# CRYPTO LAB-2

## DES Encryption and Decryption Using Socket Programming

**NAME: ABISHEK K G** **COURSE CODE: BCSE309P**

**REG NO: 23BCE1739** **DATE: 20/12/2025**

### Aim

To implement the Data Encryption Standard (DES) algorithm manually and demonstrate secure communication between a client and a server using socket programming.

### Objective

- To understand symmetric key encryption using DES
- To encrypt plaintext data at the client side
- To securely transmit encrypted data to the server
- To decrypt the received ciphertext at the server side
- To observe and analyze each round of the DES algorithm

### Description

Data Encryption Standard (DES) is a symmetric key block cipher that operates on a 64-bit block size using a 56-bit effective key. In this experiment, DES is implemented manually without using any built-in cryptographic libraries.

The client encrypts the plaintext using DES and sends the ciphertext to the server through a socket connection.

The server receives the encrypted data, decrypts it using the same secret key, and retrieves the original plaintext.

The output of all 16 DES rounds is displayed for better understanding of the internal working of the algorithm.

**Key Scheduling**

- Input a 64-bit key
- Remove parity bits using PC-1 to obtain a 56-bit key
- Split the key into two halves
- Perform left circular shifts as per shift schedule
- Generate 16 round keys (48 bits each) using PC-2

**Client–Server Communication**

**Client Side**

- Accept plaintext from the user
- Encrypt plaintext using DES
- Send ciphertext to the server using sockets

**Server Side**

- Receive ciphertext from the client
- Decrypt ciphertext using DES
- Display decrypted plaintext

**Input**

Plaintext message entered by the client (8 characters)

**Output**

- Encrypted ciphertext generated at the client side
- Round-wise DES output (L and R values)
- Decrypted plaintext displayed at the server side

**Program:**

```
import java.io.*;

import java.net.*;

import java.util.*;


public class DESLab {


    // ================= DES TABLES ==================


    static final int[] IP = {

        58,50,42,34,26,18,10,2, 60,52,44,36,28,20,12,4,

        62,54,46,38,30,22,14,6, 64,56,48,40,32,24,16,8,

        57,49,41,33,25,17,9,1, 59,51,43,35,27,19,11,3,

        61,53,45,37,29,21,13,5, 63,55,47,39,31,23,15,7

    };


    static final int[] IP_INV = {

        40,8,48,16,56,24,64,32, 39,7,47,15,55,23,63,31,

        38,6,46,14,54,22,62,30, 37,5,45,13,53,21,61,29,

        36,4,44,12,52,20,60,28, 35,3,43,11,51,19,59,27,

        34,2,42,10,50,18,58,26, 33,1,41,9,49,17,57,25

    };


    static final int[] E = {

        32,1,2,3,4,5, 4,5,6,7,8,9,

        8,9,10,11,12,13, 12,13,14,15,16,17,

        16,17,18,19,20,21, 20,21,22,23,24,25,

        24,25,26,27,28,29, 28,29,30,31,32,1
```

