

Chapter 3: Block Ciphers and Advanced Encryption Standard

COMP 7120-8120

Lih-Yuan Deng

lihdeng@memphis.edu

Outline

- Introduction
- SPN: Substitution-Permutation Network
- Linear Cryptanalysis
- Differential Cryptanalysis
- DES: Data Encryption Standard
- AES: Advanced Encryption Standard
- Modes of Operations

Introduction

- modern block ciphers: a sequence of product ciphers with substitution and permutation (transposition) cipher.
- N_r = number of rounds in the iterated cipher.
- $(K^1, K^2, \dots, K^{N_r})$: key schedule, K^r is a subkey (obtained from key K with known algorithm) for r -th round.



General encryption steps

- x : plaintext, K : key, K^r : subkey, N_r : number of rounds (iterations).
- $g(w,k)$ = encryption function with key k .
- Algorithm:
 1. Initially, set $w^0 \leftarrow x$.
 2. for r from 1 to N_r do
 - $w^r \leftarrow g(w^{r-1}, K^r)$
 3. $y \leftarrow w^{N_r}$.



General decryption steps

- y : ciphertext, K : key, K^r : subkey, N_r : number of rounds (iterations).
- $g^{-1}(w,k)$ = decryption function with key k .
- Algorithm: (reverse of encryption steps)
 1. Initially, set $w^{N_r} \leftarrow y$.
 2. for r from N_r down to 1 do
 - $w^{r-1} \leftarrow g^{-1}(w^r, K^r)$
 3. $x \leftarrow w^0$.



SPN: Substitution-Permutation Network

- Consider $P=C=\{0,1\}^{km}$ m words each of length l .
- Two components in SPN:
 - $\pi_S : \{0,1\}^l \rightarrow \{0,1\}^l$, substitution of word (of length l) to another word. [S-box]
 - $\pi_P : \{0,1\}^{km} \rightarrow \{0,1\}^{km}$, a transposition (permutation) of $l \times m$ bits.



Notations

- binary $x = (x_1, x_2, \dots, x_{km}) \in \{0,1\}^{km}$
- $x_{<l>}$ = l -th substring (word) of length l , $i=1, \dots, m$.
- $x = x_{<1>} || x_{<2>} \dots || x_{<m>}$
 - where $||$ denotes concatenation.
- π_S and π_P are encryption functions for **known substitution (of word) [S-box]** and **permutation** of (position in) length $l \times m$ bits.



More notations

- r is the running index, $r=1,2,\dots,Nr$.
- At r -th round, (initially, $w^0 \leftarrow x$)
 - $u^r \leftarrow w^{r-1} \oplus K^r$
 - K^r are only secret keys. (why this ?)
 - u^r_i is i -th word of u^r , as an **input** to the S-box, and v^r_i is its output. $v^r_i = \pi_S(u^r_i)$, $i=1,2,\dots,m$.
 - for simplicity, we write $v^r = S(u^r)$
 - $w^r = \pi_P(v^r_1, v^r_2, \dots, v^r_m) = P(v^r)$
 - last round: no permutation (or $P=I$) why ?



Algorithm 3.1 (page 76)

- K^r : $(Nr+1)$ subkeys, Nr : # of rounds
- **Algorithm:**
 1. Initially, set $w^0 \leftarrow x$.
 2. for r from 1 to Nr do
 - $u^r \leftarrow w^{r-1} \oplus K^r$
 - $v^r \leftarrow S(u^r)$, (substitution in S-box)
 - $w^r \leftarrow P(v^r)$, (permutation, $P=I$ when $r=Nr$)
 3. $y \leftarrow w^{Nr} \oplus K^{Nr+1}$

Decrypting Algorithm 3.1

- K^r : $(Nr+1)$ subkeys, Nr : # of rounds
- **Decryption Algorithm:**
 1. Initially, set $w^{Nr} \leftarrow y \oplus K^{Nr+1}$
 2. for r from Nr down to 1 do
 - $v^r \leftarrow P^{-1}(w^r)$, (permutation, $P=I$ when $r=Nr$)
 - $u^r \leftarrow S^{-1}(v^r)$, (substitution in S-box)
 - $w^{r-1} \leftarrow u^r \oplus K^r$
 3. $x \leftarrow w^0$

