IV BTECH -I SEMESTER

2024-25

Cryptography and Network Security Laboratory

Lab Manual

CRYPTOGRAPHY AND NETWORK SECURITY

1) Write a C program that contains a string(char pointer) with a value\Hello World'. The programs should XOR each character in this string with 0 and display the result.

2) Write a C program that contains a string (char pointer) with a value \Hello World'. The program should AND or and XOR each character in this string with 127 and display the result.

3) Write a C/Java program to perform encryption and decryption usingthe following algorithms:

    i. Ceaser Cipher
    ii. Substitution Cipher
    iii. Hill Cipher

4) Write a Java program to implement the DES algorithm logic.

5) Write a C/Java program to implement the Blowfish algorithm logic.
6) Write a C/Java program to implement the Rijndael algorithm logic.
7) Write the RC4 logic in Java Using Java Cryptography, encrypt text "Hello world" using Blowfish. Create your own key using Java key tool.
8) Write a Java program to implement RSA Algorithm.
9) Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript.
10) Calculate the message digest of a text using the SHA-1 algorithm in Java.
11) Calculate the message digest of a text using the MD5 algorithm in Java.

PROGRAMS

Week 1:

Write a C program that contains a string with a value\Hello World'.The program should XOR each character in this string with 0 and display the result.

PROGRAM:

```
#include<stdlib.>
void main()
{
char str[]="Hello World";
char str1[11];
int i,len;
len=strlen(str);
for(i=0;i<len;i++
)
{
str1[i]=str[i]^0;
printf("%c",str1[i]);
}
printf("\n");
}
```

Output: Hello World

Week 2

Write a C program that contains a string with a value \Hello World'. The program should AND and XOR each character in this string with 127 and display the result.

PROGRAM:

```
#include<stdio.h>
#include<stdlib.h>
void main()
{
char str[]="Hello World";
char str1[11],str2[11];
int i,len;
len = strlen(str);
for(i=0;i<len;i++)
{
str1[i] =str[i]&127;
printf("%c",str1[i]);
}
printf("\n");
for(i=0;i<len;i++)
{
str3[i]=str[i]^127;
printf("%c",str3[i]);
}
printf("\n");
}
```

OUTPUT :

Hello World

Week 3:

Write a C/Java program to performe encryption and decryption using the following algorithms:

a) Ceaser Cipher
b) Substitution Cipher
c) Hill Cipher

PROGRAM:

a) Ceaser Cipher

```
#include<stdio.h>
#include<string.h>
void main()
{
	int len,i,j,key;
	char alpha[]="abcdefghijklmnopqrstuvwxyz";
	char pt[10],ct[10],npt[10];
	printf("enter the plain text\n");
	scanf("%s",pt);
	printf("enter the key\n");
	scanf("%d",&key);
	printf("Encryption is as follows\n");
	len=strlen(pt);
	for(i=0;i<len;i++)
	{
		for(j=0;j<26;j++)
		{
			if(pt[i]==alpha[i])
			{
				k=(j+key)%26;
				ct[i]=alpha[k];
			}
		}
	}
	printf("Cipher Text is %s\n",ct);
	printf("Decryption is as follows\n");
	for(i=0;i<len;i++)
	{
		for(j=0;j<26;j++)
		{
			if(ct[i]==alpha[i])
			{
				k=(j-key)%26;
				if(k<0)
					k=k+26;
				npt[i]=alpha[k];
			}
		}
	}
	printf("Decrypted Text is %s",npt);
}
```

Output:

```
enter the plain text
horse
enter the key
3
Encryption is as follows
Cipher Text is kruvh
Decryption is as follows
Decrypted Text is horse
```

b) Substitution Cipher

PROGRAM:

```
#include<stdio.h>
#include<string.h>
void main()
{
	int len,i,j,key;
	char alpha[]="abcdefghijklmnopqrstuvwxyz";
	char keyis[]="defghijklmnopqrstuvwxyzabc";
	char pt[10],ct[10],npt[10];
	printf("enter the plain text\n");
	scanf("%s",pt);
	printf("Encryption is as follows\n");
	len=strlen(pt);
	for(i=0;i<len;i++)
	{
		for(j=0;j<26;j++)
		{
			if(pt[i]==alpha[j])
			{
				ct[i]=keyis[j];
			}
		}
	}
	printf("Cipher Text is %s\n",ct);
	printf("Decryption is as follows\n");
	for(i=0;i<len;i++)
	{
		for(j=0;j<26;j++)
		{
			if(ct[i]==keyis[j])
			{
				npt[i]=alpha[j];
			}
		}
	}
```

```
printf("Decrypted Text is %s",npt);
}
```

