# Network Security

# Homework 2

## CHANG LIU

## chang.liu@jhu.edu

# Feb. 29, 2012

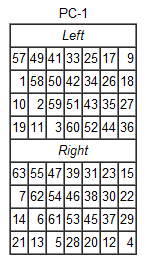
1. 1 5 9 13 17 21 25 29

**0000 0001 0010 0011 0100 0101 0110 0111**

33 37 41 45 49 53 57 61

**1000 1001 1010 1011 1100 1101 1110 1111**

1. **Derive the first round key K1.**



1111000

0110011

0010101

0100000

1010101

0110011

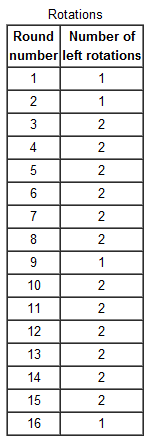
0011110

0000000

Thus,

L = 1111 0000 1100 1100 1010 1010 0000

R = 1010 1010 1100 1100 1111 0000 0000



* C1(L) = 1110 0001 1001 1001 0101 0100 0001
* D1(R) = 0101 0101 1001 1001 1110 0000 0001

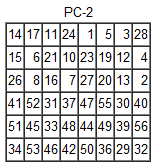
1 5 9 13 17 21 25

C1(L) = 1110 0001 1001 1001 0101 0100 0001

29 33 37 41 45 49 53

D1(R) = 0101 0101 1001 1001 1110 0000 0001

🡺



0000 1011

0000 0010

0110 0111

1001 1011

0100 1001

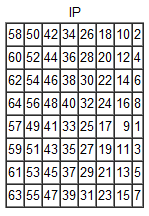
1010 0101

Thus,

K1 = 0000 1011 0000 0010 0110 0111

1001 1011 0100 1001 1010 0101

1. **Derive L0 and R0.**



1100 1100

0000 0000

1100 1100

1111 1111

1111 0000

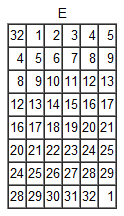
1010 1010

1111 0000

1010 1010

* L0 = 1100 1100 0000 0000 1100 1100 1111 1111
* R0 = 1111 0000 1010 1010 1111 0000 1010 1010

1. **Expand R0 to get E[R0], where E[.] is the expansion function of Table 3.2 of your textbook.**



1 5 9 13 17 21 25 29

R0 = 1111 0000 1010 1010 1111 0000 1010 1010

E[R0] = 011110 100001 010101 010101

011110 100001 010101 010101

1. **Calculate A = E[R0]⊕K1.**

A = E[R0]⊕K1 =

0111 1010 0001 0101 0101 0101

0111 1010 0001 0101 0101 0101

⊕

0000 1011 0000 0010 0110 0111

1001 1011 0100 1001 1010 0101

=

0111 0001 0001 0111 0011 0010

1110 0001 0101 1100 1111 0000

Thus

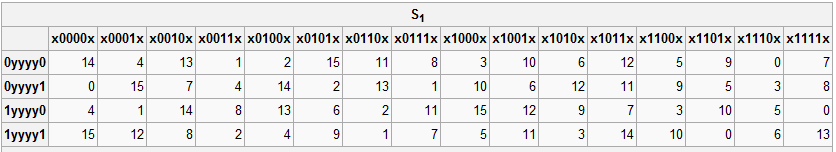
A = 011100 010001 011100 110010

111000 010101 110011 110000

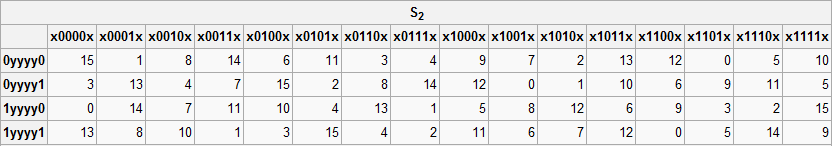
1. **Group the 48 bit results of (d) into sets of 6 bits and evaluate the corresponding S-box substitutions.**