Example 3.1 (SPN)

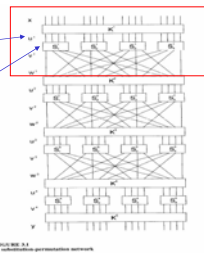
- $l = m = Nr = 4$.
- 32-bit keys with (simple) 5 sub-keys.
- $K = 0011 | 1010 | 1001 | 0100 | 1101 | 0110 | 0011 | 1111$

K1	0	0	1	1	1	0	1	0	1	0	0	1	0	1	0	0
K2	1	0	1	0	1	0	0	1	0	1	0	0	1	1	0	1
K3	1	0	0	1	0	1	0	0	1	1	0	1	0	1	1	0
K4	0	1	0	0	1	1	0	1	0	1	1	0	0	0	1	1
K5	1	1	0	1	0	1	1	0	0	0	1	1	1	1	1	1

simple key schedule used

Figure 3.1 (page 78)

- Initially, $w^0 \leftarrow x$.
- $u^1 = w^0 \oplus K^1$
- four (identical S-boxes) used: $S_1^1, S_2^1, S_3^1, S_4^1$.
- output of S-boxes (v^1) is transposed into w^1 [end of first round].





Example 3.1: tables of SPN

z	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
s(z)	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

z	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
s(z)	1110	0100	1101	0001	0010	1111	1011	1000	0011	1010	0110	1100	0101	1001	0000	0111

z	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
p(z)	1	5	9	13	2	6	10	14	3	7	11	15	4	8	12	16



Example 3.1. SPN Illustration

1st round

x	0	0	1	0	0	1	1	0	1	0	1	1	0	1	1	1
w0	0	0	1	0	0	1	1	0	1	0	1	1	0	1	1	1
K1	0	0	1	1	1	1	0	1	0	1	0	0	1	0	1	0
u ¹	0	0	0	1	1	1	0	0	0	1	0	0	1	1	1	1
w0+K1	0	0	0	1	1	1	0	0	0	1	0	0	1	1	1	1
v ¹	0	1	0	0	0	1	1	0	1	0	1	0	1	0	0	1
w1	0	1	0	0	0	1	1	0	1	0	1	0	1	0	0	1
w1+K1	0	0	1	0	1	1	1	0	0	0	0	0	1	1	1	1
v2	0	0	1	0	0	0	1	1	0	0	0	0	1	1	0	0
w2	0	0	1	0	0	0	1	1	0	0	0	0	1	1	0	0
w2+K2	0	0	1	0	0	0	1	1	0	0	0	0	1	1	0	0

z	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s(z)	1	4	4	3	1	2	5	1	8	3	0	6	2	5	9	0
z	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
p(z)	1	5	9	3	2	6	0	4	3	7	1	1	5	4	8	2



SPN: 2nd round

w1	0	0	1	0	1	1	1	0	0	0	0	0	0	1	1	1
w1+K1	1	0	1	0	1	0	0	1	0	1	0	0	1	1	0	1
K2	1	0	1	0	1	0	0	1	0	1	0	0	1	1	0	1
u ²	1	0	0	0	0	1	1	1	0	1	0	0	1	0	1	0
w1+K2	1	0	0	0	0	1	1	1	0	1	0	0	1	0	1	0
v ²	0	0	1	1	1	0	0	0	0	1	0	0	1	1	0	1
w2	0	0	1	1	1	0	0	0	0	1	0	0	1	1	0	1
w2+K2	0	0	1	0	0	0	1	1	0	1	1	1	0	0	0	0

z	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s(z)	1	4	4	3	1	2	5	1	8	3	0	6	2	5	9	0
z	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
p(z)	1	5	9	3	2	6	0	4	3	7	1	1	5	4	8	2



Summary

- What we have learned so far ?
 - SPN
 - S-box and encryption algorithm.
 - Example 3.1



Linear Cryptanalysis

- Assume that attacker has large number of plaintext-ciphertext pairs (**how large?**)
- For a S-box given, find (**how ?**) a random variable, say T, a **combination of inputs and outputs** of S-box, that contains most information. (small entropy, large bias, ...)
- Find (**how ?**) a series of S-boxes (**active S-boxes**) that can maintain this information in the SPN given.



