MALLA REDDY UNIVERSITY

III Year B.Tech– II Semester **L/T/P/C**
-/-/3/1.5

(MR20-1CS0188) CRYPTOGRAPHY & NETWORK SECURITY LAB

Name: Subhapreet Patro
Roll No.: 2211CS010547
Group: 3

Week-3

3. User A want to send the message "Meet me very urgently" to user B by using DES algorithms encrypt it at sender end and decrypt it at receiver end.

AIM: To apply Data Encryption Standard (DES) Algorithm for a practical application like User Message Encryption.

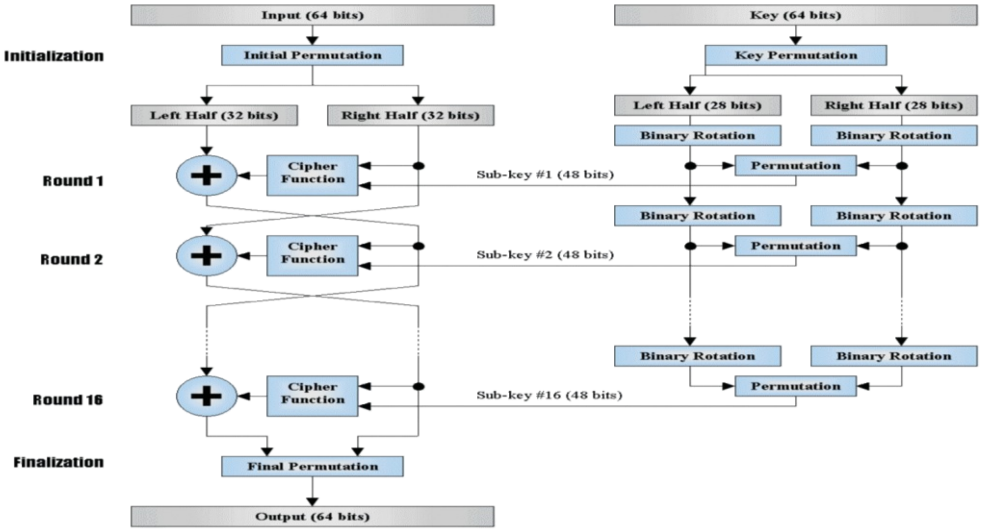
OBJECTIVE: To understand the encryption and decryption the given message by using Data Encryption Standard encryption algorithm.

THEORY:

DES is a symmetric encryption system that uses 64-bit blocks, 8 bits of which are used for parity checks. The key therefore has a "useful" length of 56 bits, which means that only 56 bits are actually used in the algorithm. The algorithm involves carrying out combinations, substitutions and permutations between the text to be encrypted and the key, while making sure the operations can be performed in both directions. The key is ciphered on64 bits and made of 16 blocks of 4 bits, generally denoted k1 to k16. Given that "only" 56 bits are actually used for encrypting, there can be 256 different keys.

The main parts of the algorithm are as follows:

1. Fractioning of the text into 64-bit blocks
2. Initial permutation of blocks
3. Breakdown of the blocks into two parts: left and right, named L and R
4. Permutation and substitution steps repeated 16 times
5. Re-joining of the left and right parts then inverse initial permutation



ALGORITHM:

STEP-1: Read the 64-bit plain text.

STEP-2: Split it into two 32-bit blocks and store it in two different arrays.

STEP-3: Perform XOR operation between these two arrays.

STEP-4: The output obtained is stored as the second 32-bit sequence and the original second 32-bit sequence forms the first part.

STEP-5: Thus the encrypted 64-bit cipher text is obtained in this way. Repeat the same process for the remaining plain text characters

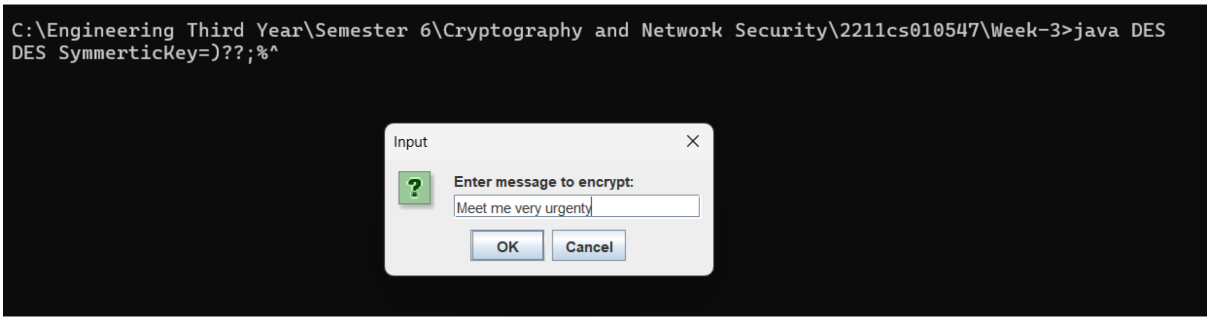
PROGRAM

```
import javax.swing.*;
import java.security.SecureRandom;
import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
import java.util.Random;

class DES {
   byte[] skey = new byte[1000];
   String skeystring;
   static byte[] raw;
   String inputmessage, encryptedata, decryptedmessage;

   public DES() {
      try {
         generatesymmetrickey();
         inputmessage = JOptionPane.showInputDialog(null, "Enter message to encrypt:");
```

```
        byte[] ibyte = inputmessage.getBytes();
        byte[] ebyte = encrypt(raw, ibyte);
        String encrypteddata = new String(ebyte);
        System.out.println("Encrypted message:" + encrypteddata);
        JOptionPane.showMessageDialog(null, "Encrypted Data" + "\n" + encrypteddata);
        byte[] dbyte = decrypt(raw, ebyte);
        String decryptedmessage = new String(dbyte);
        System.out.println("Decrypted message:" + decryptedmessage);
        JOptionPane.showMessageDialog(null, "Decrypted Data " + "\n" +
  decryptedmessage);
    } catch (Exception e) {
        System.out.println(e);
    }
}

void generatesymmetrickey() {
    try {
        Random r = new Random();
        int num = r.nextInt(10000);
        String knum = String.valueOf(num);
        byte[] knumb = knum.getBytes();
        skey = getRawKey(knumb);
        skeystring = new String(skey);
        System.out.println("DES SymmetricKey=" + skeystring);
    } catch (Exception e) {
        System.out.println(e);
    }
}

private static byte[] getRawKey(byte[] seed) throws Exception {
    KeyGenerator kgen = KeyGenerator.getInstance("DES");
    SecureRandom sr = SecureRandom.getInstance("SHA1PRNG");
    sr.setSeed(seed);
    kgen.init(56, sr);
    SecretKey skey = kgen.generateKey();
    raw = skey.getEncoded();
    return raw;
}

private static byte[] encrypt(byte[] raw, byte[] clear) throws Exception {
    SecretKey seckey = new SecretKeySpec(raw, "DES");
    Cipher cipher = Cipher.getInstance("DES");
    cipher.init(Cipher.ENCRYPT_MODE, seckey);
    byte[] encrypted = cipher.doFinal(clear);
```

```
        return encrypted;
    }

    private static byte[] decrypt(byte[] raw, byte[] encrypted) throws Exception {
        SecretKey seckey = new SecretKeySpec(raw, "DES");
        Cipher cipher = Cipher.getInstance("DES");
        cipher.init(Cipher.DECRYPT_MODE, seckey);
        byte[] decrypted = cipher.doFinal(encrypted);
        return decrypted;
    }

    public static void main(String args[]) {
        new DES();
    }
}
```

