Data Encryption Standard

Cryptography

School of Engineering and Technology
CQUniversity Australia

Prepared by Steven Gordon on 04 Jan 2022,
des.tex, r1966

Contents

Overview of the Data Encryption Standard (DES)

Simplified-DES

Details of DES

DES in OpenSSL

DES in Python

Data Encryption Standard

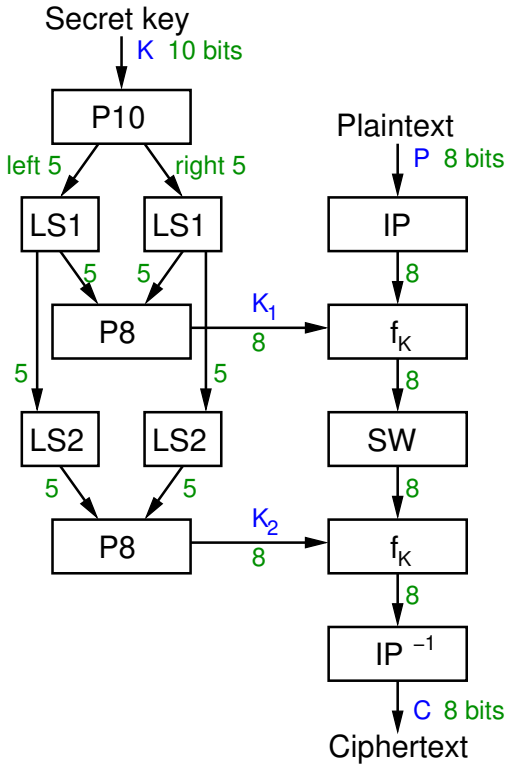
- Symmetric block cipher
- 56-bit key, 64-bit input block, 64-bit output block
- Developed in 1977 by NIST; designed by IBM (Lucifer) with input from NSA
- Principles used in other ciphers, e.g. 3DES, IDEA

Contents

Simplified DES

- Input (plaintext) block: 8-bits
- Output (ciphertext) block: 8-bits
- Key: 10-bits
- Rounds: 2
- Round keys generated using permutations and left shifts
- Encryption: initial permutation, round function, switch halves
- Decryption: Same as encryption, except round keys used in opposite order

S-DES Key Generation and Encryption

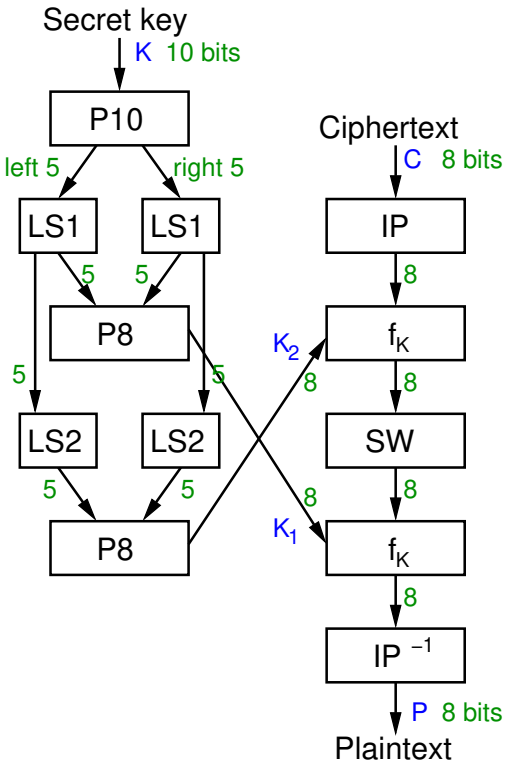


The figure on slide 6 shows the key generation and encryption steps of S-DES. Key generation, shown on the left, is used to generate round keys and is the same algorithm when used for both encryption and decryption. That is, the encrypter and decrypter will generate the exact same round keys.

The encrypter started with a shared secret key 10 bits long and 8 bits of plaintext. Two sub-keys, or round keys, K_1 and K_2 are generated using the key generation steps, which involve Permutations and Left Shifts.

