Estd.2001

Sri Indu
College of Engineering & Technology
UGC Autonomous Institution
Recognized under 2(f) & 12(B) of UGC Act 1956,
NAAC, Approved by AICTE &
Permanently Affiliated to JNTUH

NAAC
NATIONAL ASSESSMENT AND
ACCREDITATION COUNCIL

LAB MANUAL

Fourth Year CSE- Semester I

CRYPTOGRAPHY &NETWORK SECURITYLAB

R18CSE41L1

DEPARTMENT OF COMPUTERSCIENCE
AND ENGINEERING

ACADEMIC YEAR 2022-23

SRI INDU COLLEGE OF ENGINEERING &TECHNOLOGY

(An Autonomous Institution under UGC, New Delhi)
B. TECH COMPUTER SCIENCE AND ENGINEERING

INSTITUTION VISION

To be a premier Institution in Engineering & Technology and Management with competence, valuesand social consciousness.

INSTITUTION MISSION

IM1: Provide high quality academic programs, training activities and research facilities.

IM2: Promote continuous industry-institute interaction for employability, entrepreneurship, Leadership and research aptitude among stakeholders.

IM3: Contribute to the economic and technological development of the region, state and Nation.

DEPARTMENT VISION

To be a technologically adaptive center for computing by grooming the students as top notchprofessionals.

DEPARTMENT MISSION

DM1: To offer quality education in computing.

DM2: To provide an environment that enables overall development of all the stakeholders.

DM3: To impart training on emerging technologies.

DM4: To encourage participation of stakeholders in research and development.

Program Educational Object ves (PEO's)

PEO-1 Higher Studies: Graduate with an ability to pursue higher studies and get employment in reputed institutions and organizations.

PEO-2 Domain Knowledge: Graduate with an ability to design and develop a product.

PEO-3 Professional Career: Graduate with excellence by multidisciplinary approach to achievesuccessful Professional career.

PEO-4 Life Long Learning: Graduate with an ability to learn advanced skills to face professional competence through lifelong learning.

Program Specific Outcomes (PSO's)

PSO1 To Develop software projects using standard practices and suitable programmingenvironment.

PSO2 To identify, formulate and solve the real life problems faced in the society, industry and other areasby applying the skills of the programming languages, networks and databases learned.

PSO3 To apply computer science knowledge in exploring and adopting latest technologies invarious inter-Disciplinary research activities.

Program Outcomes (PO's)

PO1	Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	Problem Analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	Design / Development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	Modern tool usage: Create, select, and apply appropriate techniques, resources, andmodern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6	The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	Ethics: Apply ethical principles and commit to professional ethics and responsibilitiesand norms of the engineering practice.
PO9	Individual and team work: Function effectively as an individual, and as a member orleader in diverse teams, and in multidisciplinary settings.
PO10	Communication: Communicate effectively on complex engineering activities with theengineering community and with society at large, such as, being able to comprehendand write effective reports and design documentation, make effective presentations,and give and Receive clear instructions.
PO11	Project management and finance: Demonstrate knowledge and understanding of theengineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	Life-long learning Recognize the need for, and have the preparation and ability toengage in independent and life-long learning in the broadest context of technological change.

SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY
(An Autonomous Institution under UGC, New Delhi)
Recognized under 2(f) and 12(B) of UGC Act 1956
NBA & NAAC Accredited, Approved by AICTE and Permanently affiliated to JNTUH
Sheriguda (V), Ibrahimpatnam, R.R.Dist, Hyderabad - 501 510

D4

BR-18

Lr.No.SICET/AUTO/DAE/IV B.Tech Acade.:nic Calendar/306/2022 **Dt: 01.08.2022**

Dr.G. SURESH,
Principal,

Recd

To,
All the HODs.

IV B.TECH I SEM & II SEM ACADEMIC CALENDAR
ACADEMIC YEAR : 2022-23

Sir,

Sub: SICET (Autonomous) - Academic & Evaluation - Academic Calendar for **B.Tech – 4th Year** - For the academic year **2022-23** – Reg.

