

Information Security

Project 1

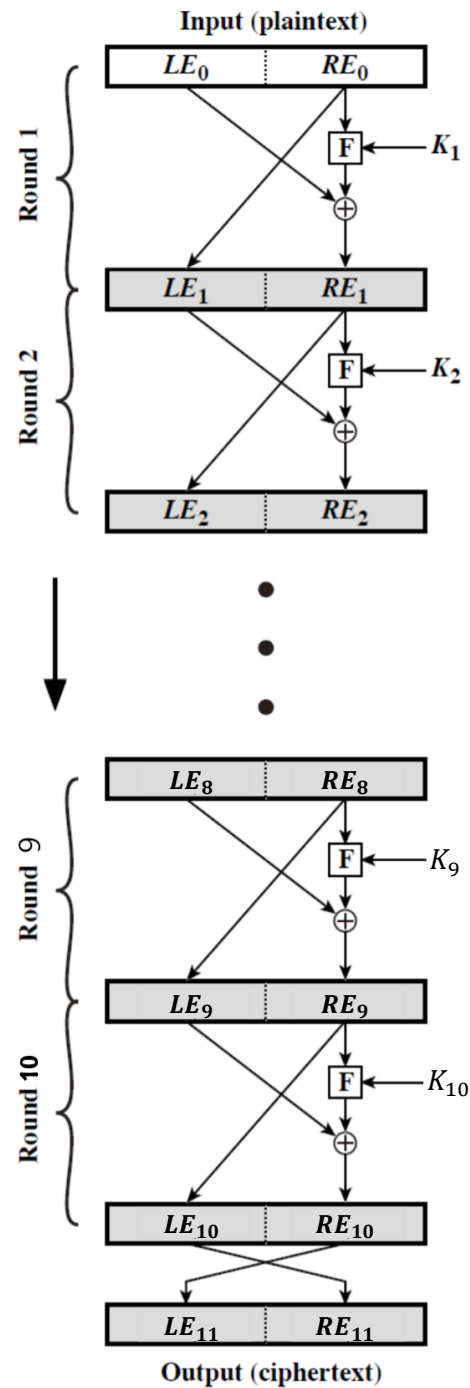
Symmetric Cipher

Prof. Junbeom Hur
Department of Computer Science and Engineering
Korea University

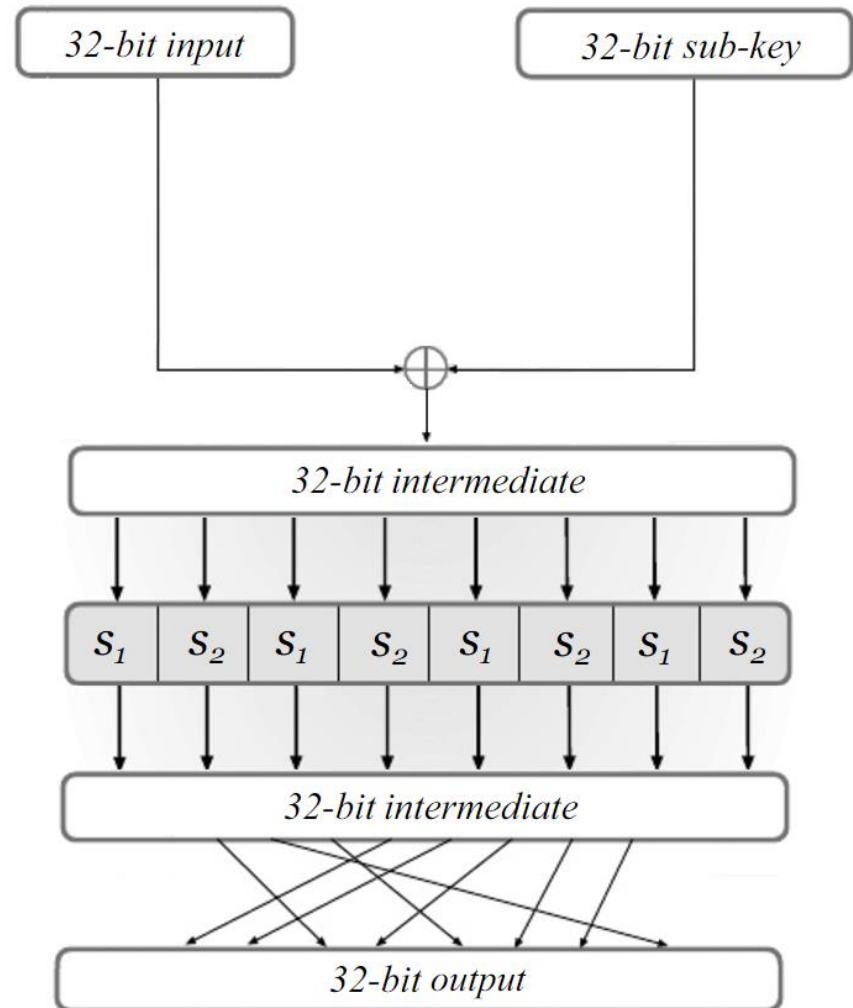
Project 1

- Consider the following encryption algorithm
 - Key size: 32 bits
 - Block size: 64 bits
 - Structure: Feistel-Network with 10 rounds
 - S-box and Permutation structure are given

