Inlamning2

Viktor

2024-03-07

Contents


```
%md
# Inlämning 2
Kryptering 1, hösten 2023
```

1 Inlämning 2

Kryptering 1, hösten 2023

```
%md
**Namn:** Viktor Listi
```

Namn: Viktor Listi

```
# Laddar användbar fil genom att ställa markören i sällen och tryck \
    Shift+Enter.
load('kryptering1.sage')
```

*** kryptering1.sage: Funktioner för kursen Kryptering 1 (SageMath) ***

```
#1)
#year = 2002
#month = 2
#day = 27

#X = Mod(5*month + 7*day + 11, 16)
#Y = Mod(3*month - 9*day + 13, 16)
#K = Mod(year*month - day, 1024)
#m = Mod(year*day + month, 256)
#X = 0010
#Y = 0000
#K = 1110001001
#m = 00101000
```

```
#2)
from sage.crypto.block_cipher.sdes import SimplifiedDES
sDES = SimplifiedDES()

K = sDES.string_to_list("1110001001")
m = sDES.string_to_list("00101000")

K1 = sDES.subkey(K)
K2 = sDES.subkey(K,2)
print("2)")
print(K1)
print(K2 ,"\n")


#3)
print("3)")
c = sDES.encrypt(m, K)

print(sDES.list_to_string(c) , "\n")
```

```
2)
[1, 1, 0, 0, 0, 1, 1, 0]
[0, 1, 0, 0, 1, 1, 0, 1]
3)
01111111
```

```
#4)
#a)
    #IP = (1,5,2,0,3,7,4,6)
    #Om L0 = 0,0,0,0
    #då måste bitarna 1,5,2,0 i m vara 0.
#b)
    #Bitarna c3,c4,c6,c7 kommer att avslöja utdatan från funtkionen \
   f1

#5)
    #f1(R0,K1) = Y = 0000
    #ger oss att P0 = 0 och P1 = 0


#6)
#X = 0010
#dX = 0101
#X' = 0010 o 0101 = 0111

#a)
    #D = 27 => i = 1
```

```
    #X -> rad0,kol2 => Y = 2 = 10
    #X' -> rad1,kol3 => = 3 = 11

#b)
    #dY = 10 o 11 = 01

#7)
    #(0110, 11) har 12/16 chans => 3/4


#8)
    #dX påverkas inte då både X och X' är konstruerade efter nyckel \
   steget.

#9)
    #Om vi utgår från av dX = 0101 så måste bitarna på plats 0 samt \
   2 vara lika        och 1 samt 3 vara olika. då 1 o 1 = 0, 0 o 0 =\
    0, 1 o 0 = 1, 0 o 1 = 1

#10)
    #IP = (1,5,2,0,3,7,4,6)
    #Om vi vill ha dm = m o m' där IP(dm) = (L,R) där R = dR0 = 0101
    #bitarna 3,7,4,6 ur dm de intressanta
    #Då måste bitarna 3 samt 4 vara lika och bitarna 7 samt 6 vara \
   olika i m och      m'

#11)
    #sbox => y = [y0,y1,y2,y3]
    #F2 och de två andra delarna sätts ihop blir det x = [x0,x1,x2,\
   x3,x4,x5,x6,x7] --> y kan vara x4,x5,x,6,x7

    #IP^-1 --> c = [x3,x0,x2,x4,x6,x1,x7,x5]
    #De bitar som motsvarar dY i c är då c3,c4,c6,c7
```