Modern Crypto

1 DES

Question 1. Next, use the DES simulator DEScalc.jar in the “SSN Lab 3” directory on the Desktop of the Virtual Box image (taken from http://lpb.canb.auug.org.au/adfa/src/DEScalc/). Step through the process of encrypting your name with the key 0x0101010101010101 and write the internal state of the device at the 8th round.

I- My name in hex is:

alachkar → 616c6163686b6172

II- Encrypted Value:

378a2608202c47b3

III- Encryption Internal Process:

```
setKey(0101010101010101)
encryptDES(616c6163686b6172)
  IP:   L0=ff80026d, R0=00ff32a8
  Rnd1  f(R0=00ff32a8, SK1=00 00 00 00 00 00 00 00 ) = 6fb2d280
  Rnd2  f(R1=9032d0ed, SK2=00 00 00 00 00 00 00 00 ) = 0883812d
  Rnd3  f(R2=087cb385, SK3=00 00 00 00 00 00 00 00 ) = 398edfef
  Rnd4  f(R3=a9bc0f02, SK4=00 00 00 00 00 00 00 00 ) = 9fa4b8a4
  Rnd5  f(R4=97d80b21, SK5=00 00 00 00 00 00 00 00 ) = cfc79ce9
  Rnd6  f(R5=667b93eb, SK6=00 00 00 00 00 00 00 00 ) = fb42476c
  Rnd7  f(R6=6c9a4c4d, SK7=00 00 00 00 00 00 00 00 ) = d074011a
  Rnd8  f(R7=b60f92f1, SK8=00 00 00 00 00 00 00 00 ) = a729cb52
  Rnd9  f(R8=cbb3871f, SK9=00 00 00 00 00 00 00 00 ) = 7bcf1f81
  Rnd10 f(R9=cdc08d70, SK10=00 00 00 00 00 00 00 00 ) = a4bf744b
  Rnd11 f(R10=6f0cf354, SK11=00 00 00 00 00 00 00 00 ) = ef88ac0a
  Rnd12 f(R11=2248217a, SK12=00 00 00 00 00 00 00 00 ) = 8e9ea406
  Rnd13 f(R12=e1925752, SK13=00 00 00 00 00 00 00 00 ) = 8f2a64f0
  Rnd14 f(R13=ad62458a, SK14=00 00 00 00 00 00 00 00 ) = 36c690ed
  Rnd15 f(R14=d754c7bf, SK15=00 00 00 00 00 00 00 00 ) = 2fd76f4d
  Rnd16 f(R15=82b52ac7, SK16=00 00 00 00 00 00 00 00 ) = 97d5a27e
  FP:   L=378a2608, R=202c47b3
 returns 378a2608202c47b3
```

III- Decryption Internal Process:

```
setKey(0101010101010101)
decryptDES(378a2608202c47b3)
  IP:   L0=408165c1, R0=82b52ac7
  Rnd1  f(R0=82b52ac7, SK16=00 00 00 00 00 00 00 00 ) = 97d5a27e
  Rnd2  f(R1=d754c7bf, SK15=00 00 00 00 00 00 00 00 ) = 2fd76f4d
  Rnd3  f(R2=ad62458a, SK14=00 00 00 00 00 00 00 00 ) = 36c690ed
  Rnd4  f(R3=e1925752, SK13=00 00 00 00 00 00 00 00 ) = 8f2a64f0
  Rnd5  f(R4=2248217a, SK12=00 00 00 00 00 00 00 00 ) = 8e9ea406
```

```
  Rnd6  f(R5=6f0cf354, SK11=00 00 00 00 00 00 00 00 ) = ef88ac0a
  Rnd7  f(R6=cdc08d70, SK10=00 00 00 00 00 00 00 00 ) = a4bf744b
  Rnd8  f(R7=cbb3871f, SK9=00 00 00 00 00 00 00 00 ) = 7bcf1f81
  Rnd9  f(R8=b60f92f1, SK8=00 00 00 00 00 00 00 00 ) = a729cb52
  Rnd10 f(R9=6c9a4c4d, SK7=00 00 00 00 00 00 00 00 ) = d074011a
  Rnd11 f(R10=667b93eb, SK6=00 00 00 00 00 00 00 00 ) = fb42476c
  Rnd12 f(R11=97d80b21, SK5=00 00 00 00 00 00 00 00 ) = cfc79ce9
  Rnd13 f(R12=a9bc0f02, SK4=00 00 00 00 00 00 00 00 ) = 9fa4b8a4
  Rnd14 f(R13=087cb385, SK3=00 00 00 00 00 00 00 00 ) = 398edfef
  Rnd15 f(R14=9032d0ed, SK2=00 00 00 00 00 00 00 00 ) = 0883812d
  Rnd16 f(R15=00ff32a8, SK1=00 00 00 00 00 00 00 00 ) = 6fb2d280
  FP:   L=616c6163, R=686b6172
 returns 616c6163686b6172
```

