The Data Encryption Standard in Detail

Cunsheng Ding
Department of Computer Science
Hong Kong University of Science and Technology
Clearwater Bay, Kowloon, Hong Kong, CHINA

The Data Encryption Standard in Detail

About this reading material

Although DES came to an end in 2000, its design idea is used in many block ciphers. This is a lecture on technical details of the Data Encryption Standard. It has three parts.

Part 1: The Structure of the DES

- It is a block cipher with key length 56 bits.

- It was designed by IBM in 1976 for the National Bureau of Standards (NBS), with approval from the National Security Agency (NSA).

- It had been used as a standard for encryption until 2000. From 2001 the AES will replace DES.

- After 25 years of analysis, the only security problem with DES found is that its key length is too short.

- Although its wide spread use came to an end, its design idea is still used in most block ciphers.

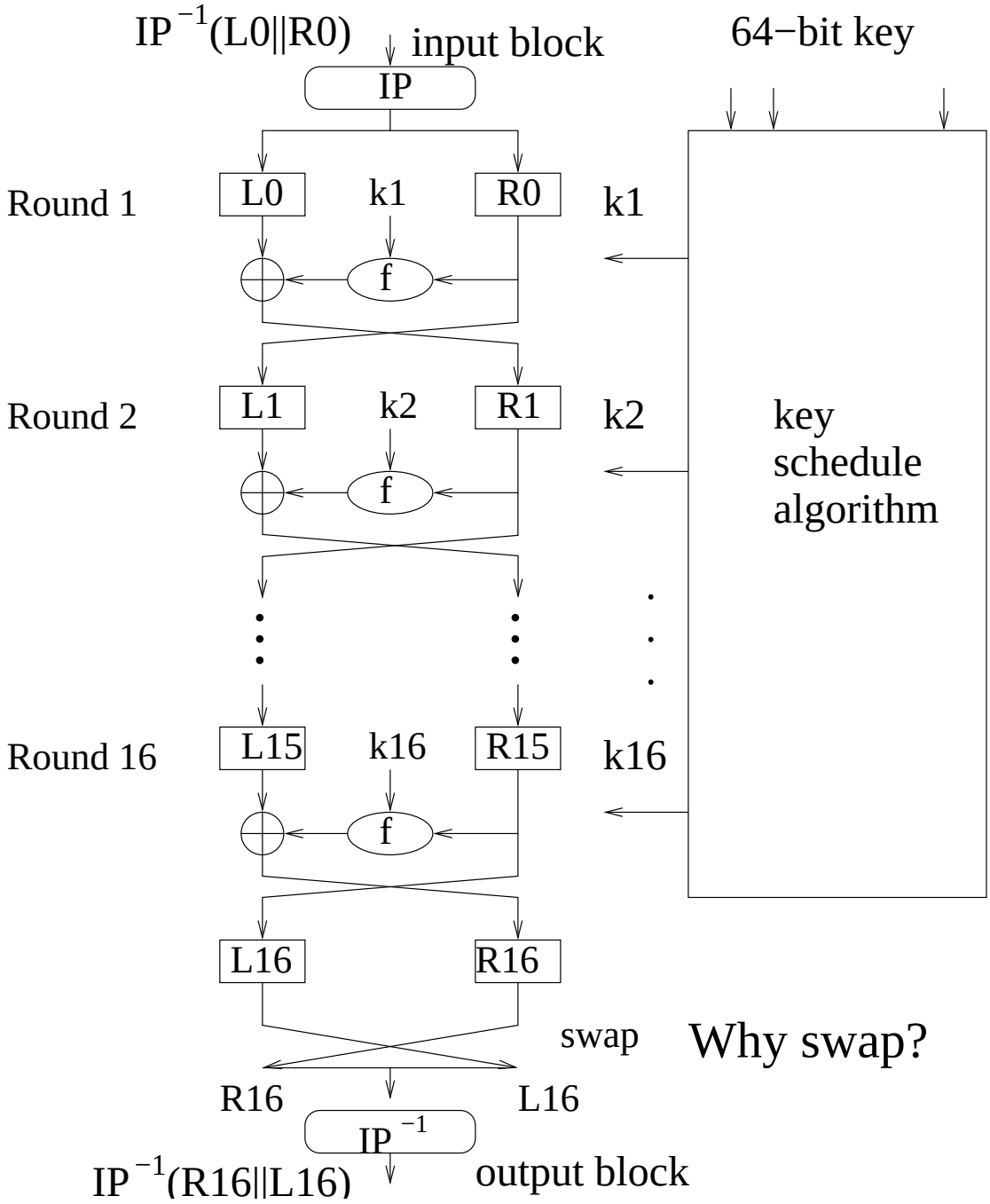
Building Blocks of the DES

- $\mathcal{M} = \mathcal{C} = \{0,1\}^*$ be the set of all finite binary strings.

- $\mathcal{K} = \{0,1\}^{56}$. A 56-bit key k is fed into a subkey generating algorithm to produce 16 round subkeys $k_1, \cdots, k_{16}$ of length 48 bits each.

- With a function $f(x,k)$ from $\{0,1\}^{32} \times \{0,1\}^{48}$ to $\{0,1\}^{32}$, the encryption is carried out as in the following figure.

The Encryption of DES



Encryption of the DES

1. Plaintext is broken into blocks of length 64 bits. Encryption is blockwise.

2. A message block is fi rst gone through an initial permutation IP, then divided into two parts $L_0||R_0$, where L_0 is the left 32 bits.

