



**VIT**<sup>®</sup>  
**Vellore Institute of Technology**  
(Deemed to be University under section 3 of UGC Act, 1956)

**Aditi Patra**

**21BIT0125**

**Lab 1**

**Intelligent Cryptography**

**L55+56**

**CODE:**

```
import random

# Sender side

n = int(input("Enter a number to encrypt: "))

M = bin(n)

n1 = random.randint(1,n)

print("The random number is",n1)

R=bin(n1)

print("The random number in binary form is",R)

if n1 >= n:
```

```
print("Since the second number is greater than or equal to the to be encrypt value, we cannot continue")
```

```
else:
```

```
    #Data split
```

```
    ds = n - n1
```

```
    print("The value after data split",ds)
```

```
    S = bin(ds)
```

```
    print("The value of binary data split value",S)
```

```
    K = random.randint(2**2048, 2**4096) #key should be in 2k bits
```

```
    print("The random key is",K)
```

```
    CT1 = n1^K #XOR gate
```

```
    #Encryption
```

```
    CT2 = ds^K #XOR gate
```

```
    print("The value of cloud A",CT1)
```

```
    print("The value of cloud B",CT2)
```

```
    print("The binary value of cloud A is", bin(CT1))
```

```
    print("The binary value of cloud B is", bin(CT2))
```

```
    # Storing them in seperate files!
```

```
    with open("cloudA.txt", "w") as file:
```

```
        file.write(str(CT1))
```

```
    with open("cloudB.txt", "w") as file:
```

```
        file.write(str(CT2))
```

```
    with open("key.txt", "w") as file:
```

```
        file.write(str(K))
```

```

# Receiver side

with open("cloudA.txt", "r") as file:

    CT1 = int(file.read())

with open("cloudB.txt", "r") as file:

    CT2 = int(file.read())

with open("key.txt", "r") as file:

    K = int(file.read())


CA1 =CT1 ^ K #XOR gate

print("The value of new cloud A",CA1)

#Decryption

CA2 =CT2 ^ K #XOR gate

print("The value of new cloud B",CA2)

#Data merge

M1 = CA1 + CA2

F = bin(M1)

print("The decrypted value is", M1)

print("The binary form of decrypted value is",F)


# To cross check

if M1 == n:

    print("The process is correct")

else:

    print("The process is wrong")

```

```
cryptofinal.py - C:/Users/patra/Desktop/books/5th sem/infosec/cryptofinal.py (3.9.6)
File Edit Format Run Options Window Help
import random

# Sender side
n = int(input("Enter a number to encrypt: "))
M = bin(n)
n1 = random.randint(1,n)
print("The random number is",n1)
R=bin(n1)
print("The random number in binary form is",R)
if n1 >= n:
    print("Since the second number is greater than or equal to the to be encrypt value, we cannot continue")
else:
    #Data split
    ds = n - n1
    print("The value after data split",ds)
    S = bin(ds)
    print("The value of binary data split value",S)
    K = random.randint(2**2048, 2**4096) #key should be in 2k bits
    print("The random key is",K)
    CT1 = n1^K #XOR gate
    #Encryption
    CT2 = ds^K #XOR gate
    print("The value of cloud A",CT1)
    print("The value of cloud B",CT2)
    print("The binary value of cloud A is", bin(CT1))
    print("The binary value of cloud B is", bin(CT2))

    # Storing them in separate files!
    with open("cloudA.txt", "w") as file:
        file.write(str(CT1))
    with open("cloudB.txt", "w") as file:
        file.write(str(CT2))
    with open("key.txt", "w") as file:
        file.write(str(K))

# Receiver side
with open("cloudA.txt", "r") as file:
    CT1 = int(file.read())
with open("cloudB.txt", "r") as file:
    CT2 = int(file.read())
with open("key.txt", "r") as file:
    K = int(file.read())

CA1 =CT1 ^ K #XOR gate
print("The value of new cloud A",CA1)
#Decryption
CA2 =CT2 ^ K #XOR gate
print("The value of new cloud B",CA2)
#Data merge
M1 = CA1 + CA2
F = bin(M1)
print("The decrypted value is", M1)
print("The binary form of decrypted value is",F)

# To cross check
if M1 == n:
    print("The process is correct")
else:
    print("The process is wrong")
```

Ln: 34 Col: 26

## Key:

```
key - Notepad
File Edit Format View Help
2730767807634246062924728186303101899897800852350899670830302512142763115515115893226151377592710480835450728894949
3731353205650672350965586482164571693994527711479768629630172356860178375528692799183303364480843411259612620316240

Ln 1, Col 1 100% Windows (CRLF) UTF-8
```