***

The approved Academic Calendar for **B.Tech – 4th Year (I & II Sem)** for the academic year **2022-23** is given below:

Academic Calendar for B.Tech – 4th Year Students (2019 - 20 Batch), BR-18 Regulation.

I - Semester

Commencement of I Semester class work	**25.08.2022 (Thursday)**	
I Spell of Instructions (Including Dussehra Holidays).	25.08.2022	26.10.2022 - 9 Weeks
Dussehra Holidays.	03.10.2022	08.10.2022 - 1 Week
I Mid Term Examinations for IV B.Tech I Sem Students.	27.10.2022	29.10.2022 - 3 Days
II Spell of Instructions.	31.10.2022	24.12.2022 - 8 Weeks
II Mid Term Examinations for IV B.Tech I Sem Students.	27.12.2022	29.12.2022 - 3 Days
Preparation Holidays, Practical Examinations and Remedial Mid Test (RMT).	30.12.2022	07.01.2023 - 9 Days
IV B.Tech I Semester End Examinations (Main) and Supplementary Examinations.	09.01.2023	25.01.2023 - 2 Weeks 3 Days
Sankranti Holidays.	13.01.2022	16.01.2022 - 4 Days
Commencement of class work of IV B.Tech II Semester - 27.01.2023 (Friday)		

II - Semester

Commencement of II Semester class work	**27.01.2023 (Friday)**	
I Spell of Instructions.	27.01.2023	23.03.2023 - 8 Weeks
I Mid Examinations for IV B.Tech II Sem Students.	24.03.2023	25.03.2023 - 2 Days
II Spell of Instructions.	27.03.2023	20.05.2023 - 8 Weeks
II Mid Examinations for IV B.Tech II Sem Students.	22.05.2023	23.05.2023 - 2 Days
Preparation Holidays, Project Evaluation and Remedial Mid Test (RMT).	24.05.2023	31.05.2023 - 8 Days
IV B.Tech II Semester End Examinations (Main) and Supplementary Examinations.	01.06.2023	07.06.2023 - 1 Week

ACE CE DIRECTOR PRINCIPAL

Copy to DAE,
Copy to all the Heads of the Depts.

CONTROLLER OF EXAMINATIONS
Sri Indu College of Engineering & Technology
(An Autonomous Institution under JNTUH)
Sheriguda (V), Ibrahimpatnam, R.R. Dist-501510.

DIRECTOR
(Academic Audit)
Sri Indu College of Engineering & Technology
Sheriguda, IBP, R.R. Dist-501510.

PRINCIPAL
Sri Indu College of Engineering & Technology
(An Autonomous Institution Under JNTUH)
Sheriguda (V), Ibrahimpatnam, R.R. Dist-501510

CRYPTOGRAPHY &NETWORK SECURITYLAB

R18CSE41L1

COURSE OUTCOMES (CO's)