V- The internal state of the device at the 8th round:

```
Rnd8    f(R7=b60f92f1, SK8=00 00 00 00 00 00 00 00 ) = a729cb52
<code>

//Sources://
1- http://www.online-toolz.com/tools/text-hex-convertor.php
2- http://lpb.canb.auug.org.au/adfa/src/DEScalc/




Question 2. Inspect the key schedule phase for the given key and explain how
the sub keys are generated for each of the 16 steps.

The 56-bit key is stored in eight bytes, in which the least significant bit
of each byte is used for parity. The permutation, called PC1 ( Permuted
Choice 1) which divides the 56-bits key into two 28-bit key halves, acts on
bit 1 through 9, 9 through 15, and so on. Since the first entry in the table
is "57", this means that the 57th bit of the original key K becomes the
first bit of the permuted key K+. The 49th bit of the original key becomes
the second bit of the permuted key. The 4th bit of the original key is the
last bit of the permuted key. Note only 56 bits of the original key appear
in the permuted key.

So how it works using our secret key "0x0101010101010101" :

First we have to convert the key from hex to binary

<code>
Hex:
0101010101010101

Binary:
0000 0001 0000 0001 0000 0001 0000 0001
0000 0001 0000 0001 0000 0001 0000 0001
```

Now the 64-bit key is permuted according to this table, PC-1

```
                            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
<code>

Since the first entry in the table is "57", this means that the 57th bit of
the original key K becomes the first bit of the permuted key K+. The 49th
bit of the original key becomes the second bit of the permuted key. The 4th
bit of the original key is the last bit of the permuted key. Note only 56
bits of the original key appear in the permuted key.

From the Original Key K:
00000001 00000001 00000001 00000001 00000001 00000001 00000001 00000001

We get the 56-bit permutation, because "the input key size is 64 which
contains 56 bit key and 8 bit parity. Parity bits are 8th bit of every 8 bit
(one byte). So they are all multiple of eight: 8, 16, 24, 32, 40, 48, 56 and
64. Permuted choice PC-1 is used to remove these bits from the 64 bit input
key. So PC-1 gives 56 bits as output."

K+:
0000000 0000000 0000000 0000000 0000000 0000000 0000000 0000000

Now, we split this key into left and right halves, where each half has 28
bits.

From the permuted key K+, we get

L0 = 0000000 0000000 0000000 0000000
R0 = 0000000 0000000 0000000 0000000

With L0 and R0 defined, we now create sixteen blocks Ln and Rn, 1<=n<=16.
Each pair of blocks Ln and Rn is formed from the previous pair Ln-1 and
Rn-1, respectively, for n = 1, 2, ..., 16, using the following schedule of
"left shifts" of the previous block. To do a left shift, move each bit one
place to the left, except for the first bit, which is cycled to the end of
the block.

<code>

                      Iteration      Number of
                       Number      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
```

This means that L3 and R3 are obtained from L2 and R2 by two left shifts, and L16 and R16 are obtained from L15 and R15 by one left shift.