Cloud A:

cloudA - Notepad

File Edit Format View Help

2730767807634246062924728186303101899897800852350899670830302512142763115515115893226151377592710480835450728894949  
3731353205650672350965586482164571693994527711479768629630172356860178375528692799183303364480843411259612620316240

Ln 1, Col 1100%Windows (CRLF)UTF-8

Cloud B:

cloudB - Notepad

File Edit Format View Help

2730767807634246062924728186303101899897800852350899670830302512142763115515115893226151377592710480835450728894949  
3731353205650672350965586482164571693994527711479768629630172356860178375528692799183303364480843411259612620316240

Ln 1, Col 1100%Windows (CRLF)UTF-8

OUTPUT:

IDLE Shell 3.9.6

File Edit Shell Debug Options Window Help

>>>  
===== RESTART: C:/Users/patra/Desktop/books/5th sem/infosec/cryptofinal.py =====  
Enter a number to encrypt: 69  
The random number is 44  
The random number in binary form is 0b101100  
The value after data split 25  
The value of binary data split value 0b11001  
The random key is 2730767807634246062924728186303101899897800852350899670830302512142763115515115893226151377592710480835450728894949325671121688099  
858255752613299155194545841236356758425584493502065972474682496765189150136634913456342355128902748858294462818555117745859880597069117736515395969  
25644284481903986253652658521612389481931046808735489589993143346948114438079835294048206241731447038393463370652566598826446166152656090194513714131  
68312753185380349250899580048234726846020193467364269439654311882139881417251000422426102788735043971622384672618471986973096137660037047200565266805  
21128171347082270372401214710474309453926654195484557135603806402751707483691195870166723380532243172051971763907025418368544836217852807604648981244  
8606020528919026459298408626851777507918577676635315311626949256936505573859811535062076974736724987216130031892122862342835700635462277208524333633  
0161289221734601540575157536083135279148463300057578939248811066834249516286862157567102324520226171251783221208612211393012319181752165824880467333  
73135320565067235096558648216457169399452771147976862963017235686017837552869279918330336448084341125961262031624083444783598116820579852121789748  
30103866782245203123112225762945538519112313184775375342144  
The value of cloud A 27307678076342460629247281863031018998978008523508996708303025121427631155151158932261513775927104808354507288949493256711216880  
099858255752613299155194545841236356758425584493502065972474682496765189150136634913456342355128902748858294462818555117745859880597069117736515395  
96925644284481903986253652658521612389481931046808735489589993143346948114438079835294048206241731447038393463370652566598826446166152656090194513714  
13168312753185380349250899580048234726846020193467364269439654311882139881417251000422426102788735043971622384672618471986973096137660037047200565266  
80521128171347082270372401214710474309453926654195484557135603806402751707483691195870166723380532243172051971763907025418368544836217852807604648981  
2448606020528919026459298408626851777507918577676635315311626949256936505573859811535062076974736724987216130031892122862342835700635462277208524333  
6330161289221734601540575157536083135279148463300057578939248811066834249516286862157567102324520226171251783221208612211393012319181752165824880467  
33373135320565067235096558648216457169399452771147976862963017235686017837552869279918330336448084341125961262031624083444783598116820579852121789748  
03430103866782245203123112225762945538519112313184775375342144  
The value of cloud B 27307678076342460629247281863031018998978008523508996708303025121427631155151158932261513775927104808354507288949493256711216880  
099858255752613299155194545841236356758425584493502065972474682496765189150136634913456342355128902748858294462818555117745859880597069117736515395  
96925644284481903986253652658521612389481931046808735489589993143346948114438079835294048206241731447038393463370652566598826446166152656090194513714  
13168312753185380349250899580048234726846020193467364269439654311882139881417251000422426102788735043971622384672618471986973096137660037047200565266  
80521128171347082270372401214710474309453926654195484557135603806402751707483691195870166723380532243172051971763907025418368544836217852807604648981  
2448606020528919026459298408626851777507918577676635315311626949256936505573859811535062076974736724987216130031892122862342835700635462277208524333  
6330161289221734601540575157536083135279148463300057578939248811066834249516286862157567102324520226171251783221208612211393012319181752165824880467  
33373135320565067235096558648216457169399452771147976862963017235686017837552869279918330336448084341125961262031624083444783598116820579852121789748  
03430103866782245203123112225762945538519112313184775375342144  
The binary value of cloud A is Squeezed text (52 lines).  
The binary value of cloud B is Squeezed text (52 lines).  
The value of new cloud A 44  
The value of new cloud B 25  
The decrypted value is 69  
The binary form of decrypted value is 0b1000101  
The process is correct  
>>>

Ln: 20 Col: 4