Academic Year: 2022-23

Class: IV YEAR-I SEM**.**

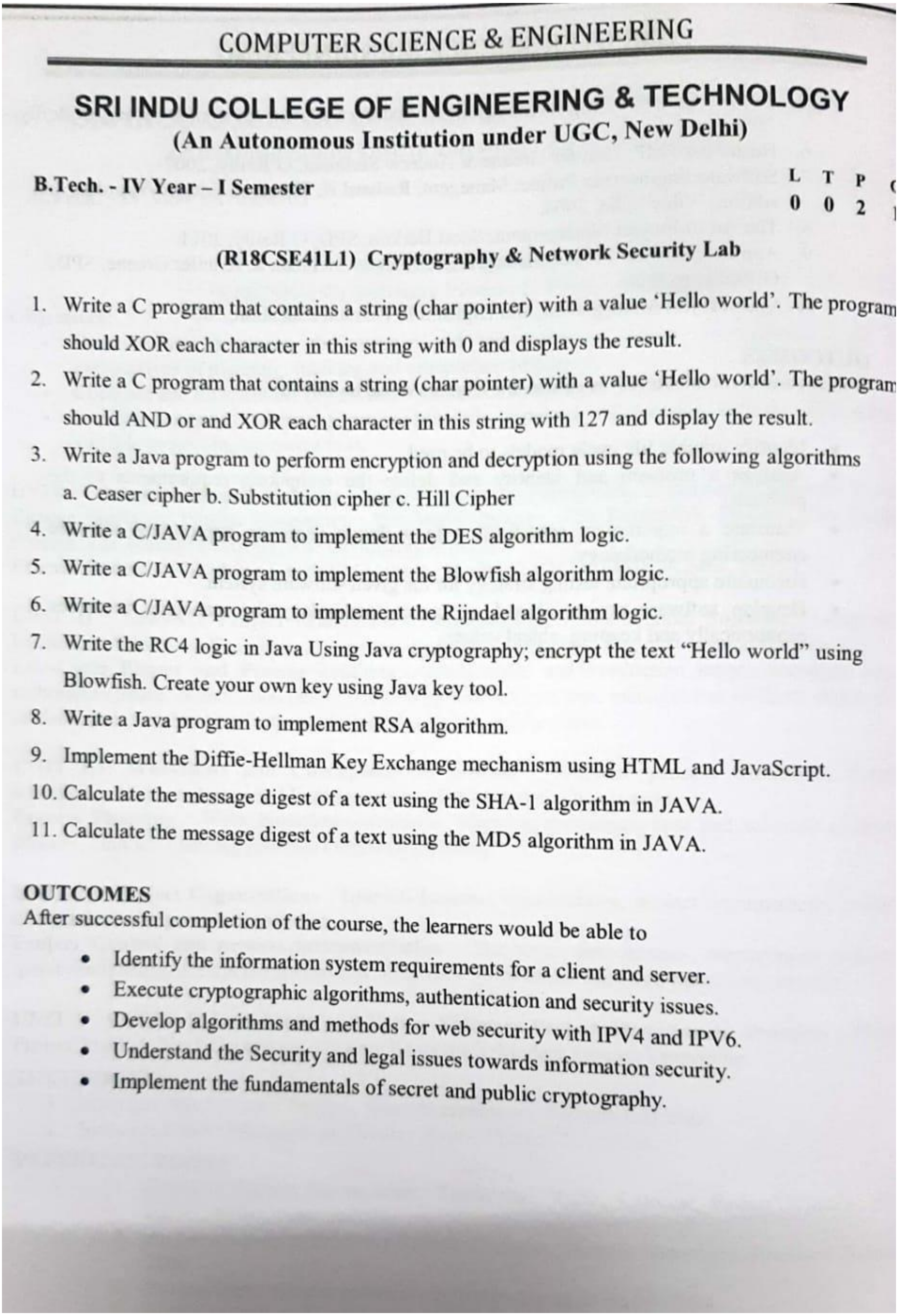
Course Name: CRYPTOGRAPHY &NETWORK SECURITY LAB (R18CSE41L1)

At the end of the course, the student will be able to

Course Outcomes (COs)	
C41L1.1	Explain security concepts, Ethics in Network Security. Identify and classify various Attacks and explain the same.
C41L1.2	Compare and contrast symmetric and asymmetric encryption systems and their vulnerability to various attacks.
C41L1.3	Explain the role of third-party agents in the provision of authentication services.
C41L1.4	Comprehend and apply authentication, email security, web security services and mechanisms.
C41L1.5	Distinguish and explain different protocol like SSL, TLS Vis-à-vis their applications
C41L1.6	Discuss the effectiveness of passwords in access control. Explain firewall principles.

Mapping of Course Outcomes(CO's) with PO's:

CO	PO 1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C41L1.1	3	2	-	-	-	-	-	-	-	-	-	-	3	3	3
C41L1.2	3	3	3	-	-	-	-	-	-	-	-	-	3	3	3
C41L1.3	2	3	3	3	-	-	-	-	-	-	-	-	3	3	3
C41L1.4	2	3	2	2	-	-	-	-	-	-	-	-	3	3	3
C41L1.5	2	3	3	2	-	-	-	-	-	-	-	-	3	3	3
C41L1.6	2	2	3	3	-	-	-	-	-	-	-	-	3	3	3
	2.33	**2.67**	**2.8**	**2.5**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	3	3	3



SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY

(An Autonomous Institution under UGC, New Delhi)

B.Tech. - IV Year – I Semester

L	T	P	C
0	0	2	1

(R18CSE41L1) Cryptography & Network Security Lab

1. Write a C program that contains a string (char pointer) with a value 'Hello world'. The program should XOR each character in this string with 0 and displays 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 Java program to perform encryption and decryption using the following algorithms a. Ceaser cipher b. Substitution cipher c. Hill Cipher
4. Write a C/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 the 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.