3. Round i has input $L_{i-1}||R_{i-1}$ and output $L_i||R_i$, where

$$L_i = R_{i-1},\ R_i = L_{i-1} \oplus f(R_{i-1}, k_i)$$

and k_i is the subkey for the ith round, where $1 \leq i \leq 16$.

4. After Round 16, L_{16} and R_{16} are swapped, so that the decryption algorithm has the same structure as the encryption algorithm.

5. Finally, the block is gone through the inverse permutation IP^{-1} and then output.

The DES Building Blocks

The following will be described in the next lecture.

1. The IP is a permutation on $\{1, 2, \cdots, 64\}$.

2. $f(x, k)$ is a function from $\{0, 1\}^{32} \times \{0, 1\}^{48}$ to $\{0, 1\}^{32}$.

3. The key scheduling algorithm for producing the 16 round subkeys k_i.

Decryption of the DES

Question: How to decrypt?

Observation: In encryption, we have

$$L_i = R_{i-1},\ R_i = L_{i-1} \oplus f(R_{i-1}, k_i)$$

and k_i is the subkey for the ith round. Hence

$$R_{i-1} = L_i, \quad L_{i-1} = R_i \oplus f(L_i, k_i) \tag{1}$$

for each i.

TO BE CONTINUED

Decryption of the DES ctd.

1st observation: Due to the swap after the 16th round encryption, the output of encryption is $IP^{-1}(R_{16}||L_{16})$.

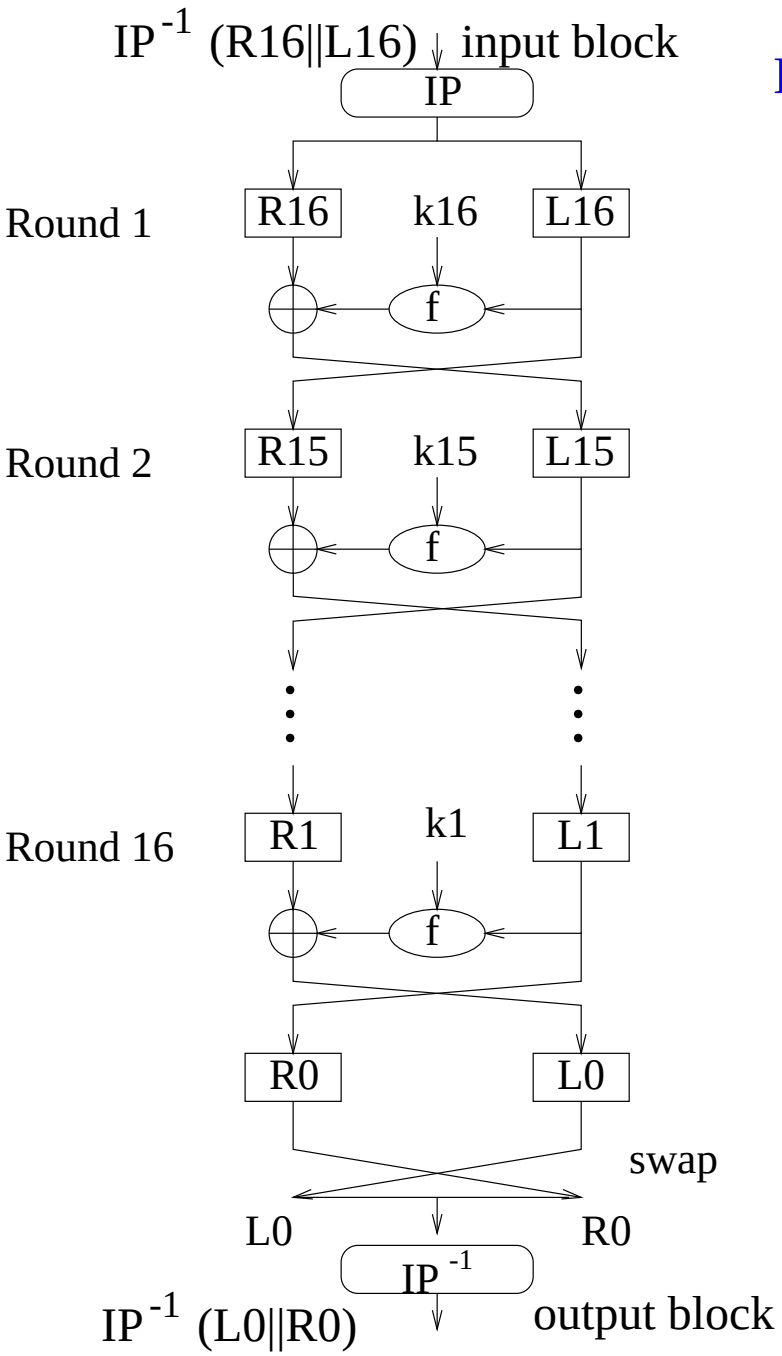
2nd observation: Equation (1) as follows:

$$
\begin{array}{lcllcll}
R_{15} & = & L_{16}, & L_{15} & = & R_{16} \oplus & f(L_{16}, k_{16}) \\
R_{14} & = & L_{15}, & L_{14} & = & R_{15} \oplus & f(L_{15}, k_{15}) \\
R_{13} & = & L_{14}, & L_{13} & = & R_{14} \oplus & f(L_{14}, k_{14}) \\
 & \vdots & \quad \vdots & & \vdots & & \\
R_{2} & = & L_{3}, & L_{2} & = & R_{3} \oplus & f(L_{3}, k_{3}) \\
R_{1} & = & L_{2}, & L_{1} & = & R_{2} \oplus & f(L_{2}, k_{2})
\end{array}
$$

3rd observation: If we give $IP^{-1}(R_{16}||L_{16})$ as the input for the same algorithm with the round subkeys $(k_{16}, k_{15}, ..., k_1)$, then the output is $IP^{-1}(L_0||R_0)$, the original message block.