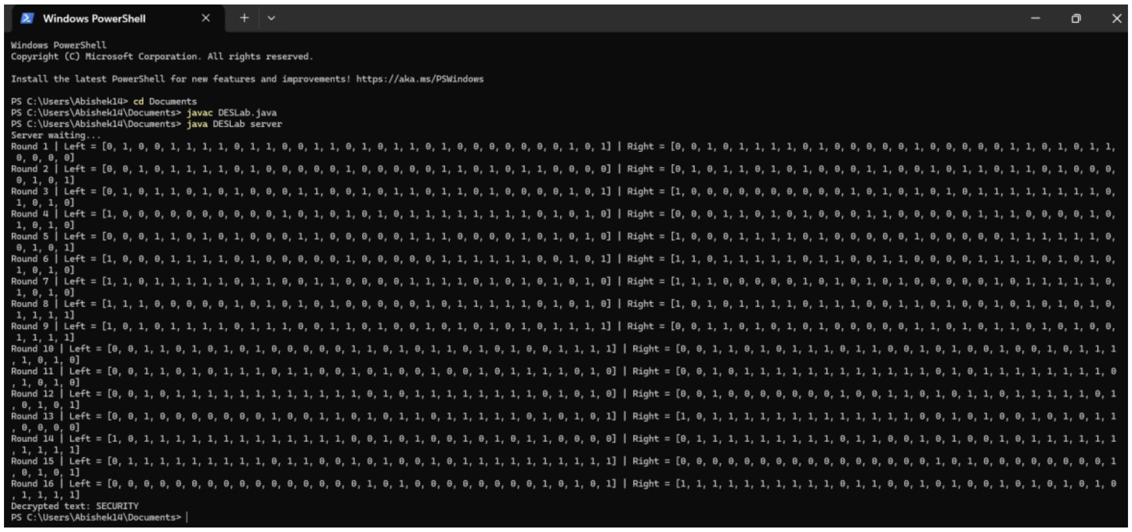
```
};

static final int[] P = {
    16,7,20,21, 29,12,28,17,
    1,15,23,26, 5,18,31,10,
    2,8,24,14, 32,27,3,9,
    19,13,30,6, 22,11,4,25
};

static final int[] PC1 = {
    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
};

static final int[] PC2 = {
    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
};

static final int[] shiftPlan = {
    1,1,2,2,2,2,2,2,
    1,2,2,2,2,2,2,1
};

// ================= BASIC HELPERS =================

static int[] shuffleBits(int[] data, int[] table) {
    int[] result = new int[table.length];
    for (int i = 0; i < table.length; i++)
        result[i] = data[table[i] - 1];
    return result;
}

static int[] doXor(int[] a, int[] b) {
    int[] temp = new int[a.length];
    for (int i = 0; i < a.length; i++)
        temp[i] = a[i] ^ b[i];
    return temp;
}

static int[] rotateLeft(int[] bits, int times) {
    int[] out = new int[bits.length];
    for (int i = 0; i < bits.length; i++)
        out[i] = bits[(i + times) % bits.length];
    return out;
}

static int[] textToBits(String msg) {
    byte[] arr = msg.getBytes();
    int[] bits = new int[64];
    for (int i = 0; i < 8; i++)
        for (int j = 0; j < 8; j++)
            bits[i * 8 + j] = (arr[i] >> (7 - j)) & 1;
    return bits;
}

static String bitsToText(int[] bits) {
    byte[] arr = new byte[8];
    for (int i = 0; i < 8; i++)
        for (int j = 0; j < 8; j++)
            arr[i] |= bits[i * 8 + j] << (7 - j);
    return new String(arr);
}

// ================= KEY GENERATION =================

static int[][] makeAllKeys(int[] originalKey) {
    int[] noParityKey = shuffleBits(originalKey, PC1);
    int[] leftKey = Arrays.copyOfRange(noParityKey, 0, 28);
    int[] rightKey = Arrays.copyOfRange(noParityKey, 28, 56);

    int[][] allRoundKeys = new int[16][];

    for (int i = 0; i < 16; i++) {
        leftKey = rotateLeft(leftKey, shiftPlan[i]);
        rightKey = rotateLeft(rightKey, shiftPlan[i]);

        int[] mergedKey = new int[56];
        System.arraycopy(leftKey, 0, mergedKey, 0, 28);
        System.arraycopy(rightKey, 0, mergedKey, 28, 28);

        allRoundKeys[i] = shuffleBits(mergedKey, PC2);
    }
    return allRoundKeys;
}

// ================= DES FUNCTION =================

static int[] runDES(int[] dataBlock, int[][] allRoundKeys, boolean isDecrypt) {

    int[] permutedData = shuffleBits(dataBlock, IP);
    int[] leftPart = Arrays.copyOfRange(permutedData, 0, 32);
    int[] rightPart = Arrays.copyOfRange(permutedData, 32, 64);

    for (int round = 0; round < 16; round++) {
        int[] temp = rightPart;
        int keyIndex = isDecrypt ? 15 - round : round;

        rightPart = doXor(leftPart, shuffleBits(rightPart, E));
        leftPart = temp;

        System.out.println("Round " + (round + 1) +
                " | Left = " + Arrays.toString(leftPart) +
                " | Right = " + Arrays.toString(rightPart));
    }

    int[] combined = new int[64];
    System.arraycopy(rightPart, 0, combined, 0, 32);
    System.arraycopy(leftPart, 0, combined, 32, 32);

    return shuffleBits(combined, IP_INV);
}

// ================ SOCKET PART ================

public static void main(String[] args) throws Exception {

    int[] secretKeyBits = textToBits("ABCDEFGH");
    int[][] allRoundKeys = makeAllKeys(secretKeyBits);

    if (args[0].equalsIgnoreCase("server")) {
```

```
    ServerSocket server = new ServerSocket(5000);
    System.out.println("Server waiting...");
    Socket socket = server.accept();

    ObjectInputStream input = new ObjectInputStream(socket.getInputStream());
    int[] encryptedStuff = (int[]) input.readObject();

    int[] decryptedBits = runDES(encryptedStuff, allRoundKeys, true);
    System.out.println("Decrypted text: " + bitsToText(decryptedBits));

    socket.close();
    server.close();

} else {

    Socket socket = new Socket("localhost", 5000);
    Scanner scan = new Scanner(System.in);

    System.out.print("Enter 8-character message: ");
    String message = scan.nextLine();

    int[] messageBits = textToBits(message);
    int[] encryptedStuff = runDES(messageBits, allRoundKeys, false);

    ObjectOutputStream output = new ObjectOutputStream(socket.getOutputStream());
    output.writeObject(encryptedStuff);
```

