Data Encryption Standard (DES)

Math 4175

§4.5. DES Timeline

History of DES Timeline:

- 1973: The National Bureau of Standards (NBS) solicits submissions for a cryptosystem to become a national standard.
- 1974: IBM submits an algorithm called LUCIFER.
- LUCIFER is reviewed by the NSA, which suggests some modifications.
- 1975: NBS releases the modified algorithm as the Data Encryption Standard (DES).
- 1977: DES is adopted as the standard algorithm for non-classified sensitive data.
- 1977-2002 DES is widely used in government and commercial applications.

§4.5. Data Encryption Standard

The Data Encryption Standard (DES) is a 16 round Feistel Cipher.

DES Specifications:

- Both plaintexts and ciphertexts are 64-bit blocks.
- Keys are represented as 64-bit strings.
- Bits in position $8, 16, 24, 32, 40, 48, 56, 64$ of the key are called parity bits and will not be used.
- The effective key length is therefore 56 bits.
- Prior to 16 rounds of encryption, there is a fixed initial permutation **IP** that is applied to the plaintext to get $\mathbf{IP}(x) = L^0R^0$.
- After 16 rounds of encryption, the inverse permutation $\mathbf{IP}^{-1}$ is applied to $R^{16}L^{16}$ to get the ciphertext y, that is, $\mathbf{IP}^{-1}(R^{16}L^{16}) = y$ (note that L^{16} and R^{16} are swapped).

§4.5. Data Encryption Standard

Overview of DES Algorithm:

- Input plaintext x.
- Apply a fixed initial permutation **IP** to obtain $\mathbf{IP}(x) = u^0 = L^0R^0$.
- For $1 \leq i \leq 16$, do

$$\begin{aligned} L^i &= R^{i-1} \\ R^i &= L^{i-1} \oplus f(R^{i-1}, K^i) \end{aligned}$$

  (This is the round function where f is bit more complicated).
- Let $y = \mathbf{IP}^{-1}(R^{16}L^{16})$. (note that L^{16} and R^{16} are swapped before $\mathbf{IP}^{-1}$ is applied).
- Output y.

$ 4.5 Initial Permutation

The initial permutation **IP** is frequently represented in the following form:

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

This notation should be interpreted in the following fashion:

- the 1st bit of $\mathbf{IP}(x)$ is the 58th bit of the plaintext x,
- the 2nd bit of $\mathbf{IP}(x)$ is the 50th bit of x,
- the 9th bit of $\mathbf{IP}(x)$ is the 60th bit of x,
- the 64th bit of $\mathbf{IP}(x)$ is the 7th bit of x, and so on.

§4.5. Key Schedule

The round keys $K^1, K^2, \ldots, K^{16}$ (collectively called the key schedule) are derived from the 64-bit DES key, K. An example of key schedule is described below:

Step 1: The parity bits $8, 16, 24, \ldots, 64$ of K are discarded and the fifty six non-parity bits of K are permuted and split up, using the permutations C and D (see below), to form two 28-bit subkeys, $C_0 = C(K)$ and $D_0 = D(K)$.

Notice that PC-1 (in the next slide) does not contain any numbers that are congruent to 0 modulo 8. This is because, as previously discussed, the bits in positions $8, 16, 24, 32, 40, 48, 56$, and 64 are parity bits and do not form part of the actual key.

§4.5. Key Schedule

PC-1: the permutations C and D

The permutations C and D are collectively given the name PC-1, for *permutation choice 1*.

$$C = \begin{pmatrix} 57 & 49 & 41 & 33 & 25 & 17 & 9 \\ 1 & 58 & 50 & 42 & 34 & 26 & 18 \\ 10 & 2 & 59 & 51 & 43 & 35 & 27 \\ 19 & 11 & 3 & 60 & 52 & 44 & 36 \end{pmatrix}$$

$$D = \begin{pmatrix} 63 & 55 & 47 & 39 & 31 & 23 & 15 \\ 7 & 62 & 54 & 46 & 38 & 30 & 22 \\ 14 & 6 & 61 & 53 & 45 & 37 & 29 \\ 21 & 13 & 5 & 28 & 20 & 12 & 4 \end{pmatrix}$$

This notation should be read in the same way as that used for the initial permutation, IP, i.e., the first bit of $C_0 = C(K)$ is the 57th bit of K, the 2nd bit of C_0 is the 49th bit of K, the 8th bit of $D_0 = D(K)$ is the 7th bit of K, and so on.