Output:

```
enter the plain text
horse
Encryption is as follows
Cipher Text is kruvh
Decryption is as follows
Decrypted Text is horse
```

c)Hill Cipher

```
class HillCipher
{

// Following function generates the
// key matrix for the key string
static void getKeyMatrix(String key, int keyMatrix[][])
{
   int k = 0;
   for (int i = 0; i < 3; i++)
   {
      for (int j = 0; j < 3; j++)
      {
         keyMatrix[i][j] = (key.charAt(k)) % 65;
         k++;
      }
   }
}

// Following function encrypts the message
static void encrypt(int cipherMatrix[][], int keyMatrix[][],int messageVector[][])
{
   int x, i, j;
   for (i = 0; i < 3; i++)
   {
      for (j = 0; j < 1; j++)
      {
         cipherMatrix[i][j] = 0;

         for (x = 0; x < 3; x++)
         {
            cipherMatrix[i][j] +=keyMatrix[i][x] * messageVector[x][j];
         }

         cipherMatrix[i][j] = cipherMatrix[i][j] % 26;
      }
   }
}
```

```
// Function to implement Hill Cipher
static void HillCipher(String message, String key)
{
    // Get key matrix from the key string
    int [][]keyMatrix = new int[3][3];
    getKeyMatrix(key, keyMatrix);

    int [][]messageVector = new int[3][1];

    // Generate vector for the message
    for (int i = 0; i < 3; i++)
        messageVector[i][0] = (message.charAt(i)) % 65;

    int [][]cipherMatrix = new int[3][1];

    // Following function generates
    // the encrypted vector
    encrypt(cipherMatrix, keyMatrix, messageVector);

    String CipherText="";

    // Generate the encrypted text from
    // the encrypted vector
    for (int i = 0; i < 3; i++)
        CipherText += (char)(cipherMatrix[i][0] + 65);

    // Finally print the ciphertext
    System.out.print(" Ciphertext:" + CipherText);
}

// Driver code
public static void main(String[] args)
{
    // Get the message to be encrypted
    String message = "ACT";

    // Get the key
    String key = "GYBNQKURP";

    HillCipher(message, key);
    }
}
```

Output:
Ciphertext: POH

Week 4

PROGRAM:

Write a Java program to implement the RSA algorithm logic.

```
 import java.math.*;
 import java.util.*;
public class RSA
 {
        public static int getGCD(int mod, int num)
        {
                if (mod == 0)
                        return num;
                else
                        return getGCD(num % mod, mod);
        }
        public static void main(String args[])
        {
                 int d = 0, e;
                 int message = 32;
                 int prime1 = 5;
                 int prime2 = 7;
                int n = prime1 * prime2;
                int etf = (prime1 - 1) * (prime2 - 1);
                System.out.println("primeMul1 is equal to : " + etf + "\n");
                for (e = 2; e < etf; e++)
                {
                         if (getGCD(e, etf) == 1)
                        {
                                break;
                        }
                }
                System.out.println("Public key e is = " + e);
                 // Calculating the private key
                 for (int m = 0; m <= 9; m++)
                {
                 int temp = 1 + (m * etf);
                if (temp % e == 0)
                {
```

```
d = temp / e;
break;
} }
System.out.println("d is : " + d);
double cipher;
BigInteger d_message; // getting the cipher text
cipher = (Math.pow(message, e)) % n;
System.out.println("Cipher text is : " + cipher); // Int to BigInteger
BigInteger bigN = BigInteger.valueOf(n); // Float to bigINt
BigInteger bigC = BigDecimal.valueOf(cipher).toBigInteger(); // decrypting the msg
d_message = (bigC.pow(d)).mod(bigN); // print decrypted message
System.out.println("Decrypted text is : " + d_message); } }
```

Output:

Public key e is 3

d is 107

Cipher text is : 44.0

Decrypted text is: 32

Week 5

Write a C/JAVA program to implement the Diffie-Hellman Key Exchange algorithm.

