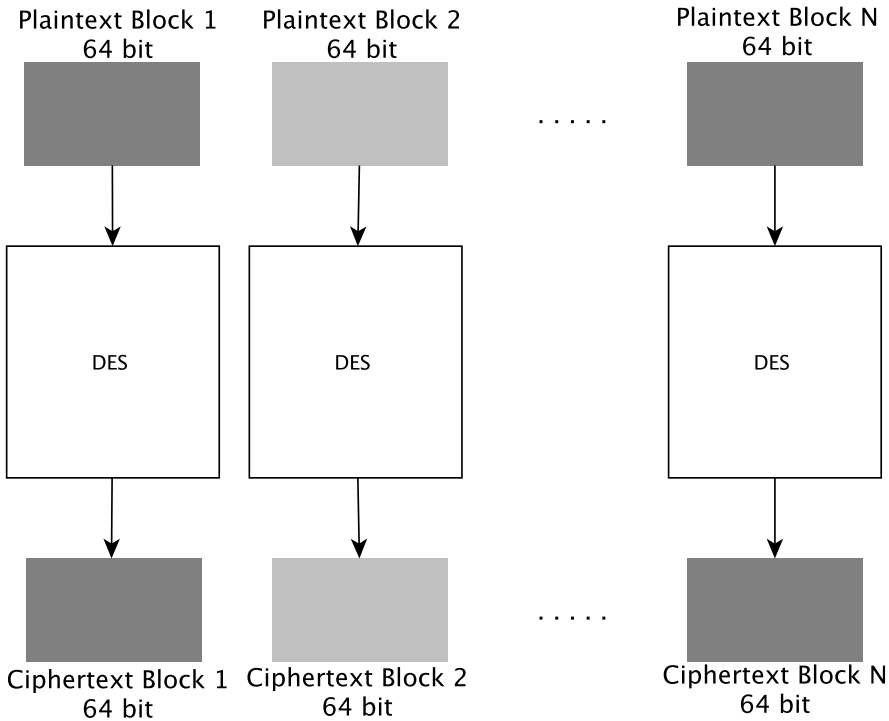
Overview of DES

The plaintext is split into 64-bit blocks.



Overview of DES

Data is encrypted block-by-block.



The ciphertext is constructed by combining the ciphertext blocks.

DES Key

The key for each DES block is expressed as a 64-bit number. Every eighth bit is used for parity checking and is ignored. (The parity check requires that each byte contains an odd number of `1` bits.)

Thus, **the key is a 54-bit number** which can be changed at any time.

DES Overview

The DES algorithm combines **confusion** and **diffusion**.

DES has **16 rounds** which consist of a **substitution** followed by a **permutation**.

DES Overview

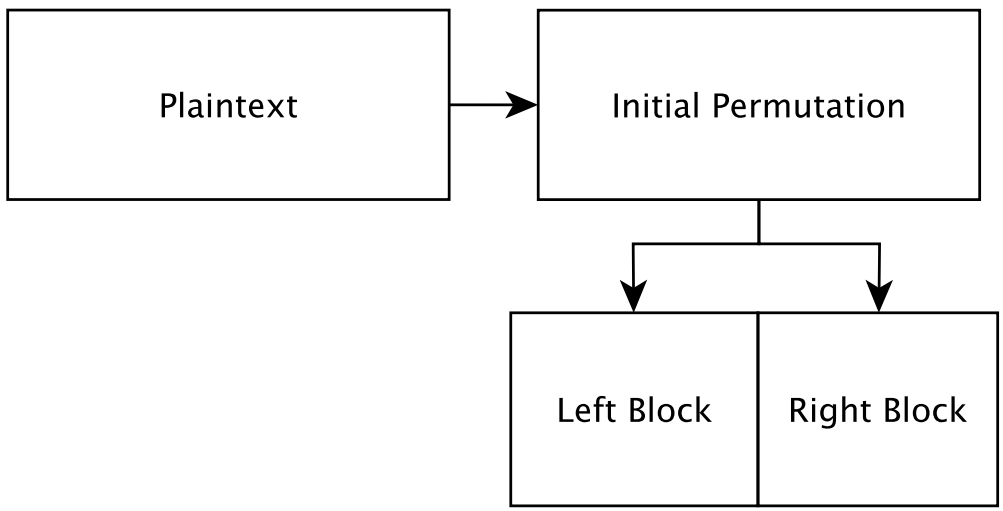
DES was exceptional in that it was easy to implement. It requires only standard arithmetic and logical operations.