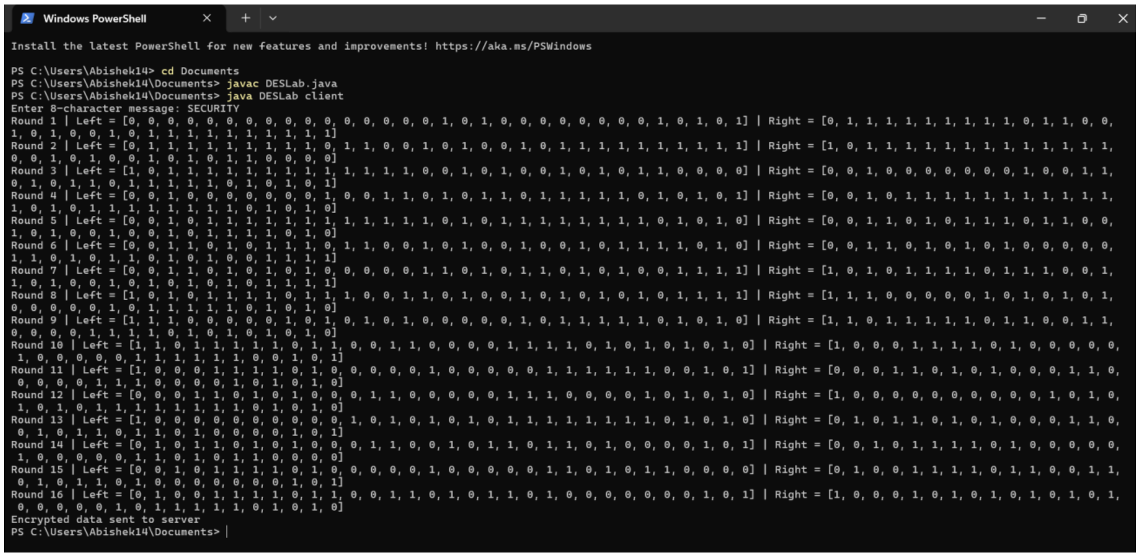
```
            System.out.println("Encrypted data sent to server");


            socket.close();
        }
    }
}
```