Decryption algorithm: Decryption is performed using the same algorithm, except that k_{16} is used the first round, k_{15} in the second, and so on, with k_1 used in the 16th round.

Decryption of the DES ctd.



Decryption

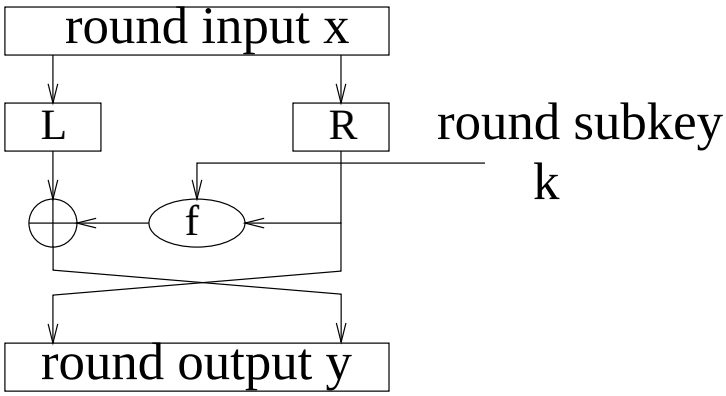
Remark and Question on the DES

Remark: The encryption and decryption process work, INDEPENDENT of how $f(x, k)$ is designed! So different designs of the building block $f(x, k)$ give different block ciphers.

Question: Given the DES encryption and decryption structure described before, how would you design your own $f(x, k)$ so that your block cipher is both secure and fast?

An Iterative View at DES

The round function $F_k(x)$



An Iterative View at DES

Encryption:

$$c = IP^{-1}([swap[F_{k_{16}}(\cdots F_{k_2}(F_{k_1}(IP(m)))\cdots)]]).$$

Where m is a 64-bit input block and c is the output block.

Thus DES encryption is essentially iterating the round function 16 times plus two permutations and a swamp of the first and second half of a block.

Remark: If each round function is viewed as an encryption algorithm, then DES is a composition of 16 small ciphers. Thus it is a product cipher.

Design Considerations of the DES

- It should be fast in both hardware and software.

- The keysize should be large enough to prevent the exhaustive search. In 1976, the keysize 56 was regarded as large enough for the next 20 years.

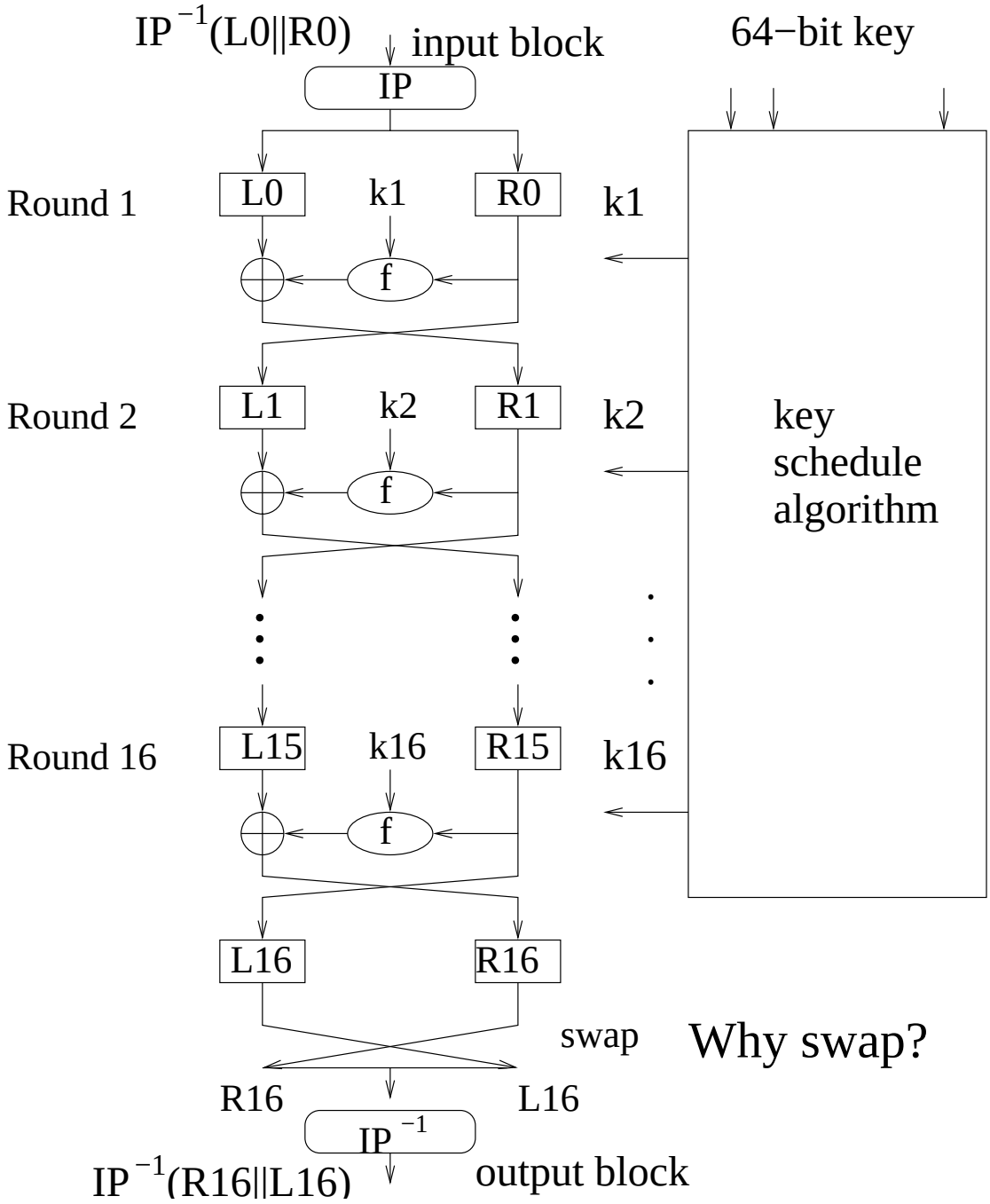
- Security of DES depends on the design of round function $f(x, k)$ and the key scheduling algorithm for producing the round subkeys. We shall look at them in the next lecture.

