PRACTICAL NO. 2

Title: Write a Java/C/C++/Python program to perform encryption and decryption using the method of Transposition technique.

HARDWARE AND SOFTWARE REQUIREMENT:

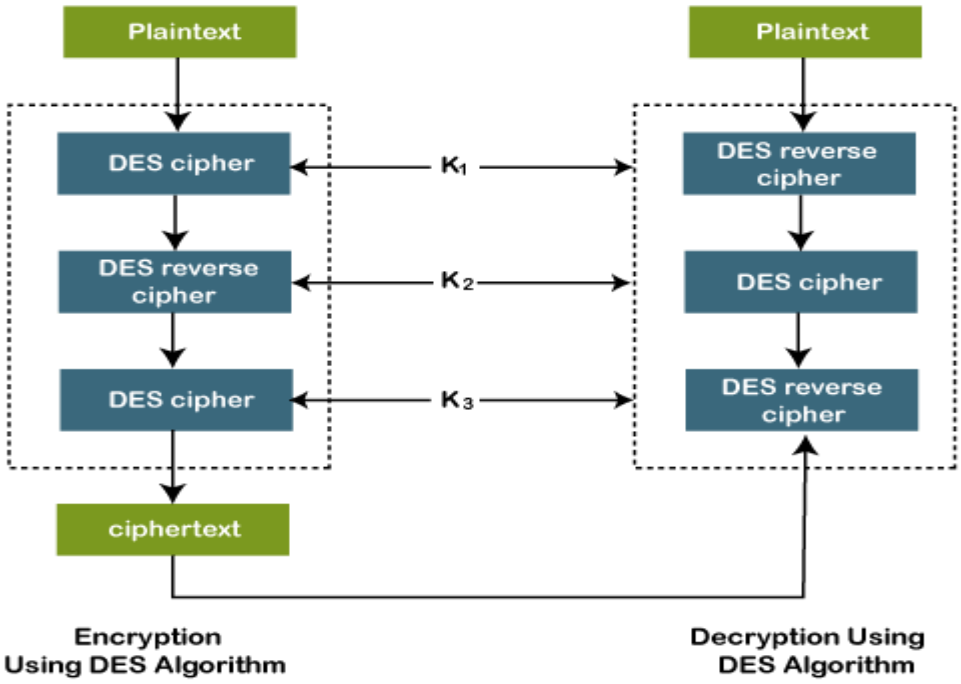
1. Intel based Desktop PC: - RAM of 512 MB
2. Notepad/Notepad ++ editor
3. Net beans / Eclipse

THEORY:

DES is a block cipher technique which encrypts data in blocks (64 bit size), i.e. 64 bits of PLAINTEXT message goes as the input to DES, which produces 64 bits of CIPHERTEXT message. DES uses a 56 bit key. DES is actually based on the two fundamental concepts of cryptography: substitution and transposition. It consists of 16 steps called as a rounds. Each round is responsible for performing the steps of substitution and transposition.

DES stands for Data Encryption Standard. It is a symmetric-key block cipher algorithm used to encrypt and decrypt data. It is developed by the IBM team in early 1970. It accepts the plaintext in 64-bit blocks and changes it into the ciphertext that uses the 64-bit keys to encrypt the data. The algorithm uses the same key to encrypt and decrypt the data.

It is based on LUCIFER (also known as Feistel block cipher algorithm) which is a direct predecessor of the DES algorithm. It is developed by eminent scholar and researcher Horst Feistel at IBM. It provides high security by using a 128-bit key block and a 128-bit block size. The DES algorithm uses the 16 rounds of the Feistel structure. The structure uses a unique key for each round. Finally, in 1976, it was approved by the federal encryption standard.



Program:

```
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.KeyGenerator;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.SecretKey;
public class DES
{
public static void main(String[] args)
 {
try{
 System.out.println("Message Encryption Using DES Algorithm\n-------");
 KeyGenerator keygenerator = KeyGenerator.getInstance("DES");
 SecretKey myDesKey = keygenerator.generateKey();
 Cipher desCipher;
desCipher = Cipher.getInstance("DES/ECB/PKCS5Padding");
 desCipher.init(Cipher.ENCRYPT_MODE, myDesKey);
 byte[] text = "Secret Information ".getBytes();
 System.out.println("Message [Byte Format] : " + text);
 System.out.println("Message : " + new String(text));
 byte[] textEncrypted = desCipher.doFinal(text);
 System.out.println("Encrypted Message: " + textEncrypted);
 desCipher.init(Cipher.DECRYPT_MODE, myDesKey);
 byte[] textDecrypted = desCipher.doFinal(textEncrypted);
 System.out.println("Decrypted Message: " + new
String(textDecrypted));
```

```
}catch(NoSuchAlgorithmException e){
e.printStackTrace();
}catch(NoSuchPaddingException e){
e.printStackTrace();
}catch(InvalidKeyException e){
e.printStackTrace();
}catch(IllegalBlockSizeException e){
e.printStackTrace();
}catch(BadPaddingException e){
e.printStackTrace();
}
}
}
```

Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.4529]
(c) Microsoft Corporation. All rights reserved.

D:\JavaFd>javac DES.java

D:\JavaFd>java DES
Message Encryption Using DES Algorithm
-------
Message [Byte Format] : [B@5a77a7f9
Message : Secret Information
Encrypted Message: [B@4cbfea1d
Decrypted Message: Secret Information

D:\JavaFd>_
```

Conclusion:

We conclude that the basic steps of the algorithm are: At last, write and read the encrypted or decrypted data by using the CipherOutputStream and CipherInputStream. Let's implement the DES algorithm in a Java program and see how data is encrypted and decrypted using the algorithm.