A = 011100 010001 011100 110010

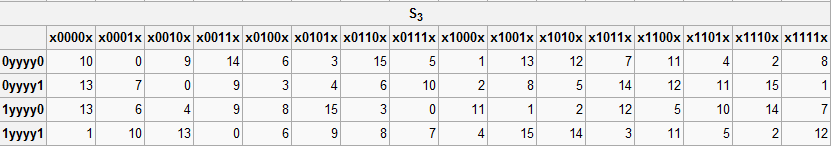
111000 010101 110011 110000



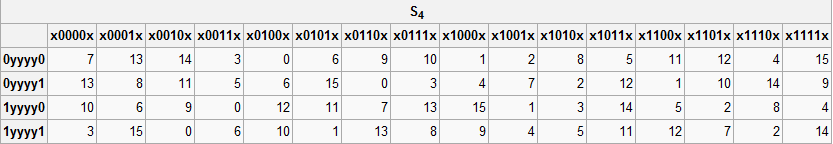
S1(011100) = 0000



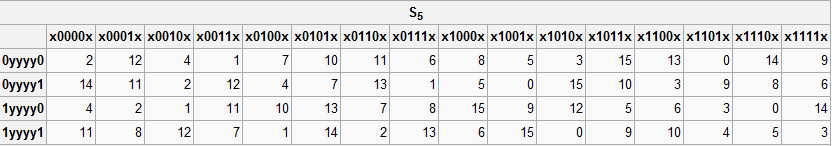
S2(010001) = 1100



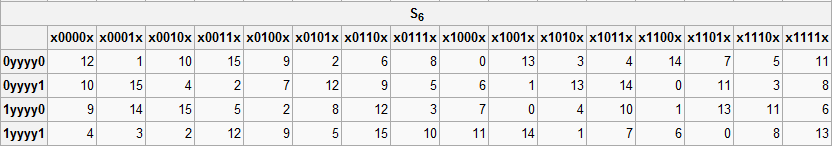
S3(011100) = 0010



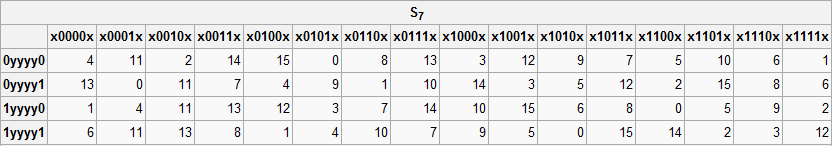
S4(110010) = 0001



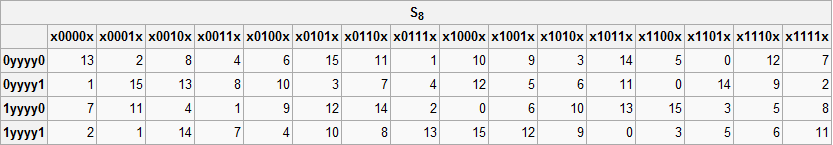
S5(111000) = 0110



S6(010101) = 1101



S7(110011) = 0101

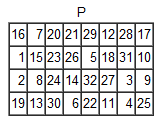


S8(110000) = 0000

1. **Concatenate the results of (e) to get a 32 bit result, B.**

B = 0000 1100 0010 0001 0110 1101 0101 0000

1. **Apply the permutation to get P(B).**



1 5 9 13 17 21 25 29

B = 0000 1100 0010 0001 0110 1101 0101 0000

* 10010010
* 00011100
* 00100000
* 10011100

P(B) = 1001 0010 0001 1100 0010 0000 1001 1100

1. **Calculate R1 = P(B)⊕L0.**

R1 = P(B)⊕L0

1001 0010 0001 1100 0010 0000 1001 1100

⊕

1100 1100 0000 0000 1100 1100 1111 1111

R1 = 0101 1110 0001 1100 1110 1100 0110 0011

1. **Write down the ciphertext.**

Cypertext = L1 + R1 = R0 + R1 =

1111 0000 1010 1010 1111 0000 1010 1010

0101 1110 0001 1100 1110 1100 0110 0011

1. **(a) Using RSA, choose p=3, and q=11, and encode the word “dog” by encrypting each letter separately. Apply the decryption algorithm to the encrypted version to recover the original plaintext message.**

N = pq = 33

(p-1)(q-1) = 2x10 = 20

e = 3

d: 

**Encrypting:**

d: 

o: 

g: 

The cypertext is “eim”

**Decryption:**

m1: 

m2: 

m3: 

Thus, we get the plaintext “dog”.

**(b)** **Repeat part (a) but now encrypt “dog” as one message.**

dog = 4,15,7 = 00100 01111 00111 = 10747 = 22(mod 33) = v

**Encryption:**

v: 

**Decryption:**

v: 

1. **Consider RSA with p=5 and q=11.**

**(a) what are n and z.**

n = pq = 55

z = (p-1)(q-1) = 4x10 = 40

**(b)** **Let e be 3, why is this an acceptable choice for e.**

e = 3 is acceptable because we need to choose an integer e such that 1 < e < z and e, z are coprime.

When e = 3, it fulfill all the above requirements, thus e = 3 is acceptable.

**(c) Find d such that de=1(mod z) and d<160.**

d: 

**(d) Encrypt the message m=8 using the key (n,e). Let c denote the corresponding ciphertext. Show all the work.**

**Encrypting:**

8: 

Thus the ciphertext is 17.