OUTCOMES

After successful completion of the course, the learners would be able to

- Identify the information system requirements for a client and server.
- Execute cryptographic algorithms, authentication and security issues.
- Develop algorithms and methods for web security with IPV4 and IPV6.
- Understand the Security and legal issues towards information security.
- Implement the fundamentals of secret and public cryptography.

SRI INDU COLLEGE OF ENGG & TECH LESSON PLAN of LAB (Regulation :R18) Department of COMPUTER SCIENCE AND ENGINEERING			Prepare d on Rev1: **Page: 1 of 2**
Sub. Code & Title	(R18CSE41L1) CRYPTOGRAPHY & NETWORK SECURITY LAB		
Academic Year: 2022-23	Year/Sem./Section	IV-I A,B,C,D	
Faculty Name & Designation	RAMAVATH VINOD KUMAR/P.HYMAVATHI Assistant Professor		

Lab Plan

2022-23 IV Year –I Semester CSE

S No	Topics	No. of weeks
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.	1
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.	1
3.	Write a Java program to perform encryption and decryption using the following algorithms: a. Ceaser Cipher b. Substitution Cipher c. Hill Cipher	1
4.	Write a Java program to implement the DES algorithm logic	1
5	Write a C/JAVA program to implement the Blowfish algorithm logic	1
6	Write a C/JAVA program to implement the Rijndael algorithm logic	1

7	1) Write the RC4 logic in Java Using Java Cryptography, encrypt the text "Hello world" using Blowfish. Create your own key using Java key tool. 2) Write a Java program to implement RSA Algorithm	1
8	1. Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript. 2. Calculate the message digest of a text using the SHA-1 algorithm in JAVA.	1
9	Calculate the message digest of a text using the MD5 algorithm in JAVA.	1

CONTENT BEYOND THE SYLLABUS

S.No	Topics	No of weeks
10	Write a java program to implement Diffie Hellman Key Exchange	1
11	Write a java program to implement triple DES	1
12	Write a java program for Knapsack using Dynamic Programming based solution	1

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 Java program to perform encryption and decryption using the following algorithms:
   1. Ceaser Cipher
   2. Substitution Cipher
   3. 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(charpointer) witha value\Hello World’.The program should XOR each character in this string with 0 and display the result.

PROGRAM:

```
#include<stdlib.h>
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 Hello World

Week 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.

PROGRAM:

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

Output:

OUTPUT :Hello
World

```
Hello     World

Hello World
```

Week 3

Write a Javaprogramtoperformencryptionanddecryption usingthe following algorithms:

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

PROGRAM:

a) Ceaser Cipher

```
import java.io.BufferedReader; import java.io.IOException; import java.io.InputStreamReader; import
java.util.Scanner;
public class CeaserCipher {

static Scanner sc=new Scanner(System.in);
staticBufferedReaderbr=newBufferedReader(newInputStreamReader(System.in));public static void
main(String[] args) throws IOException {
// TODO code application logic here

System.out.print("Enter any String: "); String str = br.readLine();
System.out.print("\nEnter the Key: ");
int key = sc.nextInt();

String encrypted = encrypt(str, key);
System.out.println("\nEncrypted String is: " +encrypted);
Stringdecrypted=decrypt(encrypted, key); System.out.println("\nDecrypted String is: "+decrypted);
System.out.println("\n");
}

public static String encrypt(String str, int key)