OUTPUT:

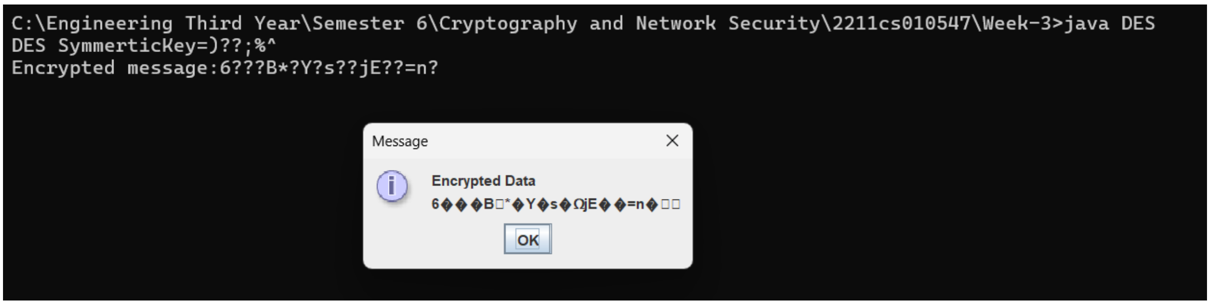


```
C:\Engineering Third Year\Semester 6\Cryptography and Network Security\2211cs010547\Week-3>java DES
DES SymmerticKey=)??;%^
```

Input

Enter message to encrypt:

Meet me very urgenty

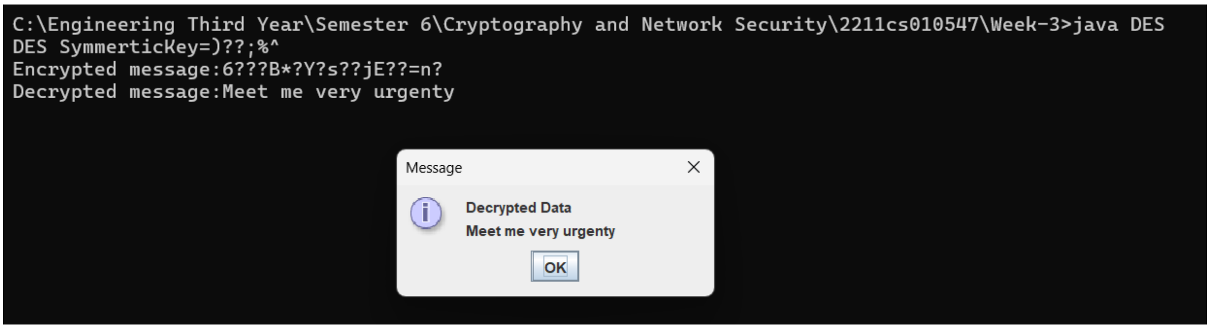


```
C:\Engineering Third Year\Semester 6\Cryptography and Network Security\2211cs010547\Week-3>java DES
DES SymmerticKey=)??;%^
Encrypted message:6???B*?Y?s??jE??=n?
```

Message

Encrypted Data

6���B□*�Y�s�ΩjE��=n�□□



```
C:\Engineering Third Year\Semester 6\Cryptography and Network Security\2211cs010547\Week-3>java DES
DES SymmerticKey=)??;%^
Encrypted message:6???B*?Y?s??jE??=n?
Decrypted message:Meet me very urgenty
```

Message

Decrypted Data

Meet me very urgenty

LAB VIVA QUESTIONS:

1. What are symmetric and asymmetric key systems?
2. What is private key and public key?
3. Compare stream cipher with block cipher
4. List out different types of encryption algorithms?
5. What is the Data Encryption Standard (DES)?