DOP: / /2023 **DOS: / /2023**

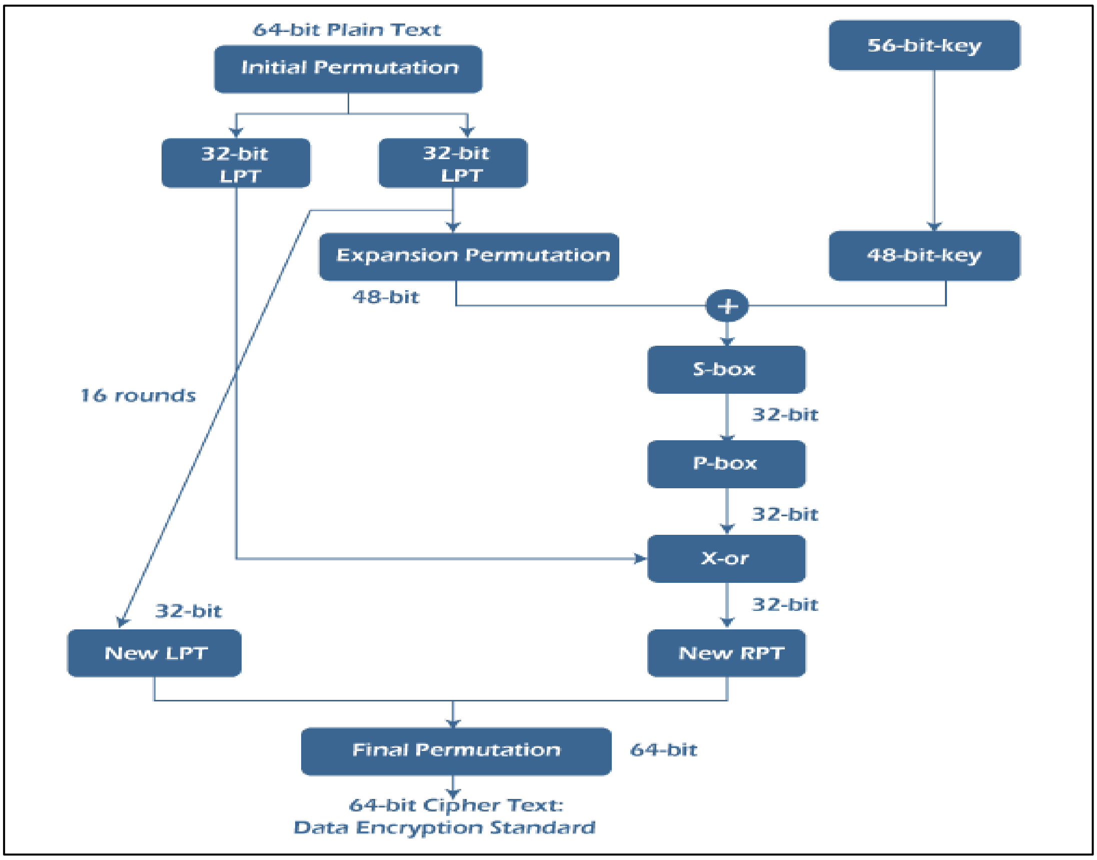
Experiment No: 04

Aim: Encrypt long messages using various modes of operation using AES or DES.

Theory:

🔷What is DES?:

The Data Encryption Standard (DES) is a symmetric-key block cipher. In the year 1977, DES is published by the National Institute of Standards and Technology (NIST). It is based on the Feistel structure in which the plaintext is separated into two halves. It takes input as 64-bit plaintext and a 56-bit key to produce 64-bit ciphertext. Before processing, the entire plain text is separated into two pieces of 32 bits each, and the same operations are done on each portion. Each piece goes through 16 rounds of operations before the final permutation is used to obtain the 64-bit ciphertext.



Expansions, permutations, and substitutions are some of the functions used in the rounds, as well as an XOR operation with a round key. Decryption is done in the same way as encryption but in the opposite sequence. Although DES was regarded to be less safe for encrypting highly confidential data of government because it uses a smaller shared key, triples-DES was invented to counter this. Still, it was also not considered a good algorithm because it encrypts data very slowly. In DES, even a minor change in the input text results in a completely new ciphertext.

🔷Advantage of DES:

There are various advantage of DES which is as follows −

- DES has been around a long time (since 1977), even no actual weaknesses have been discovered and the most effective attack is still brute force.
- DES is an official United States Government standard. The Government is needed to re-certify, DES every five years and ask it be restored if essential.
- DES is also an ANSI and ISO standard. Because DES was designed to run on 1977 hardware, it is rapid in hardware and associatively quick in software.
- It supports functionality to save a file in an encrypted format which can only be accessed by supporting the correct password.
- It can change the system to create the directories password protected.