Feistel Cipher Structure



F function Structure



Project 1

- Key Schedule
 - Create 10 subkeys using one 32bit input
- Algorithm
 - Divide the input into 16 bits L and R
 - The i-th L_i and R_i are shift-left operations by i-bit, respectively
 - The i-th Subkey is concatenated with L_i and R_i

Key schedule example

Input - 0101010101010101 1011011101111011

L - 0101010101010101

R- 1011011101111011

L_1 - 1010101010101010

R_1 - 0110111011110111

L_2 - 0101010101010101

R_2 - 1101110111101110

L_3 - 1010101010101010

R_3 - 1011101111011101

L_4 - 0101010101010101

R_4 - 0111011110111011

•
•
•

•
•
•

Project 1

- S-box
 - Two S-boxes with 4-bit input and 4-bit output are used repeatedly
- Algorithm
 - S_1

Input	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Output	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

– S_2

Input	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Output	5	6	C	F	8	A	0	4	B	3	7	D	E	1	2	9

Project 1

- Permutation
 - A fixed Permutation table is given
 - Ex) The 0th bit goes to the 29th bit
- Algorithm

Before	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
After	29	1	17	8	30	22	28	6	18	4	12	19	21	26	2	20

Before	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
After	31	10	9	25	13	0	23	15	3	27	5	11	7	14	24	16

Permutation example

- Input
 - 0xE4 0xE8 0xEF 0x2E
 - 1110 0100 1110 1000 1110 1111 0010 1110
- Output
 - 0xCD 0x6F 0x67 0x85
 - 1100 1101 0110 1111 0110 0111 1000 0101

Project 1

1. Implement the encryption algorithm
 - You can use any language (e.g., C, Java, Python, ...)
 - E.g., Sample info to check the correctness
 - key: 0x04 0x34 0xEF 0x71
 - plaintext: 0x10 0x24 0xAA 0x9F 0x47 0x3C 0x58 0xC1
 - ciphertext: 0x4F 0xC8 0x37 0x60 0xC7 0x8F 0x6E 0xF0

Project 1

1. Implement the encryption algorithm

- E.g., Sample info to check the correctness
 - Key: 0x12 0x34 0x56 0x78
 - plaintext: 0x77 0x02 0x45 0xFF 0xBA 0xD4 0x17 0x3E
 - ciphertext: 0x41 0x79 0x18 0x3B 0x2D 0xAC 0x7C 0x0B
- key: 0x98 0x76 0x54 0x32
- plaintext: 0x77 0x02 0x45 0xFF 0xBA 0xD4 0x17 0x3A
- ciphertext: 0xC9 0x00 0x0E 0x27 0x3C 0x7D 0x0B 0x57

Project 1

1. Implement the encryption algorithm
 - E.g., Sample info to check the correctness

```
original key: 12 34 56 78
original plaintext: 77 02 45 FF BA D4 17 3E
original Left: 77 02 45 FF
original Right: BA D4 17 3E
1 round Left: BA D4 17 3E
1 round Right: DF 37 21 42
2 round Left: DF 37 21 42
2 round Right: BF D3 D4 8B
3 round Left: BF D3 D4 8B
3 round Right: 3B F7 8D AF
4 round Left: 3B F7 8D AF
4 round Right: 65 C0 75 11
5 round Left: 65 C0 75 11
5 round Right: F8 5C 0A 90
6 round Left: F8 5C 0A 90
6 round Right: 48 9C 49 9E
7 round Left: 48 9C 49 9E
7 round Right: 89 D0 6D 07
8 round Left: 89 D0 6D 07
8 round Right: 16 AA 5C 2A
9 round Left: 16 AA 5C 2A
9 round Right: 2D AC 7C 0B
10 round Left: 2D AC 7C 0B
10 round Right: 41 79 18 3B
ciphertext: 41 79 18 3B 2D AC 7C 0B
```

Project 1

2. Find the key K

- Known plaintext attack (Hint : The key starts with 0xF2)
- Sample:
 - plaintext: 0x40 0xFF 0x24 0x33 0x09 0x47 0xF6 0x10
 - ciphertext: 0xEC 0x2D 0xE1 0x30 0x5B 0x5F 0x5B 0x02
- plaintext: 0x21 0x74 0xC5 0x01 0xAC 0x12 0xF9 0xD1
- ciphertext: 0xDF 0x9F 0xCC 0x3F 0xFE 0x09 0x80 0x9D

Capture the result screen!!

Project 1

- Due date
 - 2017. Oct. 31, 23:59
 - Upload your source programs and results screen into the Blackboard
 - **Plagiarism will be “F”**
- If you have any question, send an email to T.A
 - Hyunsoo Kwon (khs910504@gmail.com)
 - Youngki Hong (gee308@naver.com)