Part 2: The Building Blocks in Detail

Objectives of Part 2

- To describe the building blocks of DES in details.
- To give information about the security of DES.
- To describe some variants of DES.

The DES Encryption Process



The Initial Permutation: 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

Input and output of the permutation layer:

$$(x_1, x_2, \cdots, x_{64}) \mapsto (x_{IP(1)}, x_{IP(2)}, \cdots, x_{IP(64)})$$

The Final Permutation: 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

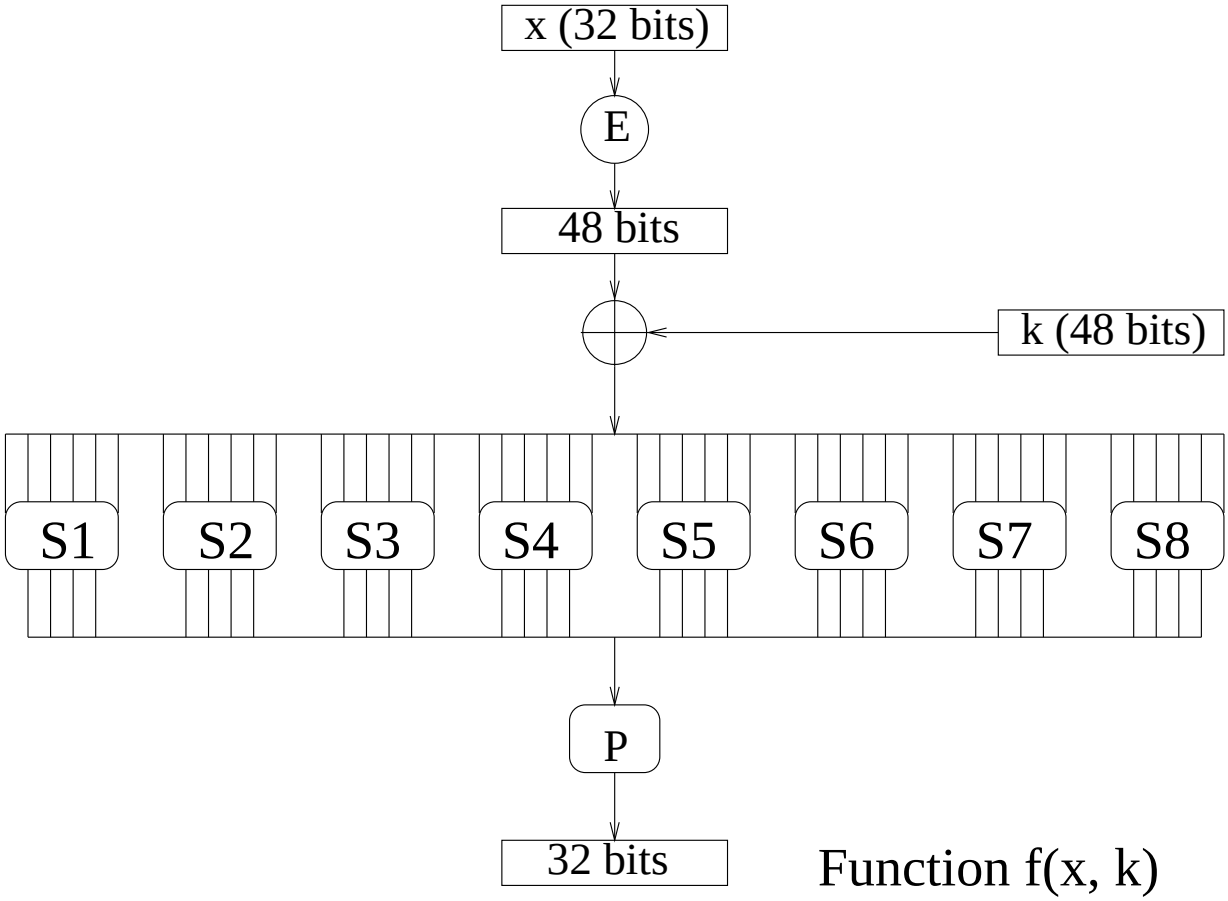
Input and output of the inverse permutation layer:

$$(x_1, \cdots, x_{64}) \mapsto (x_{IP^{-1}(1)}, \cdots, x_{IP^{-1}(64)})$$

The Function $f(x, k)$

Remark: E, P and S_i will be described later.

Remark: f should mix x and k "properly".