**Result**

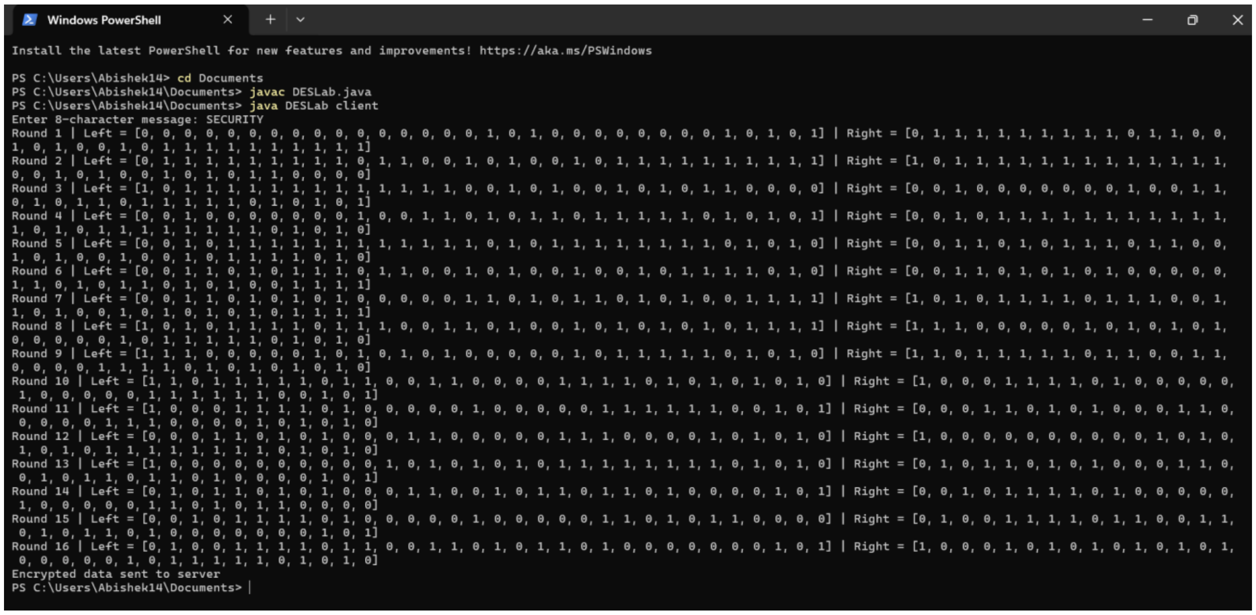
The plaintext entered at the client side was successfully encrypted using DES and transmitted securely to the server.
The server correctly decrypted the ciphertext and retrieved the original plaintext.
The program executed successfully and the output obtained was as expected.