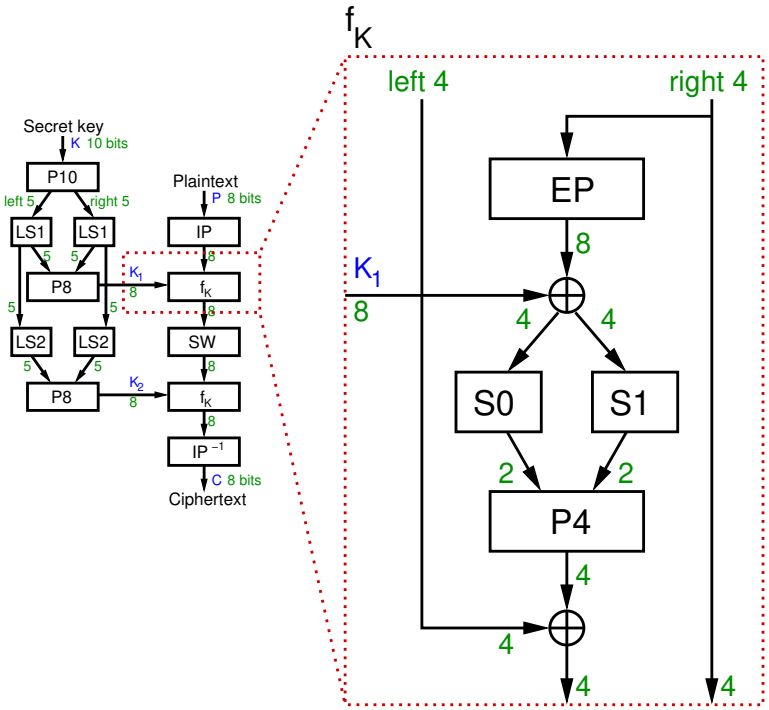
Encryption applies an Initial Permutation, then a round function f_k (with details to be shown shortly), SWaps the two halves of the 8 bit output, then reapplies the round function, but using the 2nd round key as input. Encryption ends with the inverse of the Initial Permutation.

S-DES Key Generation and Decryption



The figure on slide 7 shows the key generation and decryption. Decryption is in fact identical to encryption, except the round keys are used in the opposite order. That is, for encryption round key K_1 is used first, then round key K_2. For decryption, K_2 is used first and then K_1.

S-DES Round Function Details



The figure on slide 8 shows the details of the round function, f_k. Note that the same steps are applied in the 2nd round, but instead K_2 is used as the round key. Operations include Expand and Permutate, XOR, S-boxes and a Permutation of 4 bits. The 8 bits output (left half and right half) are then input the the SWap block (swapping the two halves).

Definitions of the permutations and S-boxes follow.

S-DES Permutations (definition)

Permutations used in S-DES:

P10 (permutate)

```
Input :  1 2 3 4 5 6 7 8 9 10
Output:  3 5 2 7 4 10 1 9 8 6
```

P8 (select and permutate)

```
Input :  1 2 3 4 5 6 7 8 9 10
Output:  6 3 7 4 8 5 10 9
```

P4 (permutate)

```
Input :  1 2 3 4
Output:  2 4 3 1
```

EP (expand and permutate)

```
Input :  1 2 3 4
Output:  4 1 2 3 2 3 4 1
```

IP (initial permutation)

```
Input :  1 2 3 4 5 6 7 8
Output:  2 6 3 1 4 8 5 7
```


As an example, permutation P4 takes a 4-bit input and produces a 4-bit output. The 1st bit of the input becomes the 4th bit of the output. The 2nd bit of the input becomes the 1st bit of the output. The 3rd bit of the input becomes the 3rd bit of the output. The 4th bit of the input becomes the 1st bit on the output.

The permutations are fixed. That is they are always these exact permutations, and known by the encrypter, decrypter and attacker.

Other Operations in S-DES

- LS-1: left shift by 1 position
- LS-2: left shift by 2 positions
- IP^{-1}: inverse of IP, such that $X = \text{IP}^{-1}(\text{IP}(X))$
- SW: swap the halves
- f_K: a round function using round key K
- F: internal function in each round

S-DES S-Boxes (definition)

S-Box considered as a matrix: input used to select row/column; selected element is output

4-bit input: $bit_1, bit_2, bit_3, bit_4$

$bit_1 bit_4$ specifies row (0, 1, 2 or 3 in decimal)

$bit_2 bit_3$ specifies column

$$S0 = \begin{bmatrix} 01 & 00 & 11 & 10 \\ 11 & 10 & 01 & 00 \\ 00 & 10 & 01 & 11 \\ 11 & 01 & 11 & 10 \end{bmatrix} \quad S1 = \begin{bmatrix} 00 & 01 & 10 & 11 \\ 10 & 00 & 01 & 11 \\ 11 & 00 & 01 & 00 \\ 10 & 01 & 00 & 11 \end{bmatrix}$$

Encrypt with S-DES (exercise)

Show that when the plaintext 01110010 is encrypted using S-DES with key 1010000010 that the ciphertext obtained is 01110111.

