

Network Security

Programming Exercise-2

Dharani Kumar S. (2021039)

Arnav Singh (2021019)

Algorithm to Pick the Project

$$k = (A1 + A2) \% 2$$

$$k = (1019 + 1039) \% 2 = 0$$

Assigned Project:– Data Encryption Standard

DES

The **Data Encryption Standard (DES** [/ˈdɪːtʃɪˈɛs, dɛz/](#)) is a [symmetric-key algorithm](#) for the [encryption](#) of digital data. Although its short key length of 56 bits makes it too insecure for modern applications, it has been highly influential in the advancement of [cryptography](#).

We chose to implement it in Python with bitwise operations and predefined permutation tables.

Key Components

- Permutations: Initial and final permutation.
- Key Scheduling: Generates 16 round keys using PC-1, PC-2 and shifts.
- Rounds: Each round applies XOR, substitution, permutation and other operations.
- Decryption: Goes through the same rounds in reverse.

Implementation

```
def desProcess(inputHex, keyHex, action='encrypt', printRounds=False):
    inputBits=hexToBin(inputHex)
    keyBits=hexToBin(keyHex)

    permuted=applyPerm(inputBits, DES_TABLES["permutations"]["initial"])
    left=permuted[:32]
    right=permuted[32:]

    roundDetails=[]
    roundDetails.append({
        "Round":0,
        "Left":binToHex(left),
        "Right":binToHex(right),
        "Combined":binToHex(left+right),
        "Round Key":""
    })

    roundKeys=genRoundKeys(keyBits)
    if action == 'decrypt':
        roundKeys.reverse()

    for i in range(16):
        newLeft,newRight=desRound(left, right, roundKeys[i])

        if i == 15:
            left=newRight
            right=newLeft
        else:
            left=newLeft
            right=newRight

        roundDetails.append({
            "Round":i+1,
            "Left":binToHex(left),
            "Right":binToHex(right),
            "Combined":binToHex(left+right),
            "Round Key":binToHex(roundKeys[i])
        })

    combined=left+right
    finalOutput=applyPerm(combined, DES_TABLES["permutations"]["final"])
    finalOutputHex=binToHex(finalOutput)

    if printRounds:
        print(tabulate(roundDetails, headers="keys", tablefmt="pretty"))

    return {"final":finalOutputHex, "round_details":roundDetails}
```


Output

Enter the message to be encrypted (hexadecimal):0123456789ABCDEF
Enter the 64-bit key for encryption (hexadecimal):133457799BBCDF1

--- Encryption Process ---

Round	Left	Right	Combined	Round Key
0	CC00CCFF	F0AAF0AA	CC00CCFFF0AAF0AA	
1	F0AAF0AA	EF4A6544	F0AAF0AAEF4A6544	1B02EFFC7072
2	EF4A6544	CC017709	EF4A6544CC017709	79AED9DBC9E5
3	CC017709	A25C0BF4	CC017709A25C0BF4	55FC8A42CF99
4	A25C0BF4	77220045	A25C0BF477220045	72ADD6DB351D
5	77220045	8A4FA637	772200458A4FA637	7CEC07EB53A8
6	8A4FA637	E967CD69	8A4FA637E967CD69	63A53E507B2F
7	E967CD69	064ABA10	E967CD69064ABA10	EC84B7F618BC
8	064ABA10	D5694B90	064ABA10D5694B90	F78A3AC13BFB
9	D5694B90	247CC67A	D5694B90247CC67A	E0DBEBEDE781
10	247CC67A	B7D5D7B2	247CC67AB7D5D7B2	B1F347BA464F
11	B7D5D7B2	C5783C78	B7D5D7B2C5783C78	215FD3DED386
12	C5783C78	75BD1858	C5783C7875BD1858	7571F59467E9
13	75BD1858	18C3155A	75BD185818C3155A	97C5D1FABA41
14	18C3155A	C28C960D	18C3155AC28C960D	5F43B7F2E73A
15	C28C960D	43423234	C28C960D43423234	BF918D3D3F0A
16	0A4CD995	43423234	0A4CD99543423234	CB3D8B0E17F5

Encrypted:85E813540F0AB405

Encrypted:85E813540F0AB405

--- Decryption Process ---

Round	Left	Right	Combined	Round Key
0	0A4CD995	43423234	0A4CD99543423234	
1	43423234	C28C960D	43423234C28C960D	CB3D8B0E17F5
2	C28C960D	18C3155A	C28C960D18C3155A	BF918D3D3F0A
3	18C3155A	75BD1858	18C3155A75BD1858	5F43B7F2E73A
4	75BD1858	C5783C78	75BD1858C5783C78	97C5D1FABA41
5	C5783C78	B7D5D7B2	C5783C78B7D5D7B2	7571F59467E9
6	B7D5D7B2	247CC67A	B7D5D7B2247CC67A	215FD3DED386
7	247CC67A	D5694B90	247CC67AD5694B90	B1F347BA464F
8	D5694B90	064ABA10	D5694B90064ABA10	E0DBEBEDE781
9	064ABA10	E967CD69	064ABA10E967CD69	F78A3AC13BFB
10	E967CD69	8A4FA637	E967CD698A4FA637	EC84B7F618BC
11	8A4FA637	77220045	8A4FA63777220045	63A53E507B2F
12	77220045	A25C0BF4	77220045A25C0BF4	7CEC07EB53A8
13	A25C0BF4	CC017709	A25C0BF4CC017709	72ADD6DB351D
14	CC017709	EF4A6544	CC017709EF4A6544	55FC8A42CF99
15	EF4A6544	F0AAF0AA	EF4A6544F0AAF0AA	79AED9DBC9E5
16	CC00CCFF	F0AAF0AA	CC00CCFFF0AAF0AA	1B02EFFC7072

Decrypted:0123456789ABCDEF

References

- <https://page.math.tu-berlin.de/~kant/teaching/hess/krypto-ws2006/des.htm>
- <https://www.geeksforgeeks.org/data-encryption-standard-des-set-1/>