Ex. No : 4	**Data Encryption Standard (DES) Algorithm (User Message Encryption)**

AIM:

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

Theory:

1. Create a DES Key.
2. Create a Cipher instance from Cipher class, specify the following information and separated by a slash (/).
   a. Algorithm name
   b. Mode (optional)
   c. Padding scheme (optional)
3. Convert String into ***Byte[]*** array format.
4. Make Cipher in encrypt mode, and encrypt it with ***Cipher.doFinal()*** method.
5. Make Cipher in decrypt mode, and decrypt it with ***Cipher.doFinal()*** method.

PROGRAM:

DES.java

```
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[] argv) {

                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:

```
Message Encryption Using DES Algorithm
-------------------------------------------------------------

Message [Byte Format] : [B@4dcbadb4
Message : Secret Information Encrypted
Message: [B@504bae78 Decrypted Message:
Secret Information
```

RESULT:

Thus the java program for DES Algorithm has been implemented and the output verified successfully.

Conclusion:

……………………..

Question

1. What is DES
2. What are the mode operation
3. How many rounds are use in DES
4. Key length of DES
5. What is initial permutation/matrix size

Ex. No : 5	**RSA Algorithm**

AIM:

To implement RSA (Rivest–Shamir–Adleman) algorithm by using HTML and Javascript.

ALGORITHM:

1. Choose two prime number p and q
2. Compute the value of n and **p**
3. Find the value of ***e*** (public key)
4. Compute the value of ***d*** (private key) using gcd()
5. Do the encryption and decryption
    a. Encryption is given as,

    $$c = t^{e} \bmod n$$

    b. Decryption is given as,

    $$t = c^{d} \bmod n$$

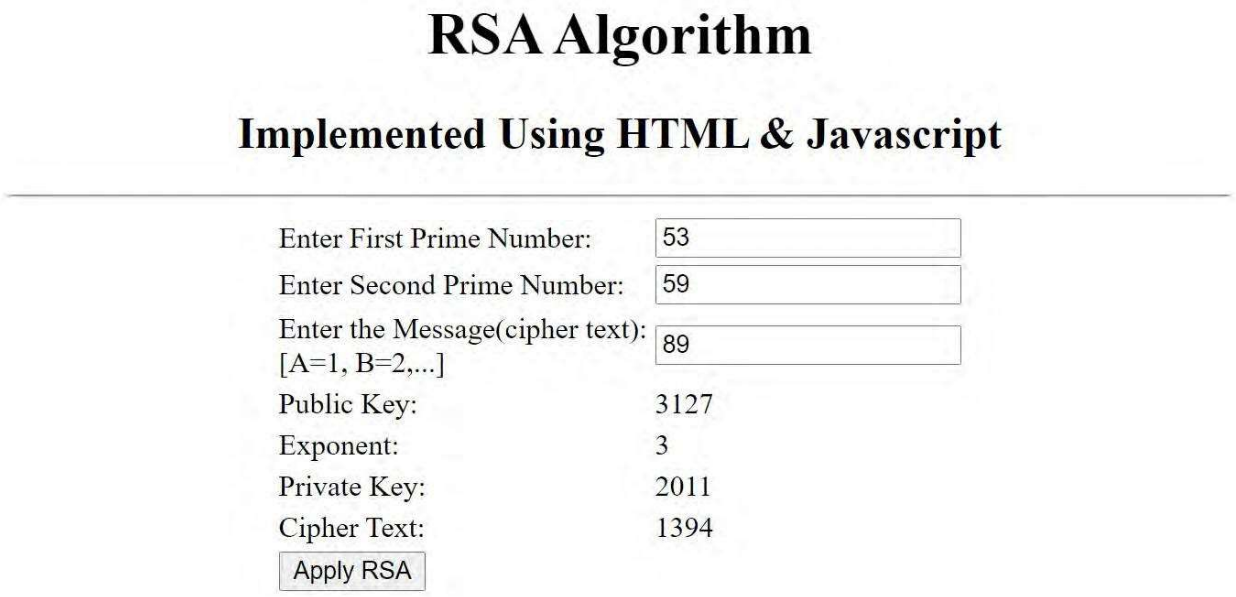
PROGRAM:

rsa.html

```
<html>

<head>
    <title>RSA Encryption</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>

<body>
    <center>
        <h1>RSA Algorithm</h1>
        <h2>Implemented Using HTML & Javascript</h2>
        <hr>
        <table>
            <tr>
                <td>Enter First Prime Number:</td>
                <td><input type="number" value="53" id="p"></td>
            </tr>
            <tr>
                <td>Enter Second Prime Number:</td>
                <td><input type="number" value="59" id="q"></p>
```

```
                    </td>
                </tr>
                <tr>
                    <td>Enter the Message(cipher text):<br>[A=1, B=2,...]</td>
                    <td><input type="number" value="89" id="msg"></p>
                    </td>
                </tr>
                <tr>
                    <td>Public Key:</td>
                    <td>
                        <p id="publickey"></p>
                    </td>
                </tr>
                <tr>
                    <td>Exponent:</td>
                    <td>
                        <p id="exponent"></p>
                    </td>
                </tr>
                <tr>
                    <td>Private Key:</td>
                    <td>
                        <p id="privatekey"></p>
                    </td>
                </tr>
                <tr>
                    <td>Cipher Text:</td>
                    <td>
                        <p id="ciphertext"></p>
                    </td>
                </tr>
                <tr>
                    <td><button onclick="RSA();">Apply RSA</button></td>
</tr>
</table>
        </center>
    </body>
    <script    type="text/javascript">    function
        RSA() {
            var gcd, p, q, no, n, t, e, i, x;
        gcd = function (a, b) { return (!b) ? a : gcd(b, a % b); }; p =
        document.getElementById('p').value;
        q =document.getElementById('q').value;
        no = document.getElementById('msg').value; n = p *
        q;
        t = (p - 1) * (q - 1);

        for (e = 2; e < t; e++) { if
            (gcd(e, t) == 1) {
                break;
            }
        }

        for (i = 0; i < 10; i++) { x =
            1 + i * t
            if (x % e == 0) { d
                = x / e; break;
            }
        }

        ctt = Math.pow(no, e).toFixed(0); ct =
        ctt % n;

        dtt = Math.pow(ct, d).toFixed(0); dt =
        dtt % n;

        document.getElementById('publickey').innerHTML        =        n;
        document.getElementById('exponent').innerHTML         =        e;
        document.getElementById('privatekey').innerHTML       =        d;
        document.getElementById('ciphertext').innerHTML = ct;
    }
</script>
</html>
```

OUTPUT:



RSA Algorithm

Implemented Using HTML & Javascript

Enter First Prime Number: 53

Enter Second Prime Number: 59

Enter the Message(cipher text): [A=1, B=2,...] 89

Public Key: 3127

Exponent: 3

Private Key: 2011

Cipher Text: 1394

RESULT:

Thus the RSA algorithm has been implemented using HTML & CSS and the output has been verified successfully.

Conclusion:

Question:

1. What is RSA Algorithm
2. Who discovered the RSA
3. What is Symmetric cryptography
4. What is Asymmetric cryptography
5. For p = 11 and q = 19 and choose e=17. Apply RSA algorithm where message=5 and find the cipher text.

.