🔷Disadvantage of DES:

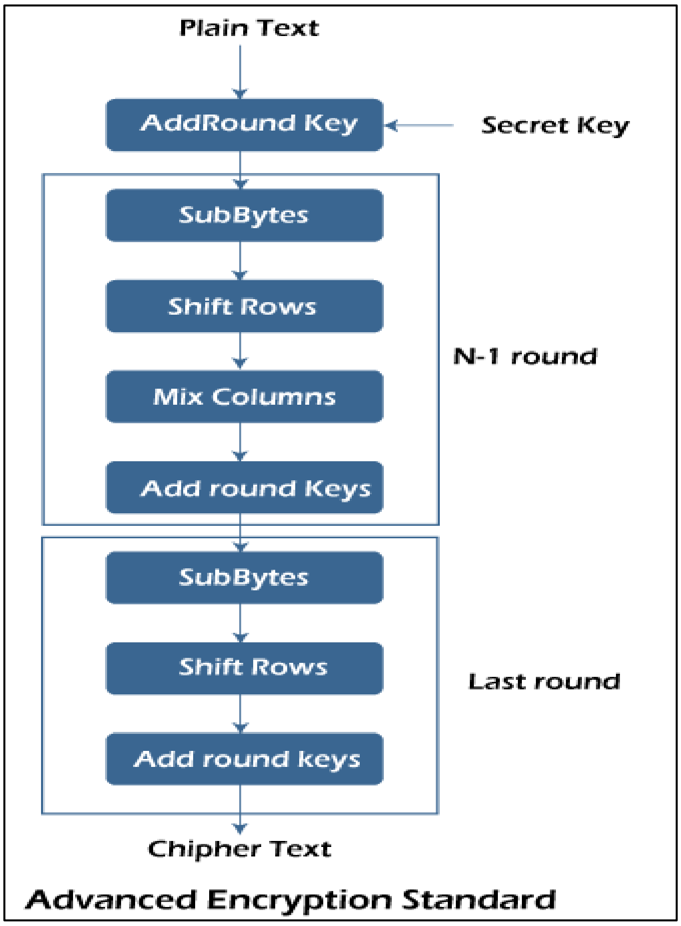
There are various disadvantage of DES which is as follows −

- The 56 bit key size is the largest defect of DES and the chips to implement one million of DES encrypt or decrypt operations a second are applicable (in 1993).
- Hardware implementations of DES are very quick.
- DES was not designed for application and therefore it runs relatively slowly.
- In a new technology, it is improving a several possibilities to divide the encrypted code, therefore AES is preferred than DES.

🔷What is AES?

Advanced Encryption Standard (AES) is also a symmetric key block cipher. The National Institute of Standard and Technology published AES in 2001. Because DES utilises a relatively short cipher key and the algorithm was quite slower, AES was introduced to replace it.

It is currently one of the most popular symmetric block cipher algorithms. It is at least six times faster than triple-DES encryption. Unlike DES, it is based on the "Substitution and Permutation'. It takes a step-by-step method. In AES, bytes are used instead of bits.