PROGRAM:

```
public class DHK {
   // Power function to return value of a ^ b mod P
   private static long power(long a, long b, long p)
   {
      if (b == 1)
         return a;
      else
         return (((long)Math.pow(a, b)) % p);
   }

   // Driver code
   public static void main(String[] args)
   {
      long q,a,xa,xb,ya,yb,ka, kb;
      // Both the persons will be agreed upon the
      // Global public elements G and P
      // A prime number q is taken
      q = 23;
      System.out.println("The value of q:" + q);
      // A primitive root for q, a is taken
      a = 9;
      System.out.println("The value of a:" + a);
      // Alice will choose the private key xa
      // xa is the chosen private key
      xa = 4;
      System.out.println("The private key a for Alice:" + xa);
      // Generate the public key by Alice
      ya = power(a, xa, q);
      System.out.println("The public  key of Alice:"+ya);
      // Bob will choose the private key xb
      // xb is the chosen private key
      xb = 3;
      System.out.println("The private key b for Bob:"+ xb);
```

```
  // Generate the public key by Bob
   yb = power(a, xb, q);
  System.out.println("The public  key of Alice:"+yb);

 // Generating the secret key after the exchange  of keys
        ka = power(yb, xa, q); // Secret key for Alice
        kb = power(ya, xb, q); // Secret key for Bob
        System.out.println("Secret key for the Alice is:" + ka);
        System.out.println("Secret key for the Bob is:" + kb);
    }
  }
```

Output:

The value of q:23

The value of a:9

The private key a for Alice:4

The public key of Alice:

The private key b for Bob:3

The public key of Alice:

Secret key for the Alice is:

Secret key for the Bob is:

Week 6

Calculate the message digest of a text using the SHA-1 algorithm in JAVA.

PROGRAM:

```
import java.security.*;

public class SHA1 {

public static void main(String[] a)
 {
try

{

MessageDigest md = MessageDigest.getInstance("SHA1");

System.out.println("Message digest object info: ");

 System.out.println(" Algorithm = " +md.getAlgorithm());

System.out.println(" Provider = " +md.getProvider());

System.out.println(" ToString = " +md.toString());

String input = "";

md.update(input.getBytes());

byte[] output = md.digest();

System.out.println();

System.out.println("SHA1(\""+input+"\") = " +bytesToHex(output));

 input = "abc";

md.update(input.getBytes());

output = md.digest();

System.out.println();

System.out.println("SHA1(\""+input+"\") = " +bytesToHex(output));

input = "abcdefghijklmnopqrstuvwxyz";

md.update(input.getBytes());

output = md.digest();

 System.out.println();
```

```
System.out.println("SHA1(\"" +input+"\") = " +bytesToHex(output));
System.out.println();
}
 catch (Exception e) {
System.out.println("Exception:  " +e);
}
}

public static String bytesToHex(byte[] b) {
char hexDigit[] = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F'};
StringBufferbuf=new
StringBuffer();for (int j=0;
j<b.length;j++)
{ buf.append(hexDigit[(b[j] >> 4) &
0x0f]);buf.append(hexDigit[b[j] &
0x0f]);
}
returnbuf.toString(); }
}
```

OUTPUT:

Message digest object info: Algorithm = SHA1 Provider = SUN version 1.6ToString = SHA1 Message Digest from SUN, <initialized> SHA1("") = DA39A3EE5E6B4B0D3255BFEF95601890AFD80709 SHA1("abc") = A9993E364706816ABA3E25717850C26C9CD0D89D SHA1("abcdefghijklmnopqrstuvwxyz")=32D10C7B8CF96570CA04CE37F2A19D8424 0D3A89

Week 7:

Calculate the message digest of a text using the MD5 algorithm in JAVA.

PROGRAM:

```
import java.security.*;

public class MD5 {
public static void main(String[] a) {
// TODO code application logic heretry {
MessageDigest md = MessageDigest.getInstance("MD5");
System.out.println("Message digest object info: "); System.out.println("
Algorithm = " +md.getAlgorithm()); System.out.println(" Provider = "
+md.getProvider()); System.out.println(" ToString = " +md.toString());
String input = ""; md.update(input.getBytes()); byte[] output
= md.digest(); System.out.println();
System.out.println("MD5(\""+input+"\") = " +bytesToHex(output));

input = "abc"; md.update(input.getBytes());

output = md.digest();

System.out.println();
System.out.println("MD5(\""+input+"\") = " +bytesToHex(output));

input = "abcdefghijklmnopqrstuvwxyz";

md.update(input.getBytes());

output = md.digest();
System.out.println();
System.out.println("MD5(\"" +input+"\") = "+bytesTo Hex(output));
System.out.println("");

}
    catch (Exception e)
    { System.out.println("Exception: " +e); }
    }
    public static String bytesToHex(byte[] b) {
    char hexDigit[] = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F'};
    StringBufferbuf = new StringBuffer(); for (int j=0; j<b.length;j++)
```

```
{ buf.append(hexDigit[(b[j] >> 4) & 0x0f]);
buf.append(hexDigit[b[j] &0x0f]); } return buf.toString(); } }
```