Function f(x, k)

The Function $f(x, k)$

The bit-selection table E:

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

Input and output of the bit-selection layer:

$$(x_1, x_2, \cdots, x_{32}) \mapsto (x_{E(1)}, x_{E(2)}, \cdots, x_{E(48)})$$

The Function $f(x, k)$ – Permutation P

The permutation P:

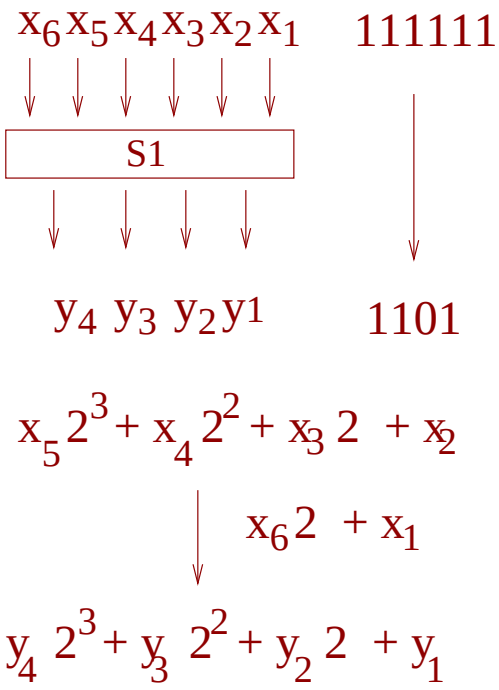
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Input and output of the permutation layer P:

$$(x_1, x_2, \cdots, x_{32}) \mapsto (x_{P(1)}, x_{P(2)}, \cdots, x_{P(32)})$$

The Function $f(x, k)$ – S-boxes

S_1: The first and last bits of the 6-bit input determine which column permutation is used. It provides nonlinearity (confusion).



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

Remark: S_2, ..., S_8 are similar and omitted (see other references for detail).

Parity Check Bits for Error Detection

Definition: For any binary string $a_1 a_2 \cdots a_n$, append another bit $a_{n+1} = a_1 \oplus a_2 \oplus \cdots \oplus a_n$, obtaining $a_1 a_2 \cdots a_n a_{n+1}$. This new sequence can detect one error.

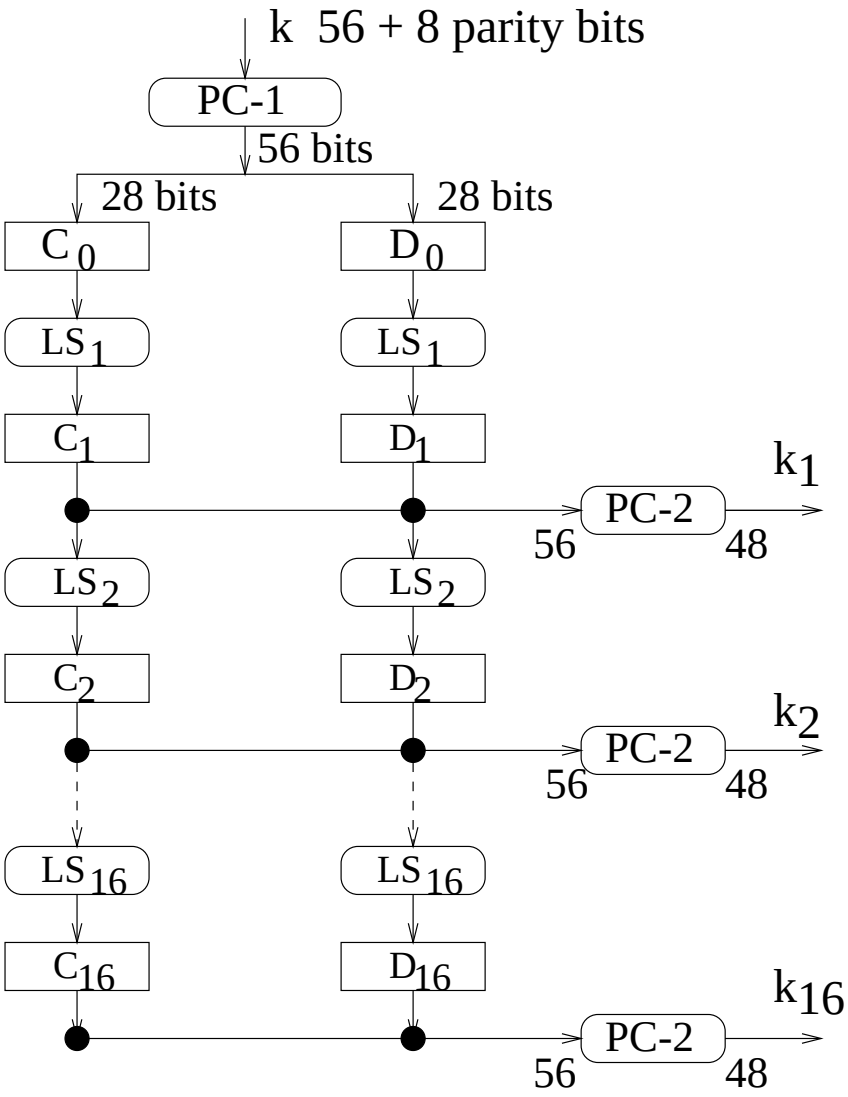
Adding 8 parity check bits in DES key:

$$8 \left\{ \begin{array}{cccccccc} * & * & * & * & * & * & * & p_1 \\ * & * & * & * & * & * & * & p_2 \\ * & * & * & * & * & * & * & p_3 \\ * & * & * & * & * & * & * & p_4 \\ * & * & * & * & * & * & * & p_5 \\ * & * & * & * & * & * & * & p_6 \\ * & * & * & * & * & * & * & p_7 \\ * & * & * & * & * & * & * & p_8 \end{array} \right.$$

Remark: Each p_i in position $8i$ is the parity check bit of the previous 7 bits.