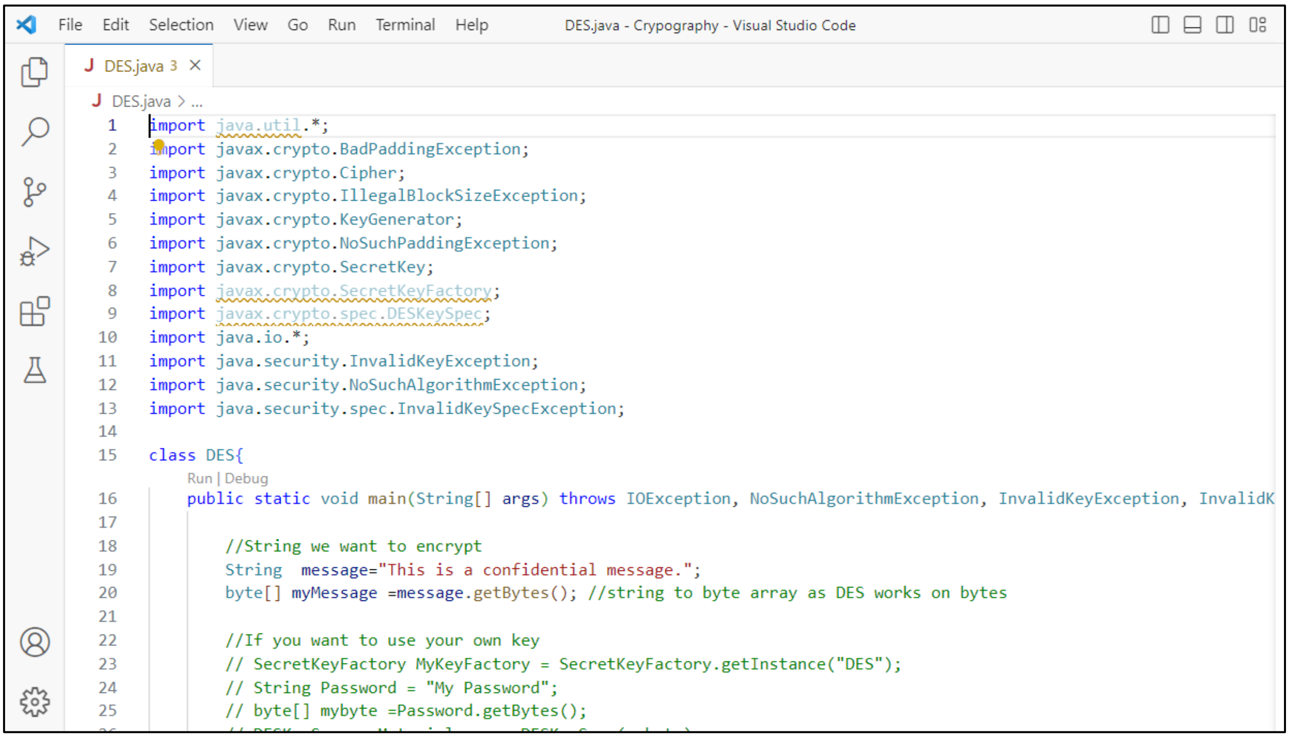
In AES, plain text is considered 126 bits equivalent to 16 bytes with a 128-bit secret key to generate a 44-bit matrix (having 4 rows and 4 columns). It then does 10 rounds after this step. Each round has its own subprocesses, with 9 rounds including Sub bytes, Shift Rows, Mix Columns and Add Round Keys. The 10th round includes all the above operations excluding 'Mix columns' in order to produce the 126-bit ciphertext.

The number of rounds in AES is determined by the key size, which is 10 for 128-bit keys, 12 for 192-bit keys, and 14 for 256-bit keys. We can use it in several protocols such as TLS, SSL and numerous modern application which need high encryption security. We can also use AES for hardware which needs high throughput.

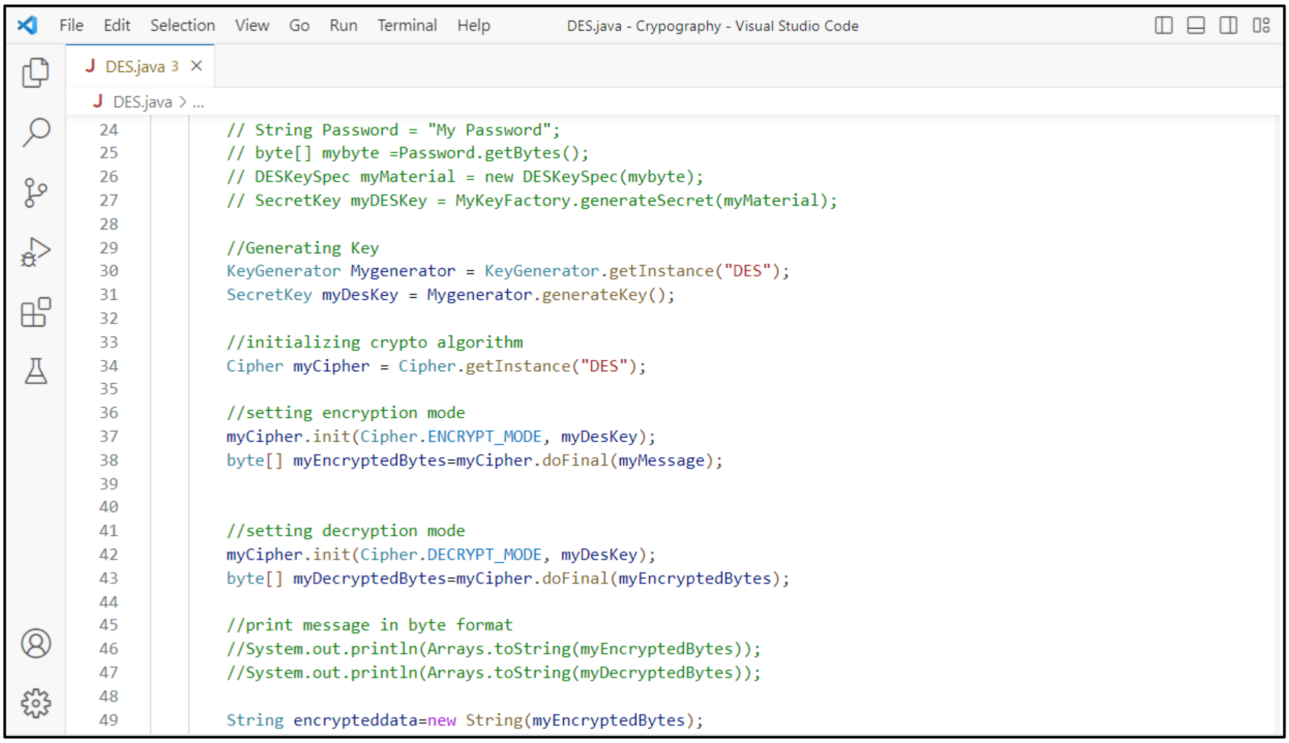
🔷Advantages of AES over 3DES:

- AES is more secure (it is less susceptible to cryptanalysis than 3DES).
- AES supports larger key sizes than 3DES's 112 or 168 bits.
- AES is faster in both hardware and software.
- AES's 128-bit block size makes it less open to attacks via the birthday problem than 3DES with its 64-bit block size.
- AES is required by the latest U.S. and international standards.

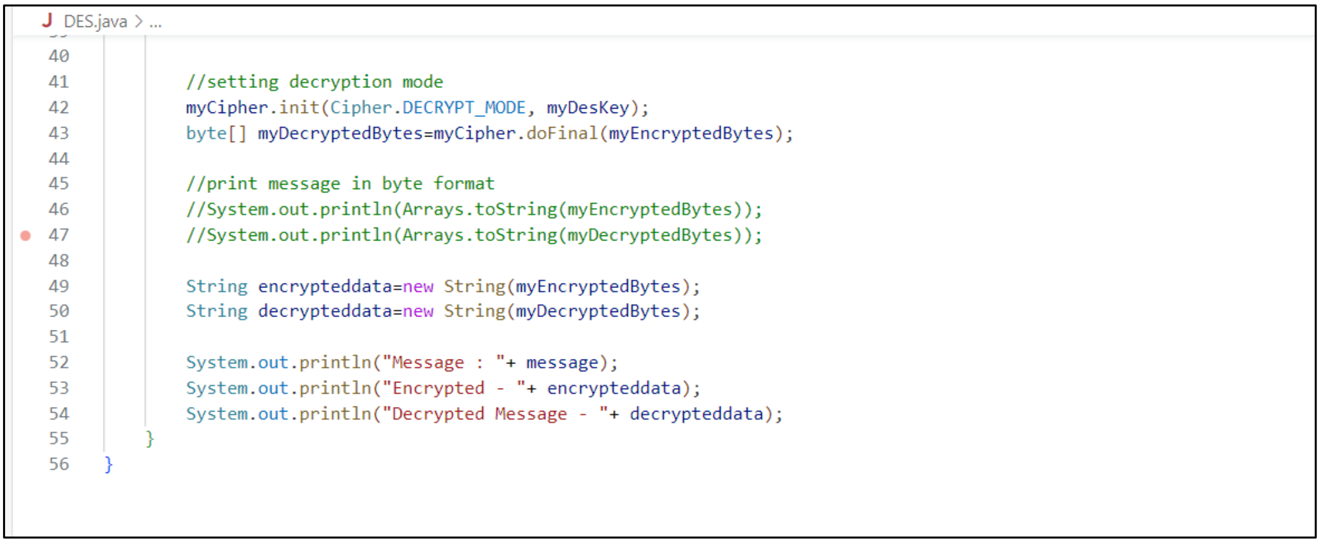
Input:


File Edit Selection View Go Run Terminal Help
DES.java - Crypography - Visual Studio Code
DES.java 3
DES.java > ...
Run | Debug

```java
import java.util.*;
import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.KeyGenerator;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.SecretKey;
import javax.crypto.SecretKeyFactory;
import javax.crypto.spec.DESKeySpec;
import java.io.*;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.security.spec.InvalidKeySpecException;

class DES{
    public static void main(String[] args) throws IOException, NoSuchAlgorithmException, InvalidKeyException, InvalidK

        //String we want to encrypt
        String  message="This is a confidential message.";
        byte[] myMessage =message.getBytes(); //string to byte array as DES works on bytes

        //If you want to use your own key
        // SecretKeyFactory MyKeyFactory = SecretKeyFactory.getInstance("DES");
        // String Password = "My Password";
        // byte[] mybyte =Password.getBytes();
```