{
String encrypted ="";
for(int i = 0; i < str.length(); i++)
{
```

```
int c= str.charAt(i);
if  (Character.isUpperCase(c))
{
c = c + (key % 26);
if (c > 'Z')
                c = c - 26;
}

else if (Character.isLowerCase(c)) {
c = c + (key % 26);
if (c > 'z')
               c = c - 26;
}

encrypted += (char) c;
}
return encrypted;
}

public static String decrypt(String str, int key)
{
 String decrypted = "";
 for(int i= 0; i< str.length(); i++)
 {
int c= str.charAt(i);
if(Character.isUpperCase(c))
{
c = c - (key % 26);
if (c < 'A')
                c = c + 26;
}

else if (Character.isLowerCase(c))
{
c = c - (key % 26);
if (c < 'a')
}
                c = c + 26;
                }
```

Output:

```
Enterany String: HelloWorld Enter the Key: 5
Encrypted String is: MjqqtBtwqi Decrypted String is: HelloWor
```

b) Substitution Cipher

PROGRAM:

```
import java.io.*;
import java.util.*;
public class SubstitutionCipher
{
static Scanner sc = new Scanner(System.in);
staticBufferedReaderbr=newBufferedReader(newInputStreamReader(System.in))
;public static void main(String[] args) throws IOException
 {
// TODO code application logic here
 String a
String a= "abcdefghijklmnopqrstuvwxyz";
String b = "zyxwvutsrqponmlkjihgfedcba";
System.out.print("Enter any string: ");
String str = br.readLine();
String decrypt = "";
 char c;
for(int i=0;i<str.length();i++)
{
c=str.charAt(i);
int j = a.indexOf(c);
decrypt = decrypt+b.charAt(j);
}
System.out.println("The encrypted data is: " +decrypt);
}

}
```

Output:

```
Enter any string: aceho

The encrypted data is: zxvsl
```

c)HillCipher

PROGRAM:

```
import java.io.*;
 import  java.util.*;
import java.io.*; public class HillCipher {
staticfloat[][] decrypt= newfloat[3][1];
staticfloat[][] a= newfloat[3][3];
static float[][]b=newfloat[3][3];
staticfloat[][] mes=newfloat[3][1];
staticfloat[][]res= new float[3][1];
static BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
static Scanner sc = new Scanner(System.in);
public static void main(String[] args) throws IOException {
//TODOcode applicationlogic here getkeymes();
for(int i=0;i<3;i++)
for(int j=0;j<1;j++)
(int k=0;k<3;k++) {
res[i][j]=res[i][j]+a[i][k]*mes[k][j];
}
System.out.print("\nEncrypted string is : ");
for(int i=0;i<3;i++)
{ System.out.print((char)(res[i][0]%26+97));
[i][0]=res[i][0];


}
inverse();
for(int i=0;i<3;i++)
for(int j=0;j<1;j++)
for(int k=0;k<3;k++)
 {
decrypt[i][j] = decrypt[i][j]+b[i][k]*res[k][j]; } System.out.print("\nDecrypted string is : ");
```

```
for(inti =0;i<3;i++)
{
System.out.print((char)(decrypt[i][0]% 26+97));
}
System.out.print("\n");
}
public static void getkeymes() throws IOException
 {
System.out.println("Enter 3x3 matrix for key (It should be inversible): ");
for(int i=0;i<3;i++)
for(int j=0;j<3;j++)
a[i][j]= sc.nextFloat();
System.out.print("\nEnter a 3 letter string: ");
String msg = br.readLine();
for(int i=0;i<3;i++) mes[i][0]
= msg.charAt(i)-97;
}
public static void inverse()
{
 floatp,q;
float[][] c= a;
for(int i=0;i<3;i++)
for(int j=0;j<3;j++) {
//a[i][j]=sc.nextFloat();
if(i==j)
b[i][j]=1;
else
b[i][j]=0;
}
for(int k=0;k<3;k++)
{
for(int  i=0;i<3;i++)
 {
```

```
p = c[i][k];
q = c[k][k];
 for(int j=0;j<3;j++)
 {
if(i!=k)

c[i][j] = c[i][j]*q-p*c[k][j];
b[i][j]  =b[i][j]*q-p*b[k][j];
} } } }
for(int i=0;i<3;i++) for(int j=0;j<3;j++)
{ b[i][j] = b[i][j]/c[i][i]; }