OUTPUT:

```
Message digest object info:
Algorithm = MD5
Provider = SUNversion 1.6
ToString=MD5 MessageDigest from SUN,<initialized>
MD5("")= D41D8CD98F00B204E9800998ECF8427E
MD5("abc") =900150983CD24FB0D6963F7D28E17F72
MD5("abcdefghijklmnopqrstuvwxyz")= C3FCD3D76192E4007DFB496CCA67E13B
```

Week 8

Using Java Cryptography, encrypt the text“Hello world” using BlowFish. Create your own key using Java keytool.

PROGRAM:

```
import javax.crypto.Cipher;

import javax.crypto.KeyGenerator;

import javax.crypto.SecretKey;

importjavax.swing.JOptionPane;

public class BlowFishCipher {
   public static void main(String[] args) throws Exception {
   // create a key generator based upon the Blowfish cipher

   KeyGenerator keygenerator =KeyGenerator.getInstance("Blowfish");
   // create a key
   // create a cipher based upon Blowfish
   Cipher cipher= Cipher.getInstance("Blowfish");
   // initialise cipher to with secret key
    cipher.init(Cipher.ENCRYPT_MODE, secretkey);
   // get the text to encrypt
   String inputText = JOptionPane.showInputDialog("Input your message: "); // encrypt
   messagebyte[] encrypted = cipher.doFinal(inputText.getBytes());
   //re-initialisetheciphertobeindecryptmode
   cipher.init(Cipher.DECRYPT_MODE, secretkey);
   // decrypt message
   byte[] decrypted = cipher.doFinal(encrypted);
   // and display the results
   JOptionPane.showMessageDialog(JOptionPane.getRootFrame(), "\nEncrypted text:"+ new
   String(encrypted)+"\n"+"\nDecryptedtext:"+ new String(decrypted));
   System.exit(0);
   } }
```

OUTPUT:

Input your message: Helloworld Encrypted text: 3ooo&&(*&*4r4 Decrypted text: Hello wor

Week 9:
Write a Java program to implement the DES algorithm logic.

PROGRAM:

```
import java.util.*;
import java.io.BufferedReader; import
java.io.InputStreamReader; import
java.security.spec.KeySpec; import
javax.crypto.Cipher;
import  javax.crypto.SecretKey;
import javax.crypto.SecretKeyFactory; import
javax.crypto.spec.DESedeKeySpec; import
sun.misc.BASE64Decoder;
import sun.misc.BASE64Encoder; public
class DES{
private static final String UNICODE_FORMAT = "UTF8";
public static final String DESEDE_ENCRYPTION_SCHEME = "DESede";
privateKeySpecmyKeySpec; privateSecretKeyFactorymySecretKeyFactory; private
Cipher cipher;
byte[] keyAsBytes;
private String myEncryptionKey; private String
myEncryptionScheme; SecretKey key;
static BufferedReader br = new BufferedReader(new
InputStreamReader(System.in)); public DES() throws Exception {
        // TODO code application logic here myEncryptionKey
= "ThisIsSecretEncryptionKey"; myEncryptionScheme =
DESEDE_ENCRYPTION_SCHEME; keyAsBytes =
myEncryptionKey.getBytes(UNICODE_FORMAT);  myKeySpec
```

```
= new DESedeKeySpec(keyAsBytes);
        mySecretKeyFactory = SecretKeyFactory.getInstance(myEncryptionScheme); cipher
        = Cipher.getInstance(myEncryptionScheme);
key = mySecretKeyFactory.generateSecret(myKeySpec);

            }
public String encrypt(String unencryptedString)
                { String encryptedString = null;
try {
cipher.init(Cipher.ENCRYPT_MODE, key);
        byte[] plainText = unencryptedString.getBytes(UNICODE_FORMAT); byte[]
        encryptedText = cipher.doFinal(plainText);
                    BASE64Encoder base64encoder = new BASE64Encoder(); encryptedString
= base64encoder.encode(encryptedText); } catch
(Exception e)
{ e.printStackTrace(); }
returnencryptedString; }
public String decrypt(String encryptedString)
                { String decryptedText=null;
try {
cipher.init(Cipher.DECRYPT_MODE, key);
                    BASE64Decoder base64decoder = new BASE64Decoder(); byte[]
        encryptedText = base64decoder.decodeBuffer(encryptedString); byte[] plainText
        = cipher.doFinal(encryptedText); decryptedText= bytes2String(plainText);
        }
catch (Exception e)
{ e.printStackTrace(); }
returndecryptedText; }
private static String bytes2String(byte[] bytes)
{ StringBufferstringBuffer = new StringBuffer(); for (int i =
0; i <bytes.length;
```

```
i++) { stringBuffer.append((char) bytes[i]); }
returnstringBuffer.toString(); }
public static void main(String args []) throws Exception
    { System.out.print("Enter the string: "); DES
        myEncryptor= new DES();
        String stringToEncrypt = br.readLine();
        String encrypted = myEncryptor.encrypt(stringToEncrypt); String
        decrypted = myEncryptor.decrypt(encrypted);
        System.out.println("\nString To Encrypt: " +stringToEncrypt);
        System.out.println("\nEncrypted Value : " +encrypted);
        System.out.println("\nDecrypted Value : " +decrypted);
        System.out.println("");
    }
}
```