1. Rearrange K using P10: 1000001100
2. Left shift by 1 position both the left and right halves: 00001 11000
3. Rearrange the halves with P8 to produce K_1: 10100100
4. Left shift by 2 positions the left and right halves: 00100 00011
5. Rearrange the halves with P8 to produce K_2: 01000011

1. Apply the initial permutation, IP, on P: 10101001

S-DES Summary

- Educational encryption algorithm
- S-DES expressed as functions:

$$\text{ciphertext} = \text{IP}^{-1}(f_{K_2}(\text{SW}(f_{K_1}(\text{IP}(\text{plaintext})))))$$

$$\text{plaintext} = \text{IP}^{-1}(f_{K_1}(\text{SW}(f_{K_2}(\text{IP}(\text{ciphertext})))))$$

- Brute force attack on S-DES is easy since only 10-bit key
- If know plaintext and corresponding ciphertext, can we determine key? *Very hard*

S-DES Compared to Real DES

- S-DES vs DES
- Block size: 8 bits vs 64 bits
- Rounds: 2 vs 16
- IP: 8 bits vs 64 bits
- F: 4 bits vs 32 bits
- S-Boxes: 2 vs 8
- Round key: 8 bits vs 48 bits

The following section presents the details of DES. This is primarily for reference (or as evidence of the similarities and differences with S-DES). You are not expected to know the details of the DES operations.

Contents

Overview of the Data Encryption Standard (DES)

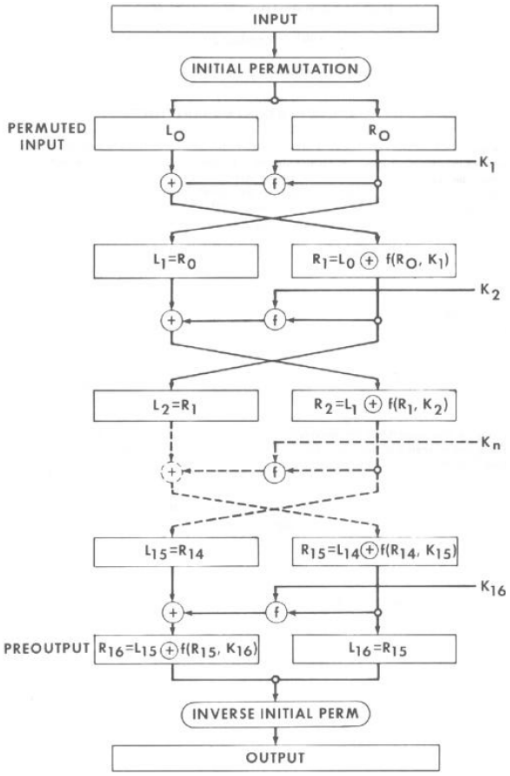
Simplified-DES

Details of DES

DES in OpenSSL

DES in Python

General DES Encryption Algorithm



The figure on slide 18 shows the overall steps in DES encryption. The details of each block are shown in the following.

Initial Permutation Tables for DES

IP

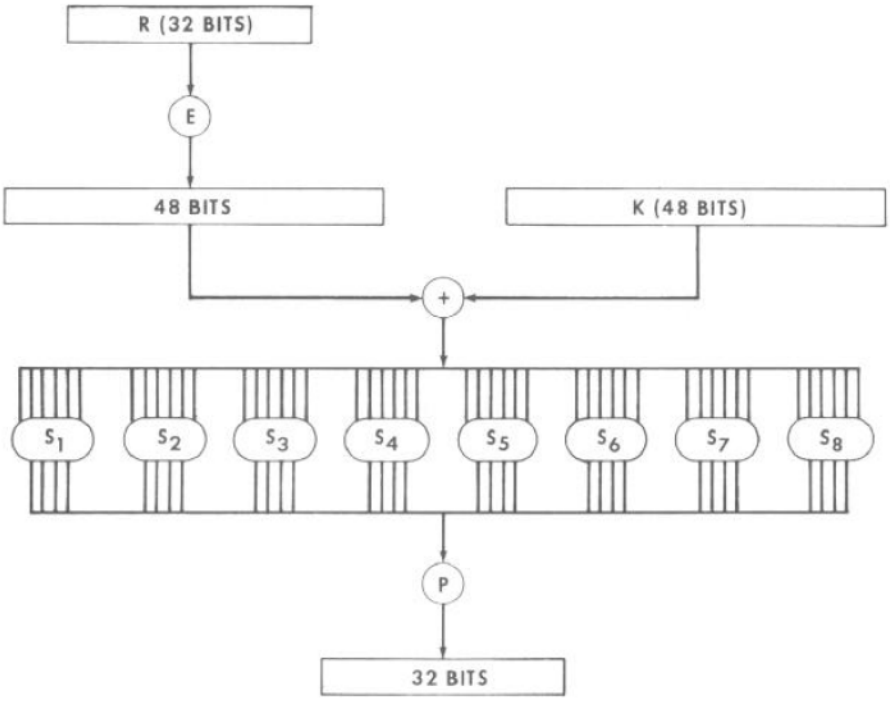
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