From original pair pair L0 and R0 we get:

```
L0 = 0000000 0000000 0000000 0000000
R0 = 0000000 0000000 0000000 0000000

L1 = 0000000 0000000 0000000 0000000
R1 = 0000000 0000000 0000000 0000000

L2 = 0000000 0000000 0000000 0000000
R2 = 0000000 0000000 0000000 0000000

L3 = 0000000 0000000 0000000 0000000
R3 = 0000000 0000000 0000000 0000000

L4 = 0000000 0000000 0000000 0000000
R4 = 0000000 0000000 0000000 0000000

L5 = 0000000 0000000 0000000 0000000
R5 = 0000000 0000000 0000000 0000000

L6 = 0000000 0000000 0000000 0000000
R6 = 0000000 0000000 0000000 0000000

L7 = 0000000 0000000 0000000 0000000
R7 = 0000000 0000000 0000000 0000000

L8 = 0000000 0000000 0000000 0000000
R8 = 0000000 0000000 0000000 0000000

L9 = 0000000 0000000 0000000 0000000
R9 = 0000000 0000000 0000000 0000000
```

```
L10 = 0000000 0000000 0000000 0000000
R10 = 0000000 0000000 0000000 0000000

L11 = 0000000 0000000 0000000 0000000
R11 = 0000000 0000000 0000000 0000000

L12 = 0000000 0000000 0000000 0000000
R12 = 0000000 0000000 0000000 0000000

L13 = 0000000 0000000 0000000 0000000
R13 = 0000000 0000000 0000000 0000000

L14 = 0000000 0000000 0000000 0000000
R14 = 0000000 0000000 0000000 0000000

L15 = 0000000 0000000 0000000 0000000
R15 = 0000000 0000000 0000000 0000000

L16 = 0000000 0000000 0000000 0000000
R16 = 0000000 0000000 0000000 0000000
```

We now form the keys Kn, for 1<=n<=16, by applying the following permutation table to each of the concatenated pairs LnRn. Each pair has 56 bits, but PC-2 only uses 48 of these.

```
                              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
```

For the first key we have L1R1 = 0000000 0000000 0000000 0000000 0000000 0000000 0000000 0000000

which, after we apply the permutation PC-2, becomes K1 = 000000 000000 000000 000000 000000 000000 000000 000000

Since we have all zero's all the sub keys (SK) also only zero's: K2-16: 000000 000000 000000 000000 000000 000000 000000 000000

Sources:

1- https://upload.wikimedia.org/wikipedia/commons/thumb/0/06/DES-key-schedule.png/250px-DES-key-schedule.png

2- http://www.quadibloc.com/crypto/co040201.htm

3- http://page.math.tu-berlin.de/~kant/teaching/hess/krypto-ws2006/des.htm

4- http://www.binaryhexconverter.com/decimal-to-binary-converter

Question 3. Comment on the behavior of DES when using the given key.

Since we have 16 subkeys only zeros keys and all identical. The encryption function is self-inverting; that is, despite encrypting once giving a secure-looking cipher text, encrypting twice produces the original plaintext.

Sources:

1- https://en.wikipedia.org/wiki/Weak_key#Weak_keys_in_DES

2 AES

Question 4. Identify the Shannon diffusion element(s).

First, lets define what Shannon diffusion element means. “It means that if we change a single bit of the plaintext, then (statistically) half of the bits in the ciphertext should change, and similarly, if we change one bit of the ciphertext, then approximately one half of the plaintext bits should change. Since a bit can have only two states, when they are all re-evaluated and changed from one seemingly random position to another, half of the bits will have changed state.”

A better way to clarify the Shannon diffusion element(s) is by following a really good figures about how AES works in general and how Shannon diffusion element works in specific, our classmate Peter send this figures to the SSN email and I found it really good to answer this question:

Diffusion element (shift rows) Step:

Diffusion element (Mix Columns) Step:

Question 5. Also identify the Shannon confusion element(s).

First, lets define what Shannon confusion element means. It “means that each binary digit (bit) of the ciphertext should depend on several parts of the key, obscuring the connections between the two.”

Again, we refer to that amazing figure to clarify the confusion phase.

In other word, confusion refers to making the relationship between the key and the ciphertext as complex and involved as possible.

Sources:

1- https://en.wikipedia.org/wiki/Confusion_and_diffusion

2- http://www.moserware.com/2009/09/stick-figure-guide-to-advanced.html

3- http://cryptography.wikia.com/wiki/Confusion_and_diffusion