The Key Schedule Algorithm

Input: 56-bit key plus 8 parity bits in positions 8, 16, ..., 64.



Comment: Each k_i should take any string of $\{0,1\}^{48}$ equally likely. Each key bit should be involved in at least one k_i. Clearly, some k_i and k_j cannot be independent.

The Key Schedule Algorithm

PC-1: The permutation PC-1 (permuted choice 1) discards the parity bits and transposes the remaining 56 bits as below:

Key permutation PC-1:

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

Without positions 8, 16, 24, 32, 40, 48, 56, 64 marked with “F”.

Remark: PC-1 is a permutation of

$$\{1, 2, ..., 64\} \setminus \{8, 16, 24, 32, 40, 48, 56, 64\}.$$

The Key Schedule Algorithm

LS_i: Each LS_i is a circular left shift of some positions. The number of shifted positions is given below.

iteration i	number of left shift
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 Key Schedule Algorithm

PC-2: It (permuted choice 2) selects 48 bits from the 56 bit input.

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

Input and output of the layer PC-2:

$$(x_1, x_2, \cdots, x_{56}) \mapsto (x_{PC-2(1)}, x_{PC-2(2)}, \cdots, x_{PC-2(48)}).$$

DES Design Criteria

Question: What are the design criteria for the building blocks of the DES algorithm?

Answer: This is out of the scope of this course. Interested parties are referred to the following references:

- B. Schneier, Applied Cryptography, 2nd Edition, John Wiley & Sons, 1996, pp. 293–294.

- D. Coppersmith, The Data Encryption Standard (DES) and Its Strength Against Attacks, IBM Journal of Research and Development, May 1994.

Security of DES

Question: Is DES really secure?

Answer: It is not regarded as secure only because its key length is too short, in view of today's hardware technology. So DES has been replaced by the AES – Advanced Encryption Standard (Rijndael).

In the public literature there is no practical attack on DES that is based on the structure of DES. But it possible that some secret organization has a practical attack.

- D. Coppersmith, The Data Encryption Standard (DES) and Its Strength Against Attacks, IBM Journal of Research and Development, May 1994.

DES Variants

Triple DES:

Encryption: $c = DES_{k_1}(DES_{k_2}(DES_{k_3}(m)))$.

Decryption: $m = DES^{-1}_{k_3}(DES^{-1}_{k_2}(DES^{-1}_{k_1}(c)))$.

Key length $= 3 \times 56 = 168$. If $k_1 = k_3 \neq k_2$, it is called TRIPLE DES WITH TWO KEYS.

Other Variants: DES with Independent Subkeys, and CRYPT(3) (used in Unix system), etc.

Reference: B. Schneier, Applied Cryptography, 2nd Edition, John Wiley & Sons, 1996, pp. 294–300.

Part 3: Looking further into DES

Objective of this Part

The Data Encryption Standard is described in the previous two lectures without giving details of the design criteria of the building blocks. The objectives of this lecture is:

- To show some of the design criteria of the building blocks published in the literature.

- To give some further explanations of the DES structure.

Linear Functions

Notation: Let $\mathbf{F}_2$ denote the set $\{0, 1\}$ and let

$$\mathbf{F}_2^n = \{(x_1, x_2, \cdots, x_n) | x_i \in \mathbf{F}_2\}.$$

We always associate $\mathbf{F}_2^n$ with the bitwise exclusive-or operation, also denoted $+$.

Linear functions: Let f be a function from $\mathbf{F}_2^n$ to $\mathbf{F}_2^m$, where n and m are positive integers. f is called **linear** if

$$f(x + y) = f(x) + f(y)$$

for all $x, y \in \mathbf{F}_2^n$.

Example: Let $f(x) = x_1 + x_2 + \cdots + x_n$, where

$$x = (x_1, \cdots, x_n) \in \mathbf{F}_2^n.$$

Then f is a linear function from $\mathbf{F}_2^n$ to $\mathbf{F}_2$. Note that $+$ denotes the modulo-2 addition.

Linear Functions

Linear permutations: Let P be a permutation of the set $\{1, \cdots, n\}$. Define a function L_P from $\mathbf{F}_2^n$ to itself by

$$L_P((x_1, x_2, \cdots, x_n)) = (x_{P(1)}, x_{P(2)}, \cdots, x_{P(n)})$$

for any $x = (x_1, x_2, \cdots, x_n) \in \mathbf{F}_n$.

Lemma: L_P is linear with respect to the bitwise exclusive-or.

Proof: Trivial.

Conclusion: All the permutation layers in DES are linear with respect to the bitwise exclusive-or, specifically, IP, IP^{-1}, P in f, and the PC-1 in key scheduling.

Linear Functions

Linear function for data expansion/compression: Let E be a function from $\{1, 2, \cdots, m\}$ to $\{1, 2, \cdots, n\}$. Define a function L_E from $\mathbf{F}_2^n$ to $\mathbf{F}_2^m$ by

$$L_E((x_1, x_2, \cdots, x_n)) = (x_{E(1)}, x_{E(2)}, \cdots, x_{E(m)})$$

for any $x = (x_1, x_2, \cdots, x_n) \in \mathbf{F}_n$.

Lemma: L_E is linear with respect to the bitwise exclusive-or.

Proof: Trivial.

Comments: It is for data expansion if $n < m$, and data compression if $n > m$.

Conclusion: The bit-selection layer E in f, and the PC-2 in the key scheduling are linear.