DES Step 1: Initialization (Overview)

Input: a 64-bit plaintext block.

(1) An initial permutation is applied to the block.

(2) The block is broken into a right and a left half, each of which is 32 bits long.



DES Rounds (Overview)

Next, the DES round is carried out 16 times.

(1) The key bits are **shifted**

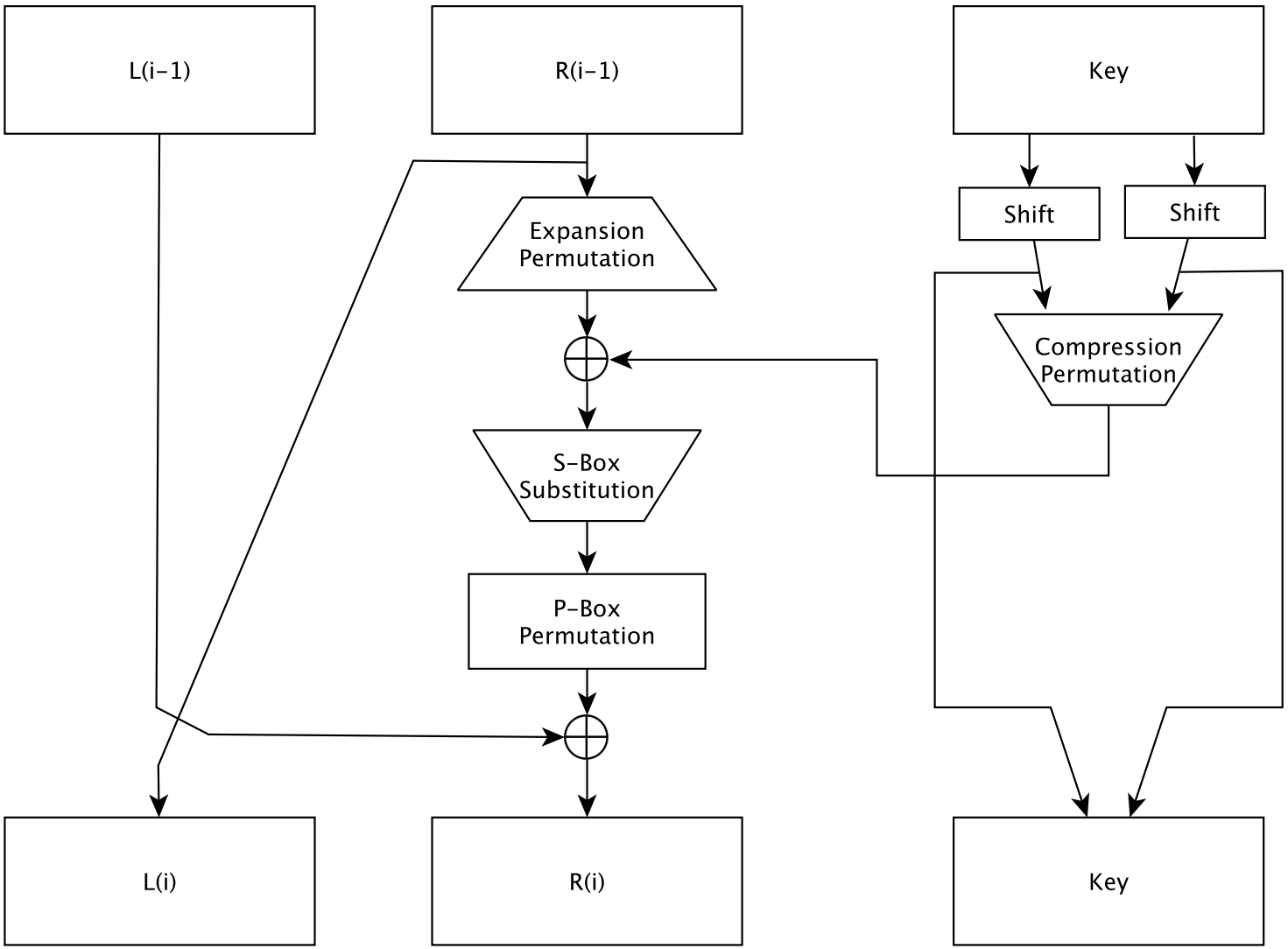
(2) Apply the function f, which does the following:
   - 48 bits are selected from the 56 bits of the key
   - The Right Block is expanded to 48 bits via an expansion permutation
   - The expanded Right Block is combined with 48 bits of a shifted and permuted key via XOR
   - This is sent through 8 **S-boxes** producing 32 new bits
   - Permute once again.