§4.5. Key Schedule

Example(Δ): Suppose that

$$K = 00100101\ 01100111\ 11001101\ 10110011$$

$$11111101\ 11001110\ 01000000\ 00101010$$

Then

$$C_0 = C(K) = 00111100\ 01110110\ 10011011\ 0001$$

$$D_0 = D(K) = 10101010\ 00110111\ 10110100\ 1000$$

§4.5. Key Schedule

Step 2: Finding the subkeys $C_1, C_2, \ldots, C_{16}$ and $D_1, D_2, \ldots, D_{16}$

Let X be a string and ℓ a positive integer. Denote by $L(X, \ell)$ the string obtained from X by shifting each character of X left ℓ positions; the ℓ leftmost characters of X are "cycled around" so that they reappear (in the same order) on the right hand side.

For instance, if $B_1 = abcde$, then $L(B_1, 2) = cdeab$.

Similarly, if $B_2 = 00110111$, then $L(B_2, 3) = 10111001$.

§4.5. Key Schedule

Once we have determined C_0 and D_0, the other subkeys $C_1, C_2, \ldots, C_{16}$ and $D_1, D_2, \ldots, D_{16}$ are found from them using the formula

$$C_i = L(C_{i-1}, \ell_i)$$

$$D_i = L(D_{i-1}, \ell_i)$$

where the values ℓ_i, $1 \leq i \leq 16$, are given in the following table.

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

§4.5. Key Schedule

We continue from Example(Δ).

We must first find C_1; from the preceding table.

We see that $C_1 = L(C_0, \ell_1) = L(C_0, 1)$.

Since $C_0 = 00111100011101101001101100001$, we have

$$C_1 = 01111000111011010011011000010.$$

Similarly, $D_1 = L(D_0, 1)$.

Since $D_0 = 10101010001101111011010010000$,

$$D_1 = 01010100011011110110100100001.$$

Continuing in this fashion, we find the following:

i	C_i	D_i
0	00111100011101101001101100 01	10101010001101111011010010 00
1	01111000111011010011011000 10	01010100011011110110100100 01
2	11110001110110100110110001 00	10101000110111101101001000 10
3	11000111011010011011000100 11	10100011011110110100100010 10
4	00011101101001101100010011 11	10001101111011010010001010 10
5	01110110100110110001001111 00	00110111101101001000101010 10
6	11011010011011000100111100 01	11011110110100100010101000 10
7	01101001101100010011110001 11	01111011010010001010100011 00
8	10100110110001001111000111 01	11101101001000101010001101 10
9	01001101100010011110001110 11	11011010010001010100011011 00
10	00110110001001111000111010 11	01101001000101010001101111 00
11	11011000100111100011101010 00	10100100010101010001101111 01
12	01100010011110001110110100 11	10010001010101000110111101 10
13	10001001111000111011010011 01	01000101010100011011110101 00
14	00100111100011101101001101 10	00010101010001101111011010 01
15	10011110001110110100110110 00	01010101000110111101101001 00
16	00111100011101101001101100 01	10101010001101111011010010 00

§3.5. Key Schedule

Step 3: Finding the round keys K^i from the subkeys C_i, D_i

For each i with $1 \leq i \leq 16$, the round key K^i is found by feeding $C_i||D_i$ to the permutation **PC2** (see below).

$$\mathbf{PC2} = \begin{pmatrix} 14 & 17 & 11 & 24 & 1 & 5 \\ 3 & 28 & 15 & 6 & 21 & 10 \\ 23 & 19 & 12 & 4 & 26 & 8 \\ 16 & 7 & 27 & 20 & 13 & 2 \\ 41 & 52 & 31 & 37 & 47 & 55 \\ 30 & 40 & 51 & 45 & 33 & 48 \\ 44 & 49 & 39 & 56 & 34 & 53 \\ 46 & 42 & 50 & 36 & 29 & 32 \end{pmatrix}$$

The round keys $K^1, K^2, \ldots, K^{16}$ are given by $K^i = \mathbf{PC2}(C_i||D_i)$.

Note that since C_i and D_i are twenty eight bits each, $C_i||D_i$ is fifty six bits. The key K^i is, however, only forty eight bits.

§4.5. Key Schedule

Continuing our example(Δ):

The round key K^3 is equal to $\mathbf{PC2}(C_3||D_3)$.