Linear Functions

Linear function by circular shift: Let i be any positive integer. Define a function LS_i from $\mathbf{F}_2^n$ to $\mathbf{F}_2^n$ by

$$\begin{aligned} & LS_i((x_0, x_1, \cdots, x_{n-1})) \\ = \; & (x_{(0-i) \bmod n}, x_{(1-i) \bmod n}, \cdots, x_{(n-1-i) \bmod n}) \end{aligned}$$

for any $x = (x_0, x_1, \cdots, x_{n-1}) \in \mathbf{F}_n$.

Lemma: LS_i is linear with respect to the bitwise exclusive-or.

Proof: Trivial.

Comment: If n is even and $i = n/2$, then LS_i just swaps the first half and the second half of x.

Conclusion: The LS_i in key scheduling and the swap in DES structure are linear operations.

Linear Functions

Bilinear functions: Define a function B from $\mathbf{F}_2^n \times \mathbf{F}_2^n$ to $\mathbf{F}_2^n$ by

$$B(x, y) = x + y$$

for any $x, y \in \mathbf{F}_n$.

Definition: B is **bilinear**, as it is linear with respect to one variable when the other one is fixed.

Nonlinearity of S-Boxes

The S-box $S1$: Note that

$$S1(111111) = 1101, \quad S1(000000) = 1110.$$

However,

$$\begin{aligned} & S1(111111 + 000000) = 1101 \\ \neq \; & S1(111111) + S1(000000) = 0011. \end{aligned}$$

So $S1$ is not linear with respect to the bitwise exclusive-or operation.

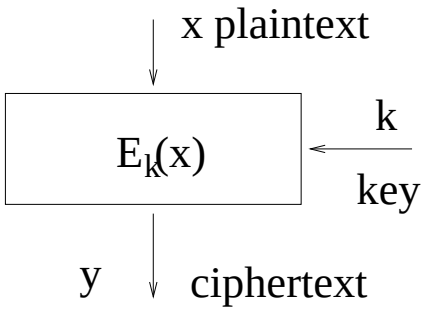
Remark: Other S-boxes are also not linear.

Conclusion: The S-boxes are the only nonlinear parts in DES!

Problem: Show that DES can be easily broken if the S-boxes are linear with respect to the bitwise exclusive-or operation (this is a large project).

Diffusion Requirement

Diffusion: Each plaintext block bit or key bit affects many bits of the ciphertext block.



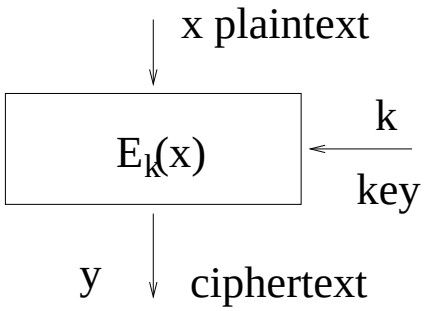
Example: Suppose that x, y and k all have 8 bits. If

$$
\begin{aligned}
y_1 &= f_1(x_1, x_2, k_1, k_2) \\
y_2 &= f_2(x_2, x_3, k_2, k_3) \\
y_3 &= f_3(x_3, x_4, k_3, k_4) \\
y_4 &= f_4(x_4, x_5, k_4, k_5) \\
y_5 &= f_5(x_5, x_6, k_5, k_6) \\
y_6 &= f_6(x_6, x_7, k_6, k_7) \\
y_7 &= f_7(x_7, x_8, k_7, k_8) \\
y_8 &= f_8(x_8, x_1, k_8, k_1)
\end{aligned}
$$

where the f_i are some functions, then it has very bad diffusion, because each plaintext bit or key bit affects only two bits in the output block y.

Diffusion Requirement

Diffusion: Each plaintext block bit or key bit affects many bits of the ciphertext block.



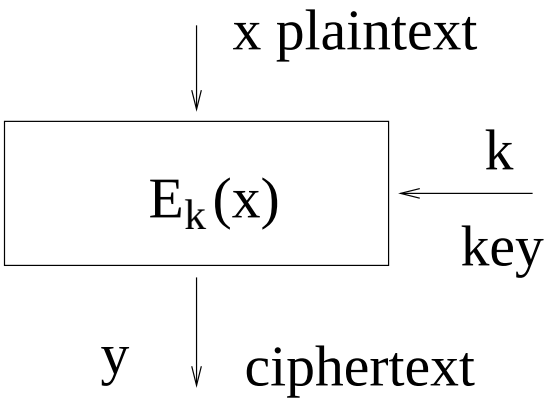
Example: Suppose that x, y and k all have 8 bits. If

$$
\begin{aligned}
y_1 &= x_1 + x_2 + x_3 + x_4 + k_1 + k_2 + k_3 + k_4 \\
y_2 &= x_2 + x_3 + x_4 + x_5 + k_2 + k_3 + k_4 + k_5 \\
y_3 &= x_3 + x_4 + x_5 + x_6 + k_3 + k_4 + k_5 + k_6 \\
y_4 &= x_4 + x_5 + x_6 + x_7 + k_4 + k_5 + k_6 + k_7 \\
y_5 &= x_5 + x_6 + x_7 + x_8 + k_5 + k_6 + k_7 + k_8 \\
y_6 &= x_6 + x_7 + x_8 + x_1 + k_6 + k_7 + k_8 + k_1 \\
y_7 &= x_7 + x_8 + x_1 + x_2 + k_7 + k_8 + k_1 + k_2 \\
y_8 &= x_8 + x_1 + x_2 + x_3 + k_8 + k_1 + k_2 + k_3
\end{aligned}
$$

then it has very good diffusion, because each plaintext bit or key bit affects half of the bits in the output block y.