(3) The output of f is combined with the left half via XOR

(4) The result is the new right half

(5) The old right half becomes the new left half.

DES Rounds (Overview)



Historical Note

The initial permutation and final permutation have no affect on the security of the DES algorithm. It is postulated that the purpose of these permutations was to make it easier to load plaintexts and ciphertexts into a DES chip in byte-sized pieces, since this algoirthm predates even 16-bit microprocessor buses.

An Example

Next, we will walk through the example provided in Section 12.2 of Schneier's *Applied Cryptography*. See this section in the textbook for further elaboration.

DES Example: Initial Permutation

Transpose the 64-bit input block according to the following table. Bit 1 of the permuted block is bit 58 of the permuted block, etc.

Table 12.1
Initial Permutation

58,	50,	42,	34,	26,	18,	10,	2,	60,	52,	44,	36,	28,	20,	12,	4,
62,	54,	46,	38,	30,	22,	14,	6,	64,	56,	48,	40,	32,	24,	16,	8,
57,	49,	41,	33,	25,	17,	9,	1,	59,	51,	43,	35,	27,	19,	11,	3,
61,	53,	45,	37,	29,	21,	13,	5,	63,	55,	47,	39,	31,	23,	15,	7

Because this initial permutation does not add to security, many modern applications left out the initial and final permutation steps. Though this did not affect security, it did not follow the DES standard and hence could not be called DES.

DES Example: Key Permutation

The 64-bit DES key is reduced to a 56-bit key by ignoring every eighth bit. This also includes the parity check.

Table 12.2
Key Permutation

57,	49,	41,	33,	25,	17,	9,	1,	58,	50,	42,	34,	26,	18,
10,	2,	59,	51,	43,	35,	27,	19,	11,	3,	60,	52,	44,	36,
63,	55,	47,	39,	31,	23,	15,	7,	62,	54,	46,	38,	30,	22,
14,	6,	61,	53,	45,	37,	29,	21,	13,	5,	28,	20,	12,	4

DES Example: Subkeys Step 1

A different 48-bit subkey K_i is generated for each of the 16 rounds.

Table 12.3
Number of Key Bits Shifted per Round

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Number	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

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, as determined by the chart above.

DES Example: Subkeys Step 2

After being shifted, 48 out of the 56 bits are selected. Because this operation permutes the order of the bits as well as selects a subset of bits, it is called a **compression permutation**. This operation provides a subset of 48 bits.

Table 12.4
Compression Permutation

14,	17,	11,	24,	1,	5,	3,	28,	15,	6,	21,	10,
23,	19,	12,	4,	26,	8,	16,	7,	27,	20,	13,	2,
41,	52,	31,	37,	47,	55,	30,	40,	51,	45,	33,	48,
44,	49,	39,	56,	34,	53,	46,	42,	50,	36,	29,	32

For example, the bit in position 14 of the shifted key moves to position 1 of the output, and the bit in position 18 of the shifted key is ignored.

DES Example: Expansion Permutation

The right half of the data is now expanded from 32 bits to 48 bits in what is called the **expansion permutation**. This operation makes the right half the same size as the key for the XOR operation. This expansion also provides an **avalanche effect** because it allows one bit to affect two substitutions.