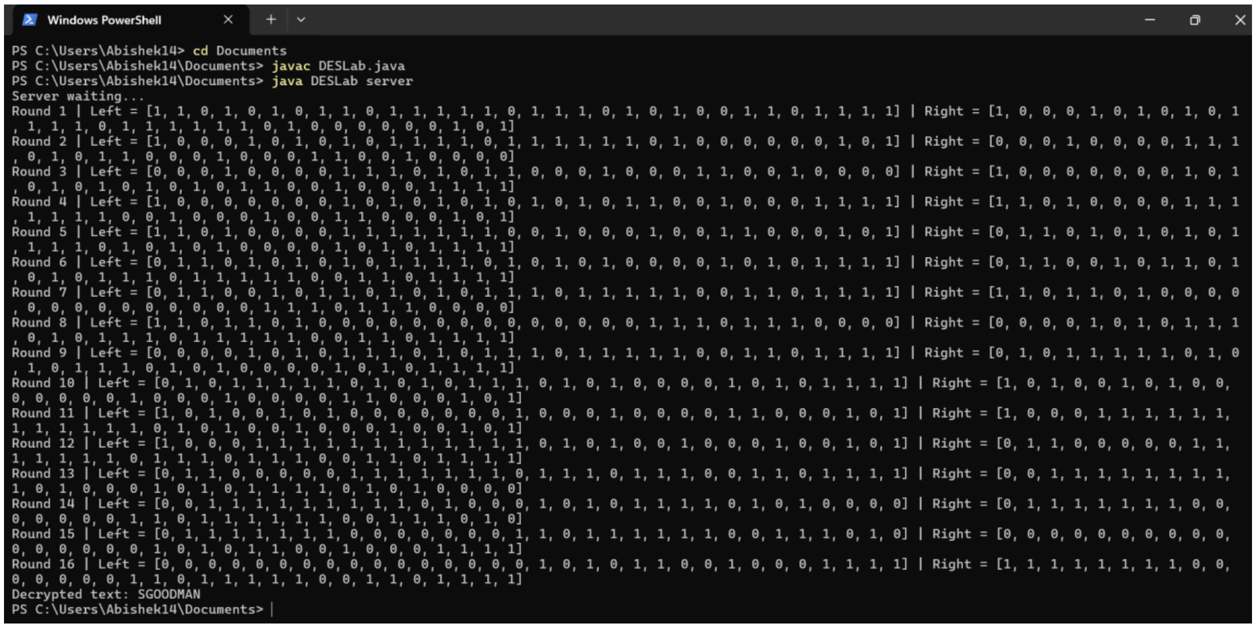
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Abishek14> cd Documents
PS C:\Users\Abishek14\Documents> javac DESLab.java
PS C:\Users\Abishek14\Documents> java DESLab server
Server waiting...
Round 1 | Left = [0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1] | Right = [0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1,
 0, 0, 0, 0]
Round 2 | Left = [0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0] | Right = [0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0,
 0, 1, 0, 1]
Round 3 | Left = [0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1] | Right = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0,
 1, 0, 1, 0]
Round 4 | Left = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0] | Right = [0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0,
 1, 0, 1, 0]
Round 5 | Left = [0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0] | Right = [1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0,
 0, 1, 0, 1]
Round 6 | Left = [1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1] | Right = [1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0,
 1, 0, 1, 0]
Round 7 | Left = [1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0] | Right = [1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0,
 1, 0, 1, 0]
Round 8 | Left = [1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0] | Right = [1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0,
 1, 1, 1, 1]
Round 9 | Left = [1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1] | Right = [0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0,
 1, 1, 1, 1]
Round 10 | Left = [0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1] | Right = [0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1
, 1, 0, 1, 0]
Round 11 | Left = [0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0] | Right = [0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0
, 1, 0, 1, 0]
Round 12 | Left = [0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0] | Right = [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1
, 0, 1, 0, 1]
Round 13 | Left = [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1] | Right = [1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1
, 0, 0, 0, 0]
Round 14 | Left = [1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0] | Right = [0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1
, 1, 1, 1, 1]
Round 15 | Left = [0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1] | Right = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1
, 0, 1, 0, 1]
Round 16 | Left = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1] | Right = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0
, 1, 1, 1, 1]
Decrypted text: SECURITY
PS C:\Users\Abishek14\Documents> |
```