Avalanche Effect

Avalanche effect requirement for encryption algorithm: A small change in either the plaintext or the key should produce a significant change in the ciphertext.



Remark: The avalanche effect is in fact a measure of diffusion.

Remark: Linear functions are usually for diffusion.

Avalanche Effect in DES

Round	No. of bits that differ
0	1
1	6
2	21
3	35
4	39
5	34
6	32
7	31
8	29
9	42
10	44
11	32
12	30
13	30
14	26
15	29
16	34

Change in plaintext: With two plaintext blocks differing in one position and one specific key. Already good after round 3.

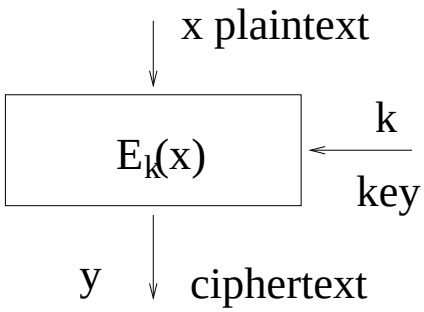
Avalanche Effect in DES

Round	No. of bits that differ
0	1
1	2
2	14
3	28
4	32
5	30
6	32
7	35
8	34
9	40
10	38
11	31
12	33
13	28
14	26
15	34
16	35

Change in key: With two keys differing in one position and one specific plaintext block. Already good after round 3.

Confusion Requirement

Confusion: Each bit of the ciphertext block has highly nonlinear relations with the plaintext block bits and the key bits.



Example: Suppose that x, y and k all have 8 bits. If

$$
\begin{aligned}
y_1 &= x_1 + x_2 + x_3 + x_4 + k_1 + k_2 + k_3 + k_4 \\
y_2 &= x_2 + x_3 + x_4 + x_5 + k_2 + k_3 + k_4 + k_5 \\
y_3 &= x_3 + x_4 + x_5 + x_6 + k_3 + k_4 + k_5 + k_6 \\
y_4 &= x_4 + x_5 + x_6 + x_7 + k_4 + k_5 + k_6 + k_7 \\
y_5 &= x_5 + x_6 + x_7 + x_8 + k_5 + k_6 + k_7 + k_8 \\
y_6 &= x_6 + x_7 + x_8 + x_1 + k_6 + k_7 + k_8 + k_1 \\
y_7 &= x_7 + x_8 + x_1 + x_2 + k_7 + k_8 + k_1 + k_2 \\
y_8 &= x_8 + x_1 + x_2 + x_3 + k_8 + k_1 + k_2 + k_3
\end{aligned}
$$

then it has bad confusion, as they are linear relations.

Remark: Nonlinear functions are responsible for confusion. In DES the eight S-boxes are for confusion.

Nonlinearity Measures

Measure: How far the function is from all linear functions.

Example: The function

$$f(x) = x_1 + x_2 + x_3 + x_4 + x_1x_2x_3x_4$$

has very bad nonlinearity, as it is very close to the linear function $l(x) = x_1 + x_2 + x_3 + x_4$. In other words, $f(x) = g(x)$ for all x except $x = (1111)$.

Remark: The discussion of nonlinearity is out of the scope of this course.

DES S-boxes Design Criteria

1. For any S-box S_i

   $$\Pr[S_i(x) = a_1x_1 + a_2x_2 + \cdots + a_6x_6] \approx \frac{1}{2}$$

   for any constants $a_i \in \{0, 1\}$.

2. For each fixed (x_6, x_1), $S_i(x_1, x_2, \cdots, x_5, x_6)$ should be a permutation of $\mathbf{F}_2^4$.

3. If two inputs to an S-box differ in exactly one bit, the outputs must differ in at least two bits.

4. If two inputs to an S-box differ in the two middle bits exactly, the outputs must differ in at least two bits.

DES S-boxes Design Criteria – Continued

5. If two inputs to an S-box differ in their first two bits and are identical in their last two bits, the two outputs must not be the same.

6. For any nonzero 6-bit difference between inputs, no more than 8 of the 32 pairs of inputs exhibiting that difference may result in the same output difference.

7. Similar to the previous one.

Permutation P Design Criteria in DES

1. The four output bits from each S-box at round i are distributed so that two of them affect (provide input for) "middle bits" of round $(i + 1)$ and the other two affect end bits. The two middle bits of input to an S-box are not shared with adjacent S-boxes. The end bits are the two left-hand bits and the two right-hand bits, which are shared with adjacent S-boxes.

To be continued

Permutation P Design Criteria in DES – Continued

2. The four output bits from each S-box affect six different S-boxes on the next round, and no two affect the same S-box.

3. For two S-boxes j and k, if an output bit from S_j affects a middle bit of S_k on the next round, then an output bit from S_k cannot affect a middle bit of S_j. This implies that for $j = k$, an output bit from S_j must not affect a middle bit of S_j.

Remark: These criteria are intended to increase the diffusion of the DES algorithm.

Key Schedule Algorithm in DES

1. Each round subkey k_i should take on each element of $\mathbf{F}_2^{48}$ equally likely.

2. Each key bit should affect at least one k_i.

3. k_i and k_{i+1} should not involve many common key bits.

Note that all the functions in the key scheduling algorithm are linear. This makes it easy to satisfy the first requirement.

Remark: These are our observations, and are not criteria published by the original designers.

The Number of Rounds in DES

Security: f, key scheduling algorithm, and the number of rounds.

1. Trade-off between security and performance.

2. 16 rounds are to thwart the “differential cryptanalysis” (out of the scope of this course), and other possible attacks.