Table 12.5
Expansion Permutation

32,	1,	2,	3,	4,	5,	4,	5,	6,	7,	8,	9,
8,	9,	10,	11,	12,	13,	12,	13,	14,	15,	16,	17,
16,	17,	18,	19,	20,	21,	20,	21,	22,	23,	24,	25,
24,	25,	26,	27,	28,	29,	28,	29,	30,	31,	32,	1

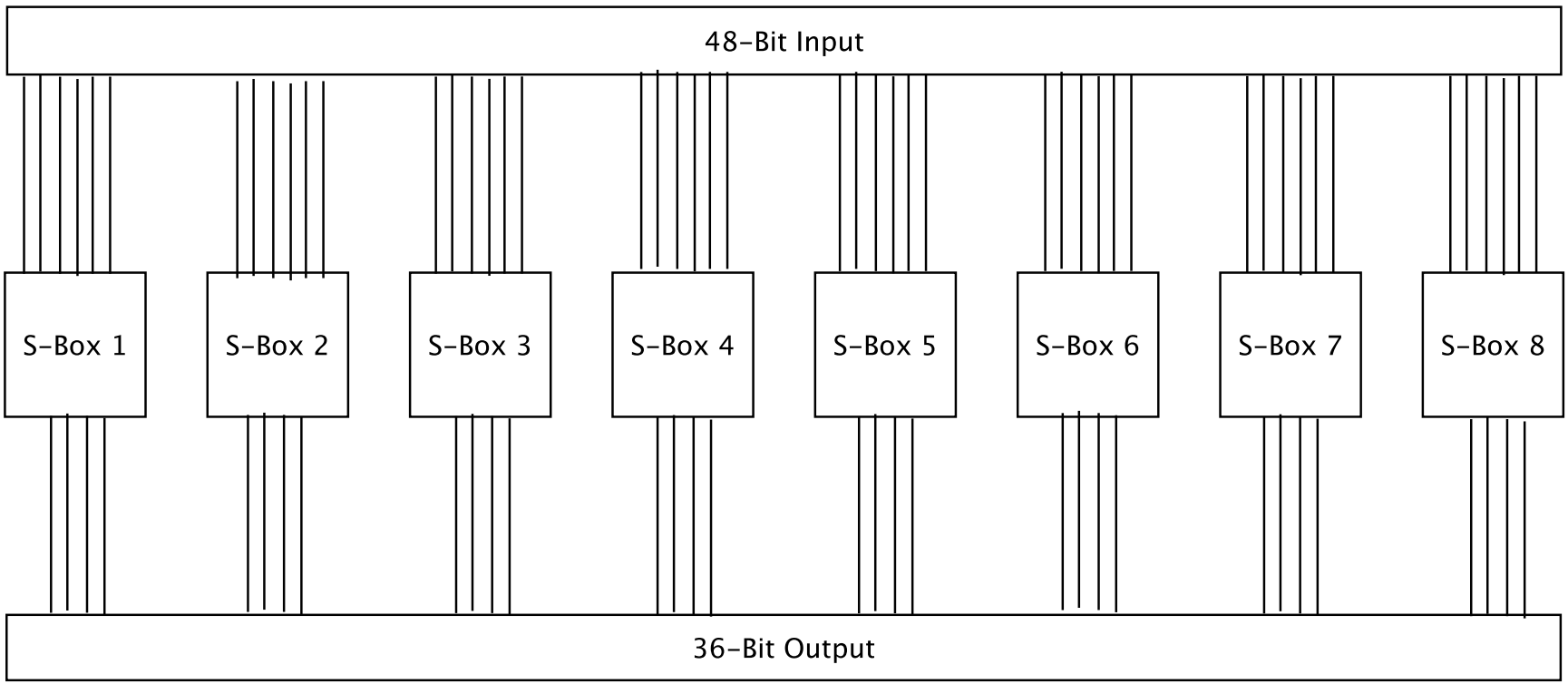
For example, the bit in position 3 of the input block moves to position 4 of the output block, and the bit in position 21 of the input block moves to positions 30 and 32 of the output block.