OUTPUT:

Enter the string: Welcome

String To Encrypt: Welcome

Encrypted Value : BPQMwc0wKvg

Decrypted Value : Welcome

Week 10:

Write a C/JAVA program to implement the BlowFish algorithm logic.

PROGRAM:

```
import java.io.*;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.security.Key;
import javax.crypto.Cipher;
import javax.crypto.CipherOutputStream;
import javax.crypto.KeyGenerator;
import sun.misc.BASE64Encoder;
public class BlowFish{
public static void main(String[] args) throws Exception {
        // TODO code application logic here
KeyGeneratorkeyGenerator= KeyGenerator.getInstance("Blowfish"); keyGenerator.init(128);
Key secretKey = keyGenerator.generateKey();
Cipher cipherOut = Cipher.getInstance("Blowfish/CFB/NoPadding");
cipherOut.init(Cipher.ENCRYPT_MODE, secretKey);
BASE64Encoder encoder = new BASE64Encoder();
    byte iv[] = cipherOut.getIV();
    if (iv != null) {
    System.out.println("Initialization Vector of the Cipher: " + encoder.encode(iv));                    }
FileInputStream fin = new FileInputStream("inputFile.txt");
FileOutputStreamfout = new FileOutputStream("outputFile.txt"); CipherOutputStreamcout = new
CipherOutputStream(fout, cipherOut);
int input= 0;
while ((input = fin.read()) != -1)
{ cout.write(input); }
fin.close(); cout.close();              } }
```

OUTPUT:

Initialization Vector of the Cipher:

dI1MXzW97oQ= Contents of inputFile.txt:

Hello World

Contents of outputFile.txt: ùJÖ˜ NåI“

Week 11:

Write a C/JAVA program to implement the Rijndael algorithm logic.

PROGRAM:

```
import java.security.*; import
javax.crypto.*; import
javax.crypto.spec.*; import
java.io.*;
public class AES {
public static String asHex (byte buf[]) { StringBuffer strbuf =
new StringBuffer(buf.length * 2); int i;
for (i = 0; i < buf.length; i++) {if
(((int) buf[i] & 0xff) < 0x10)
strbuf.append("0");
strbuf.append(Long.toString((int) buf[i] & 0xff, 16)); } return
strbuf.toString(); }
public static void main(String[] args) throws Exception
{ String message="AES still rocks!!";
// Get the KeyGenerator
KeyGenerator kgen = KeyGenerator.getInstance("AES");
kgen.init(128); // 192 and 256 bits may not be available
// Generate the secret key specs.
SecretKey skey =kgen.generateKey();
byte[] raw= skey.getEncoded();
SecretKeySpec skeySpec = new SecretKeySpec(raw, "AES");
// Instantiate the cipher
Cipher cipher = Cipher.getInstance("AES");
cipher.init(Cipher.ENCRYPT_MODE, skeySpec);
byte[] encrypted = cipher.doFinal((args.length == 0 ? message :
```

```
        args[0]).getBytes()); System.out.println("encrypted string: " +
        asHex(encrypted)); cipher.init(Cipher.DECRYPT_MODE, skeySpec); byte[]
        original = cipher.doFinal(encrypted);
        String originalString = new String(original);
    System.out.println("Original string: " + originalString + " " + asHex(original));
    }
     }
```

OUTPUT:

Input your message: HelloSSIT

Encrypted text: 3ooo&&(*&

Decrypted text: HelloSSIT