Since $C_3||D_3 =$

$$11000111011010011011000100111010001101111011010010001010,$$

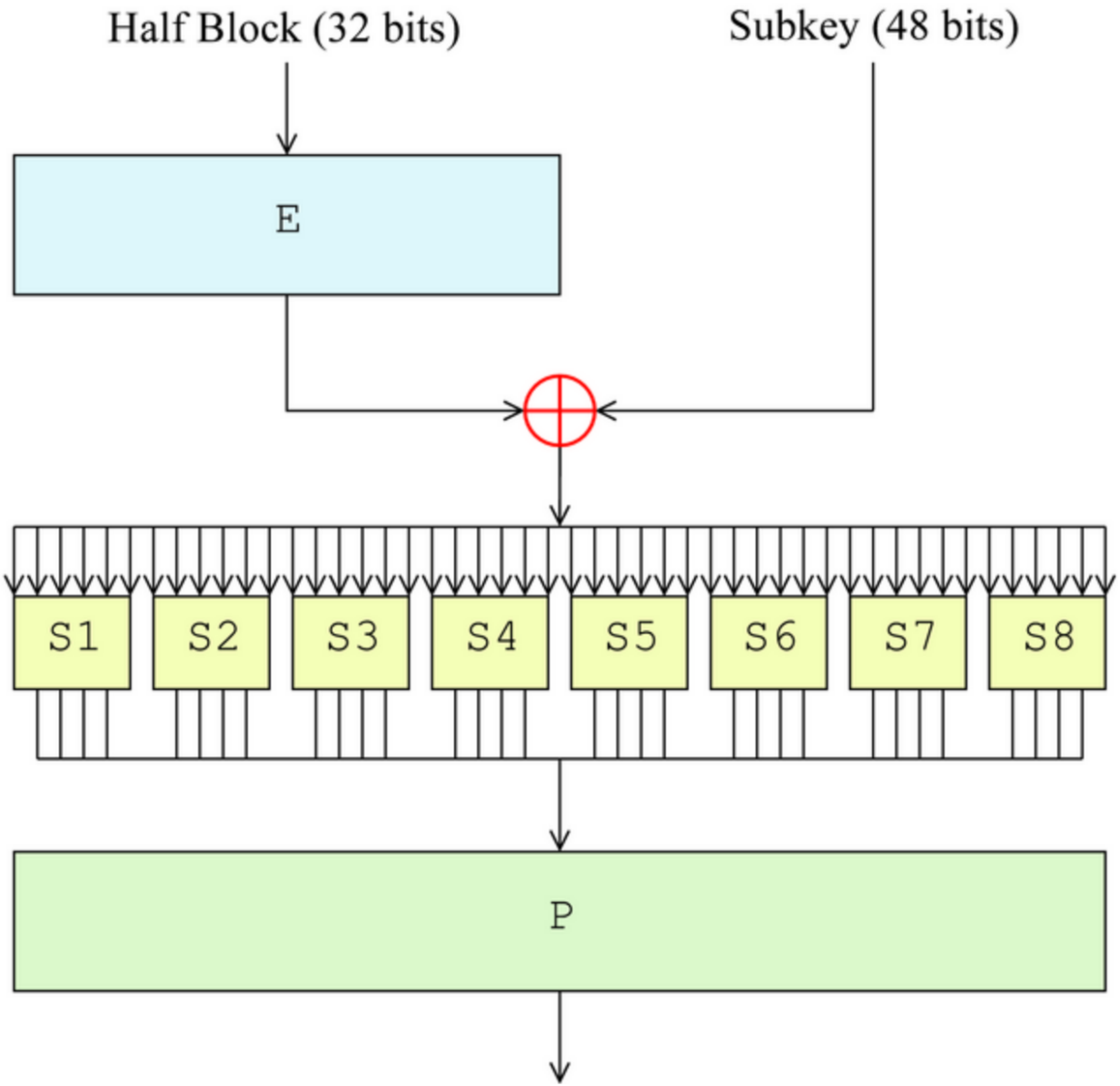
we find

$$K^3 = 011110010101010001111111101001010000111001100110.$$

§4.5. Round Function

- Most of the cryptographic content lies in the function $f(R^{i-1}, K^i)$.
- We describe it in several steps:
  (1) First R^{i-1} is expanded to 48 bits, by the "expansion permuation" E.
  (2) Compute $E(R^{i-1}) \oplus K^i$ and write it as $B_1 B_2 \dots B_8$, where each B_j has 6 bits.
  (3) There are eight S-boxes $S_1, S_2, \dots, S_8$. The S-boxes each takes a 6-bit string as input and yields an output as a 4-bit string.
  (4) For $1 \leq j \leq 8$, we let $C_j = S_j(B_j)$. This way we obtain a 32-bit string $C_1 C_2 \dots C_8$.
  (5) The index permutation P is applied to the above string to obtain $f(R^{i-1}, K^i)$.