DES Example: Combine with XOR

The compressed key is XOR-ed with the expanded right block.

DES Example: S-box

Substitutions are computed by eight **substitution boxes**, or **S-boxes**. The S-box has input of 6 bits and output of 4 bits. The 48 bits of the block are divided into eight 6-bit sub-blocks, and each sub-block is operated on by a different S-box.



DES Example: S-Box

Each S-box is a table of 4 rows and 16 columns. Each entry in the box is a 4-bit number. The 6 input bits of the S-box specify under which row and column number to look for the output. Below are examples of what the S-boxes could look like.

S-box 4:

7,	13,	14,	3,	0,	6,	9,	10,	1,	2,	8,	5,	11,	12,	4,	15,
13,	8,	11,	5,	6,	15,	0,	3,	4,	7,	2,	12,	1,	10,	14,	9,
10,	6,	9,	0,	12,	11,	7,	13,	15,	1,	3,	14,	5,	2,	8,	4,
3,	15,	0,	6,	10,	1,	13,	8,	9,	4,	5,	11,	12,	7,	2,	14,

S-box 5:

2,	12,	4,	1,	7,	10,	11,	6,	8,	5,	3,	15,	13,	0,	14,	9,
14,	11,	2,	12,	4,	7,	13,	1,	5,	0,	15,	10,	3,	9,	8,	6,
4,	2,	1,	11,	10,	13,	7,	8,	15,	9,	12,	5,	6,	3,	0,	14,
11,	8,	12,	7,	1,	14,	2,	13,	6,	15,	0,	9,	10,	4,	5,	3,

DES Example: S-box

Say that $b_1 b_2 b_3 b_4 b_5 b_6$ is the input for an S-box. We create a 2-bit number $b_1 b_6$ from 0 to 3, which corresponds to a row in the table. The middle 4 bits $b_2 b_3 b_4 b_5$ form a 4- bit number, from 0 to 15, specifying the column.

S-box 6:

12,	1,	10,	15,	9,	2,	6,	8,	0,	13,	3,	4,	14,	7,	5,	11,
10,	15,	4,	2,	7,	12,	9,	5,	6,	1,	13,	14,	0,	11,	3,	8,
9,	14,	15,	5,	2,	8,	12,	3,	7,	0,	4,	10,	1,	13,	11,	6,
4,	3,	2,	12,	9,	5,	15,	10,	11,	14,	1,	7,	6,	0,	8,	13,

Say we input `110011` to S-box 6 above. The first and last bits combine to form `11` (row 3). The middle 4 bits form `1001` (column 9) The entry under row 3, column 9 of S-box 6 is 14. The value 14 $=$ `1110` is substituted for `110011`.

DES Example: S-box

The resulting 4-bit blocks are recombined into a 32-bit block.

DES Example: P-Box

The 32-bit block we are left with is now permuted according to a **P-box**.

Table 12.7
P-Box Permutation

16,	7,	20,	21,	29,	12,	28,	17,	1,	15,	23,	26,	5,	18,	31,	10,
2,	8,	24,	14,	32,	27,	3,	9,	19,	13,	30,	6,	22,	11,	4,	25

We XOR the result with the left half of the initial 64-bit block, then switch the left and right blocks. Another round then begins. Repeat 16 times.

DES Example: Final Permutation

The final permutation is the inverse of the initial permutation.

Table 12.8
Final Permutation

40,	8,	48,	16,	56,	24,	64,	32,	39,	7,	47,	15,	55,	23,	63,	31,
38,	6,	46,	14,	54,	22,	62,	30,	37,	5,	45,	13,	53,	21,	61,	29,
36,	4,	44,	12,	52,	20,	60,	28,	35,	3,	43,	11,	51,	19,	59,	27,
34,	2,	42,	10,	50,	18,	58,	26,	33,	1,	41,	9,	49,	17,	57,	25

DES Example: Decryption

Decryption is the same as encryption, but the keys are used in reverse order and the key shift is a right shift.

References

- *Applied Cryptography* By Schneier, Chapter 12