```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Abishek14> cd Documents
PS C:\Users\Abishek14\Documents> javac DESLab.java
PS C:\Users\Abishek14\Documents> java DESLab client
Enter 8-character message: SECURITY
Round 1 | Left = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1] | Right = [0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0,
1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
Round 2 | Left = [0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1] | Right = [1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0]
Round 3 | Left = [1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0] | Right = [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1,
0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1]
Round 4 | Left = [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1] | Right = [0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0]
Round 5 | Left = [0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0] | Right = [0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0,
1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0]
Round 6 | Left = [0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0] | Right = [0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0,
1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1]
Round 7 | Left = [0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1] | Right = [1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1,
1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1]
Round 8 | Left = [1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1] | Right = [1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1,
0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0]
Round 9 | Left = [1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0] | Right = [1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1,
0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0]
Round 10 | Left = [1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0] | Right = [1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0,
 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1]
Round 11 | Left = [1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1] | Right = [0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0,
 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0]
Round 12 | Left = [0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0] | Right = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0,
 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0]
Round 13 | Left = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0] | Right = [0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0,
 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1]
Round 14 | Left = [0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1] | Right = [0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0,
 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0]
Round 15 | Left = [0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0] | Right = [0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1,
 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1]
Round 16 | Left = [0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1] | Right = [1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1,
 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0]
Encrypted data sent to server
PS C:\Users\Abishek14\Documents> |
```



```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Abishek14> cd Documents
PS C:\Users\Abishek14\Documents> javac DESLab.java
PS C:\Users\Abishek14\Documents> java DESLab client
Enter 8-character message: SECURITY
Round 1 | Left = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1] | Right = [0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0,
1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
Round 2 | Left = [0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1] | Right = [1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0]
Round 3 | Left = [1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0] | Right = [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1,
0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1]
Round 4 | Left = [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1] | Right = [0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0]
Round 5 | Left = [0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0] | Right = [0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0,
1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0]
Round 6 | Left = [0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0] | Right = [0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0,
1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1]
Round 7 | Left = [0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1] | Right = [1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1,
1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1]
Round 8 | Left = [1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1] | Right = [1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1,
0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0]
Round 9 | Left = [1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0] | Right = [1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1,
0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0]
Round 10 | Left = [1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0] | Right = [1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0,
 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1]
Round 11 | Left = [1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1] | Right = [0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0,
 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0]
Round 12 | Left = [0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0] | Right = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0,
 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0]
Round 13 | Left = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0] | Right = [0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0,
 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1]
Round 14 | Left = [0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1] | Right = [0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0,
 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0]
Round 15 | Left = [0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0] | Right = [0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1,
 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1]
Round 16 | Left = [0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1] | Right = [1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1,
 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0]
Encrypted data sent to server
PS C:\Users\Abishek14\Documents> |
```



```
PS C:\Users\Abishek14> cd Documents
PS C:\Users\Abishek14\Documents> javac DESLab.java
PS C:\Users\Abishek14\Documents> java DESLab server
Server waiting...
Round 1 | Left = [1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1] | Right = [1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1
, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1]
Round 2 | Left = [1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1] | Right = [0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1
, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0]
Round 3 | Left = [0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0] | Right = [1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1
, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1]
Round 4 | Left = [1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1] | Right = [1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1
, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1]
Round 5 | Left = [1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1] | Right = [0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1
, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1]
Round 6 | Left = [0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1] | Right = [0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1
, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1]
Round 7 | Left = [0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1] | Right = [1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0]
Round 8 | Left = [1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0] | Right = [0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1
, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1]
Round 9 | Left = [0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1] | Right = [0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0
, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1]
Round 10 | Left = [0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1] | Right = [1, 0, 1, 0, 0, 1, 0, 1, 0, 0,
0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1]
Round 11 | Left = [1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1] | Right = [1, 0, 0, 0, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1]
Round 12 | Left = [1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1] | Right = [0, 1, 1, 0, 0, 0, 0, 0, 1, 1,
1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1]
Round 13 | Left = [0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1] | Right = [0, 0, 1, 1, 1, 1, 1, 1, 1, 1,
1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0]
Round 14 | Left = [0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0] | Right = [0, 1, 1, 1, 1, 1, 1, 1, 0, 0,
0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0]
Round 15 | Left = [0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0] | Right = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1]
Round 16 | Left = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1] | Right = [1, 1, 1, 1, 1, 1, 1, 1, 0, 0,
0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1]
Decrypted text: SGOODMAN
PS C:\Users\Abishek14\Documents> |
```