The DES internal function f



§4.5. Round Function

(1) The expansion matrix E:

$$E = \begin{pmatrix} 32 & 1 & 2 & 3 & 4 & 5 \\ 4 & 5 & 6 & 7 & 8 & 9 \\ 8 & 9 & 10 & 11 & 12 & 13 \\ 12 & 13 & 14 & 15 & 16 & 17 \\ 16 & 17 & 18 & 19 & 20 & 21 \\ 20 & 21 & 22 & 23 & 24 & 25 \\ 24 & 25 & 26 & 27 & 28 & 29 \\ 28 & 29 & 30 & 31 & 32 & 1 \end{pmatrix}$$

The notation is the same as for the initial permutation.

E expands R^{i-1} from 32 bits to 48 bits, permuted in a certain way with 16 of the bits appearing twice.

(3) The S-boxes $S_1, S_2, \ldots, S_8$. The eight S-boxes are:

S_1															
14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2															
15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3															
10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_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_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

S_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

§4.5. Round Function

(3) The S-boxes $S_1, S_2, \ldots, S_8$. The eight S-boxes are:

S_7															
4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8															
13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Each S-box maps six bits to four bits as given below:

§4.5. Round Function

(4) Computing C_j:

Given a bitstring of length six, say $B = b_1 b_2 b_3 b_4 b_5 b_6$, we compute $S_j(B)$ as follows. The two bits $b_1 b_6$ determine the binary representation of the row r of S_j (where $0 \leq r \leq 3$), and the four bits $b_2 b_3 b_4 b_5$ determine the binary representation of the column c of S_j ($0 \leq c \leq 15$). Then $S_j(B)$ is defined to be the entry $S_j(r, c)$, written in binary as a bitstring of length four. In this fashion we compute $C_j = S_j(B_j)$.

Example: Consider the binary 6-tuple input 101000 in S_1-box. The first and last bits are 10, which is the binary representation of the integer 2. The middle four bits are 0100, which is the integer 4. Now 2 corresponds to row 3 (since rows are numbered 0, 1, 2, 3); similarly 4 corresponds to fifth column. Now the 3rd row and 5th column of S_1 is 13, which is 1101 in binary. Therefore 1101 is the output of the box S_1 given the input 101000.

§4.5. Round Function

(5) The index permutation P:

The 32-bit string $C_1||C_2||\cdots||C_8$ is permuted in the following fashion to produce $f(R^{i-1}, K^i)$:

$$P = \begin{pmatrix} 16 & 7 & 20 & 21 & 29 & 12 & 28 & 17 \\ 1 & 15 & 23 & 26 & 5 & 18 & 31 & 10 \\ 2 & 8 & 24 & 14 & 32 & 27 & 3 & 9 \\ 19 & 13 & 30 & 6 & 22 & 11 & 4 & 25 \end{pmatrix}$$

The notation for P is what you would expect, i.e., if we let $X = C_1||C_2||\cdots||C_8$, then

- the first bit of $P(X)$ is the 16th bit of X,
- the second bit of $P(X)$ is the 7th bit of X,
- and so on.

§4.5. DES Timeline II

- 1992: Biham and Shamir publish a differential attack on DES:
  - with lower time complexity than brute force
  - requiring unreasonably large number of chosen plaintexts.
- 1994: Matsui publishes linear cryptanalysis
  - Attack required about 1 trillion PT-CT pairs.
  - At the time, it required 40 days to generate the pairs and 10 days to find the key.
- 1997: RSA Security publishes the DES Challenge, a decryption contest with a $10,000 prize.
- Involved single CT-PT pair. Challenge: find the key

§4.5. DES Timeline II

- The DES key is found 96 days later by DESCHALL.
  - Brute force attack ($2^{56} = 72,057,594,037,927,936 \approx 72$ quadrillion possible keys). Used a distributed approach with a total of 78,000 computers involved.
  - About one fourth of the keyspace had been searched when the key was found.
  - Message was "The secret message is: Many hands make light work."
  - Showed that DES could not be considered secure for protecting highly sensitive or valuable data.
- 1998 The Electronic Frontier Foundation (EFF) finds a DES key in 56 hours.
- 1999 EFF together with distributed.net find a DES key in 22 hours.
- Message was "See you in Rome (second AES Conference, March 22-23, 1999)".

§4.5. DES Timeline II

- 1997: Submissions are solicited for Advanced Encryption System (AES).
- 2000: AES is published.
- 2001: AES becomes effective.
- 2005: DES is withdrawn.