Simplified Data Encryption Standard (SDES)

Math 4175 instructional purpose only

§4.5. Simplified Data Encryption Standard

Before we learn the Data Encryption Standard (DES), which is a 16 round Feistel Cipher, we will learn a simplified 2 round version called SDES.

SDES Specifications:

- Both plaintexts and ciphertexts are 12-bit blocks.
- Key is represented as a 10-bit string.
- Prior to 2 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 2 rounds of encryption, the inverse permutation $\mathbf{IP}^{-1}$ is applied to R^2L^2 to get the ciphertext y, that is, $\mathbf{IP}^{-1}(R^2L^2) = y$ (note that L^2 and R^2 are swapped).

§4.5. Simplified Data Encryption Standard

Overview of SDES Algorithm:

- Input plaintext x.
- Apply a fixed initial permutation **IP** to obtain $\mathbf{IP}(x) = u^0 = L^0R^0$.
- For $1 \leq i \leq 2$, 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 lengthy and will be given soon).
- Let $y = \mathbf{IP}^{-1}(R^2L^2)$. (note that L^2 and R^2 are swapped before $\mathbf{IP}^{-1}$ is applied).
- Output y.

$ 4.5 Initial Permutation

The initial permutation **IP** is represented, for example, in the following form:

$$IP = \begin{pmatrix} 10 & 2 & 12 & 4 & 6 & 8 & 9 & 1 & 11 & 3 & 5 & 7 \end{pmatrix}$$

This notation should be interpreted in the following fashion:

- the 1st bit of $\mathbf{IP}(x)$ is the 10th bit of the plaintext x,
- the 2nd bit of $\mathbf{IP}(x)$ is the 2nd bit of x,
- the 9th bit of $\mathbf{IP}(x)$ is the 11th bit of x,
- the 12th bit of $\mathbf{IP}(x)$ is the 7th bit of x, and so on.

$$IP^{-1} = \begin{pmatrix} 8 & 2 & 10 & 4 & 11 & 5 & 12 & 6 & 7 & 1 & 9 & 3 \end{pmatrix}$$

§4.5. Key Schedule

The round keys K^1, K^2 (collectively called the key schedule) are derived from the 10-bit SDES key, K. An example of key schedule is described below:

Step 1: The 10 bits of K are permuted and split up, using the permutations C and D (see next slide), to form two 5-bit subkeys, $C_0 = C(K)$ and $D_0 = D(K)$.

§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*.

$$C = \begin{pmatrix} 9 & 1 & 10 & 2 & 3 \end{pmatrix}$$

$$D = \begin{pmatrix} 7 & 6 & 8 & 5 & 4 \end{pmatrix}$$

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 9th bit of K, the 2nd bit of C_0 is the 1st bit of K, the 5th bit of $D_0 = D(K)$ is the 4th bit of K, and so on.

§4.5. Key Schedule

Example(Δ): Suppose that

$$K = 0110010111$$

Then

$$C_0 = C(K) = 10111$$

$$D_0 = D(K) = 01100$$

§4.5. Key Schedule

Step 2: Finding the subkeys C_1, C_2 and D_1, D_2

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 $B_2 = 00110$, then $L(B_2, 2) = 11000$.

§4.5. Key Schedule

Once we have determined C_0 and D_0, the other subkeys C_1, C_2 and D_1, D_2 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, $1 \leq i \leq 2$, are pre-determined, say $(\ell_1, \ell_2) = (1, 2)$.

§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 $C_0 = 10111$, we have $C_1 = 01111$.

Similarly, $D_1 = L(D_0, 1)$. Since $D_0 = 01100$, $D_1 = 11000$.

Continuing in this fashion, we find the following:

i	C_i	D_i
0	10111	01100
1	01111	11000
2	11101	00011

§3.5. Key Schedule

Step 3: Finding the round keys K^i from the subkeys C_i, D_i
For each $i = 1, 2$, the round key K^i is found by feeding $C_i||D_i$ to the permutation

$$\textbf{PC2} = \begin{pmatrix} 5 & 2 & 6 & 3 & 7 & 4 & 9 & 8 \end{pmatrix}$$

Note that since C_i and D_i are five bits each, $C_i||D_i$ is ten bits. The key K^i is, however, only eight bits.

Continuing our example(Δ):

$$K^1 = 11111100$$

$$K^2 = 11010010$$

§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 8 bits, by the "expansion permuation" E.
  (2) Compute $E(R^{i-1}) \oplus K^i$ and write it as $B_1 B_2$, where each B_j has 4 bits.
  (3) There are two S-boxes S_1, S_2. The S-boxes each takes a 4-bit string as input and yields an output as a 3-bit string.
  (4) For $j = 1, 2$, we let $C_j = S_j(B_j)$. This way we obtain a 6-bit string $C_1 C_2$.
  (5) An index permutation P is applied to the above string to obtain $f(R^{i-1}, K^i)$.