Linear Cryptanalysis

- **Idea:** For active S-boxes (S_i) chosen,
 - choose (T_i) (a combination of inputs and outputs) that keeps information about (S_i).
 - write "combination" (or "piling-up") of (T_i) as a relation between inputs and outputs.
 - for **correct candidate subkey**, the "relation" will be maintained and display some non-uniform distribution in these T pairs.



Approximating S-box

- Not all the substitution tables (S-boxes) are created equal.
- For a given S-box, we can find the "worst" combination of X_i and Y_i that reveals the most non-uniform behavior.
- How to compare two S-boxes ?



Comparing S-boxes

X1	X2	X3	X4	Y1	Y2	Y3	Y4	T1	X1	X2	X3	X4	Y1	Y2	Y3	Y4	T1	
0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0
0	0	1	0	1	1	0	1	0	0	0	1	0	0	1	0	0	0	0
0	0	1	1	0	0	0	0	1	0	0	1	1	0	0	1	1	0	0
0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0
0	1	0	1	1	1	1	1	0	0	1	0	1	0	1	0	1	0	0
0	1	1	0	1	0	1	1	1	0	1	1	0	0	1	1	0	0	0
0	1	1	1	1	0	0	0	0	0	1	1	0	1	1	1	0	0	0
1	0	0	0	0	0	1	1	1	1	0	0	0	1	0	0	0	0	0
1	0	0	1	1	0	1	0	0	0	1	0	0	1	0	0	1	0	0
1	0	1	0	0	1	1	0	1	0	1	0	1	0	1	0	1	0	0
1	0	1	1	1	1	0	0	0	0	1	1	1	0	1	1	0	0	0
1	1	0	0	0	1	0	1	0	0	1	1	0	0	1	1	0	0	0
1	1	0	1	1	0	0	1	0	0	1	1	0	1	1	0	1	0	0
1	1	1	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0	0
1	1	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	0

T1=X1⊕X3 ⊕ X4 ⊕ Y2

0.25

T1= ⊕ All Xi's and Yi's

0.5



Linear Approximations of S-boxes

- Let $[X_1, X_2, \dots, X_m]$ be input of S-box.
- Let $[Y_1, Y_2, \dots, Y_n]$ be output of S-box.
- Consider a random variable which is a piling-up of input and output variables.

$$Z = (\oplus a_i X_i) \oplus (\oplus b_j Y_j)$$
- If Z has a large bias, then we can use it to mount a linear cryptanalytic attack.



Linear Approximation Table

$$Z_{(a,b)} = (\oplus a_i X_i) \oplus (\oplus b_i Y_i)$$

- Example 3.2. $m=n=4$,
- $a=(a_1, a_2, a_3, a_4)_2$, $b=(b_1, b_2, b_3, b_4)_2$.
 - $X_1 \oplus X_4 \oplus Y_2$ [$a=1001_2=9$, $b=0100_2=4$]
 - $X_3 \oplus X_4 \oplus Y_1 \oplus Y_4$ [show $a=3$, $b=9$]
- For each of $[a,b]$ ($16 \times 16=256$), we compute table of $N_L(a,b)$ for $Z_{(a,b)}$.
 - [bias $=\epsilon(a,b) = N_L(a,b)/16-0.5$]



Linear Approximation Table

$N_L(a,b)$	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	15	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
1	8															
2	8	8	6	6	8	8	6	6	8	8	10	10	8	8	2	10
3	8															
4	8	10	8	6	6	4	6	8	8	6	8	10	10	4	10	8
5	8															
6	8															
7	8															
8	8															
9	8															
A	8															
B	8	12	8	4	12	8	12	8	8	8	8	8	8	8	8	8
C	8															
D	8															
E	8															
F	8															