IP^{-1}

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

The figure on slide 19 shows the initial permutation and it's inverse. The table is read row-by-row. So the 58th input bit becomes the 1st output bit. The 50th input bit becomes the 2nd output bit. And the 7th input bit becomes the 64th output bit.

Calculation of F(R,K)



The figure on slide 20 shows the details of a single round of encruption, i.e. the round function. Similar to S-DES, it takes the right half, applies an expand and permutate (E), XOR with the round key, applies S-Boxes, and then a final permutate (P).

Permutation Tables for DES

E BIT-SELECTION TABLE

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

P

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

The figure on slide 21 shows E and P which are used within a round of DES.

Definition of DES S-Boxes 1 to 4

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

The figure on slide 22 shows the first 4 S-Boxes. Each S-Box takes a 6 bit input. The first and last bit are used to determine the row, and the middle 4 bits determine the column. The result is a decimal values within the range 0 to 15, which determines the 4 bit output. See `https://en.wikipedia.org/wiki/DES_supplementary_material` for an example of reading the S-Boxes.

Definition of DES S-Boxes 5 to 6

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

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

The figure on slide 23 shows the last 4 S-Boxes.

DES Permutated Choice 1 and 2

PC-1

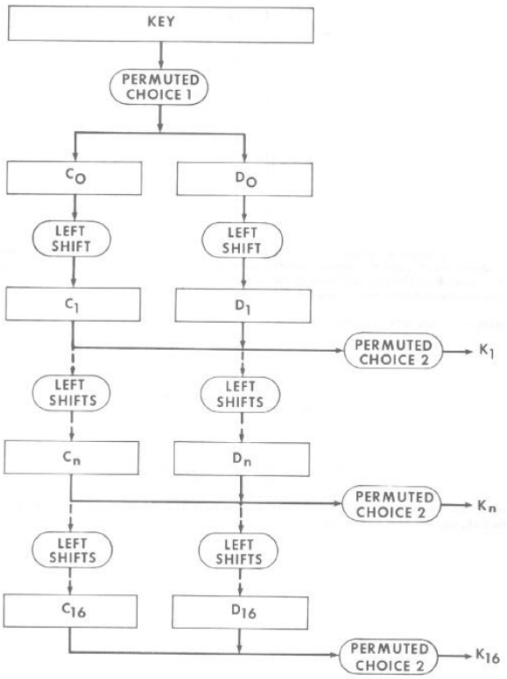
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

PC-2

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

The figure on slide 24 shows the Permutated Choices used in key generation.

DES Key Generation Schedule



The figure on slide 25 shows the overall key generation steps.

DES Schedule of Left Shifts in Key Generation

Iteration Number	Number of Left Shifts
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

The figure on slide 26 shows the schedule of left shifts indicating how many bits are shifted left when a Left Shift is applied in each round for key generation.

Contents

DES Encryption in OpenSSL

- Encrypt a file with a password using the `enc` operation
- Generate a random key using the `rand` operation
- Disable padding (with exact plaintext correct size)
- Encrypt with key and IV using `enc` operation
- View binary data (e.g. ciphertext) with `xxd`

DES Key Generation (exercise)

Generate a shared secret key to be used with DES and share it with another person.

DES Encryption (exercise)

Create a message in a plain text file and after using DES, send the ciphertext to the person you shared the key with.

DES Decryption (exercise)

Decrypt the ciphertext you received.

Contents

AES in Python Cryptography Library

- cryptography.io/en/latest/hazmat/primitives/symmetric-encryption/