§4.5. Round Function

(1) The expansion function E:
The 6-bits are expanded to 8-bits by using the expansion function:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \end{pmatrix}$$

$$\downarrow$$

$$\begin{pmatrix} 1 & 2 & 4 & 3 & 4 & 3 & 5 & 6 \end{pmatrix}$$

The notation is the same as for the initial permutation.

E expands R^{i-1} from 6 bits to 8 bits, permuted in a certain way with 2 of the bits appearing twice.

(2) $E(R^{i-1}) \oplus K^i$ is the bitwise addition key mix.

§4.5. Round Function

(3) The S-boxes S_1, S_2. The two S-boxes are:

S_1 box

5	2	1	6	3	4	7	0
1	4	6	2	0	7	5	3

S_2 box

4	0	6	5	7	1	3	2
5	3	0	7	6	2	1	4

§4.5. Round Function

(4) Computing C_j:

Each S-box maps 4-bits to 3-bits as given below: Given a 4- bits string, say $B = b_1 b_2 b_3 b_4$, we compute $S_j(B)$ as follows. The first bit b_1 determine the binary representation of the row r of S_j (where 0 for the first row and 1 for the second row), and the last three bits $b_2 b_3 b_4$ determine the binary representation of the column c of S_j $(0 \leq c \leq 7)$. Then $S_j(B)$ is defined to be the entry $S_j(r, c)$, written in binary as a 3-bit string. In this fashion we compute $C_j = S_j(B_j)$.

Example: Consider the binary 6-tuple input 1010 in S_1-box. The first bit is 1, which is the binary representation of the second row. The last 3-bits are 010, which is the integer 2. Now 2 corresponds to column 3 (since columns are numbered 0, 1, 2, 3 and so on); similarly 4 corresponds to fifth column and so on. Now the 2nd row and 3rd column of S_1 is 6, which is 110 in binary. Therefore 110 is the output of the box S_1 given the input 1010.

§4.5. Round Function

(5) The index permutation P:

The 6-bit string $C_1||C_2$ is permuted by the following permutation P to produce $f(R^{i-1}, K^i)$:

$$P = \begin{pmatrix} 5 & 2 & 4 & 1 & 6 & 3 \end{pmatrix}$$

§4.5. SDES Example

Suppose that plaintext input is 011011100110 and the key is $K = 0110010111$.

The key schedule from our example (Δ) yields:

$$K^1 = 11111100$$
$$K^2 = 11010010$$

By applying the initial permutation IP to the plaintext, we obtain:

$$\mathbf{IP}(x) = u^0 = 110010001111$$

Hence

$$L^0 = 110010$$
$$R^0 = 001111$$

§4.5. SDES Example

Round 1:

$$L^1 = R^0 = 001111$$

$$R^1 = L^0 \oplus f(R^0, K^1)$$

Therefore,

$$R^1 = 110010 \oplus f(R^0, K^1)$$

Now we need to compute $f(R^0, K^1)$.

§4.5. SDES Example

Computation of $f(R^0, K^1)$:

- $E(R^0) = E(001111) = 00111111$.
- $E(R^0) \oplus K^1 = 00111111 \oplus 11111100 = 11000011 = B_1 B_2$.
- $C_1 = S_1(B_1) = S_1(1100) = 000$.
- $C_2 = S_2(B_2) = S_2(0011) = 101$.
- $P(C_1 C_2) = P(000101) = 001010 = f(R^0, K^1)$.

Hence

$$L^1 = R^0 = 001111$$

$$R^1 = L^0 \oplus f(R^0, K^1) = 110010 \oplus 001010 = 111000$$

§4.5. SDES Example

Round 2:

$$L^2 = R^1 = 111000$$

$$R^2 = L^1 \oplus f(R^1, K^2)$$

Therefore,

$$R^2 = 001111 \oplus f(R^1, K^2)$$

Now we need to compute $f(R^1, K^2)$.

§4.5. SDES Example

Computation of $f(R^1, K^2)$:

- $E(R^1) = E(111000) = 11010100$.
- $E(R^1) \oplus K^2 = 11010100 \oplus 11010010 = 00000110 = B_1B_2$.
- $C_1 = S_1(B_1) = S_1(0000) = 101$.
- $C_2 = S_1(B_2) = S_1(0110) = 011$.
- $P(C_1C_2) = P(101011) = 100111 = f(R^1, K^2)$.

Hence

$$L^2 = R^1 = 111000$$

$$R^2 = 001111 \oplus 100111 = 101000$$

§4.5. SDES Example

- Now $R^2L^2 = 101000111000$
- So the ciphertext is $y = IP^{-1}(R^2L^2) = 100000001111$
- **Exercise:** Encrypt: 1111 0111 0100 with the same key.