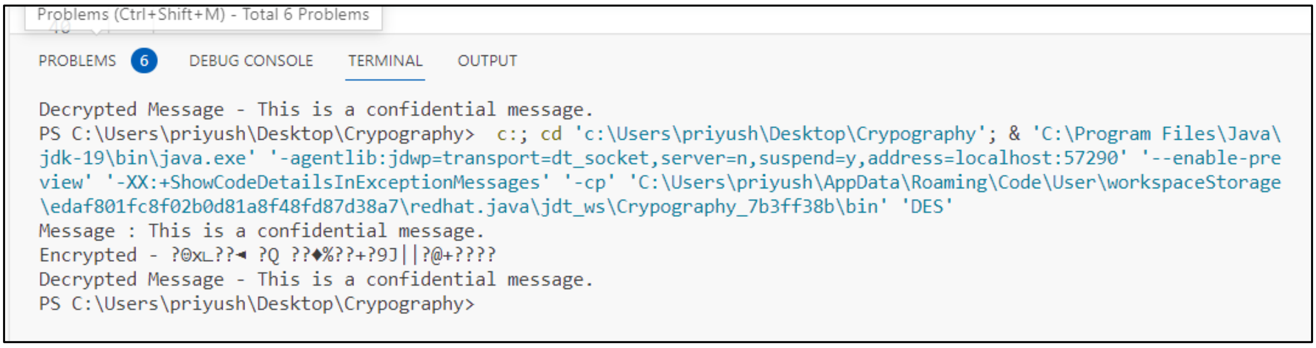

File Edit Selection View Go Run Terminal Help
DES.java - Crypography - Visual Studio Code
DES.java 3
DES.java > ...

```java
        // String Password = "My Password";
        // byte[] mybyte =Password.getBytes();
        // DESKeySpec myMaterial = new DESKeySpec(mybyte);
        // SecretKey myDESKey = MyKeyFactory.generateSecret(myMaterial);

        //Generating Key
        KeyGenerator Mygenerator = KeyGenerator.getInstance("DES");
        SecretKey myDesKey = Mygenerator.generateKey();

        //initializing crypto algorithm
        Cipher myCipher = Cipher.getInstance("DES");

        //setting encryption mode
        myCipher.init(Cipher.ENCRYPT_MODE, myDesKey);
        byte[] myEncryptedBytes=myCipher.doFinal(myMessage);


        //setting decryption mode
        myCipher.init(Cipher.DECRYPT_MODE, myDesKey);
        byte[] myDecryptedBytes=myCipher.doFinal(myEncryptedBytes);

        //print message in byte format
        //System.out.println(Arrays.toString(myEncryptedBytes));
        //System.out.println(Arrays.toString(myDecryptedBytes));

        String encrypteddata=new String(myEncryptedBytes);
```



```
        //setting decryption mode
        myCipher.init(Cipher.DECRYPT_MODE, myDesKey);
        byte[] myDecryptedBytes=myCipher.doFinal(myEncryptedBytes);

        //print message in byte format
        //System.out.println(Arrays.toString(myEncryptedBytes));
        //System.out.println(Arrays.toString(myDecryptedBytes));

        String encrypteddata=new String(myEncryptedBytes);
        String decrypteddata=new String(myDecryptedBytes);

        System.out.println("Message : "+ message);
        System.out.println("Encrypted - "+ encrypteddata);
        System.out.println("Decrypted Message - "+ decrypteddata);
    }
}
```

Output:



```
Decrypted Message - This is a confidential message.
PS C:\Users\priyush\Desktop\Crypography>  c:; cd 'c:\Users\priyush\Desktop\Crypography'; & 'C:\Program Files\Java\
jdk-19\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:57290' '--enable-pre
view' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\priyush\AppData\Roaming\Code\User\workspaceStorage
\edaf801fc8f02b0d81a8f48fd87d38a7\redhat.java\jdt_ws\Crypography_7b3ff38b\bin' 'DES'
Message : This is a confidential message.
Encrypted - ?☺xL??◄ ?Q ??♦%??+?9J||?@+????
Decrypted Message - This is a confidential message.
PS C:\Users\priyush\Desktop\Crypography>
```

Conclusion: - Thus we have implemented Encrypt long messages using various modes of operation using AES or **DES.**