System.out.println(""); System.out.println("\nInverse Matrix is : "); for(int i=0;i<3;i++) {
for(int j=0;j<3;j++) System.out.print(b[i][j] + " ");
System.out.print("\n");  }
} }
```

Output:

```
Entera3letterstring:hai Encrypted string is :fdx Inverse Matrix is:
0.083333336 0.41666666 -0.33333334
-0.41666666  -0.083333336  0.6666667
0.5833333 -0.083333336   -0.33333334
Decrypted string is: hai
```

Week 4

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;
 key;
static    BufferedReader    br    =    new    BufferedReader(new
InputStreamReader(System.in));
publicDES()throws Exception{
                // TODO code application logic here my
```

```
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:

```
Enterthestring:WelcomeString To
Encrypt: Welcome
Encrypted Value : BPQMwc0wKvg= Decrypted
Value: Welcome
```

Week 5

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); KeysecretKey =
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  Vectorofthe  Cipher:" + encoder.encode(iv));                  }
FileInputStream fin= new FileInputStream("inputFile.txt"); FileOutputStreamfout = new
FileOutputStream("outputFile.txt"); CipherOutputStreamcout = new CipherOutputStream(fout, cipherOut); intinput
= 0;
while ((input = fin.read()) != -1)
{ cout.write(input); }

fin.close(); cout.close(); } }
```

OUTPUT:

```
Initialization Vectorofthe Cipher: dI1MXzW97oQ= Contents of inputFile.txt: Hello World
Contents of outputFile.txt: ùJÖ˜ NåI“
```

Week 6

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

P**ROGRAM**:

```
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: Hello KGRCET Encrypted text: 3ooo&&(*&*4r4 Decrypted text: Hello
KGRCET
```

Week 7

UsingJavaCryptography, encryptthetext“Hello world” using BlowFish. Createyour own key using Java keytool.

PROGRAM:

```
import javax.crypto.Cipher; import javax.crypto.KeyGenerator; import javax.crypto.SecretKey; import
javax.swing.JOptionPane; public class BlowFishCipher {
public static void main(String[] args) throws Exception {
// create a key generator based upon the Blowfish cipher KeyGeneratorkeygenerator =
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 message
byte[] 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 world
```

Week 8

Write a Java program to implement RSA Algoithm.

PROGRAM:

```
importjava.io.BufferedReader;
 import java.io.InputStreamReader;
 import java.math.*;
import   java.util.Random;
 import   java.util.Scanner;
 public class RSA{
static Scanner sc = new Scanner(System.in);
public static void main(String[] args){
// TODO code application logic here
 System.out.print("Enter a Prime number: ");
BigIntegerp= sc.nextBigInteger();// Here'soneprimenumber..
System.out.print("Enter another prime number: ");
BigInteger q = sc.nextBigInteger(); // ..andanother.
BigInteger n = p.multiply(q);
BigInteger n2 = p.subtract(BigInteger.ONE).multiply(q.subtract(BigInteger.ONE));
BigInteger e
= generateE(n2);
BigInteger d = e.modInverse(n2); // Here's the multiplicative inverse

System.out.println("Encryptionkeysare:"+e+","+ n);
 System.out.println("Decryption keys are: " + d + ", " + n);
}
public static BigIntegergenerateE(BigIntegerfiofn)
{
int y, intGCD;
BigInteger e; BigInteger gcd;
Random x = new Random();
do {
```

```
y = x.nextInt(fiofn.intValue()-1);
String z = Integer.toString(y);
e= new BigInteger(z);
gcd = fiofn.gcd(e);
intGCD = gcd.intValue();
}
while(y <= 2 ||intGCD != 1); return e;
}
}
```

OUTPUT:

```
Enter a Prime number: 5
Enteranotherprimenumber:11 Encryption keys are: 33, 55
Decryption keys are: 17, 55
```

Week 9

Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript. Consider the end user as one of the parties (Alice) and the JavaScript application as other party (bob).

PROGRAM:

```
import java.math.BigInteger;
import java.security.KeyFactory;
import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.SecureRandom;
import    javax.crypto.spec.DHParameterSpec;
import    javax.crypto.spec.DHPublicKeySpec;
public class DiffeHellman{
public final static int pValue = 47;
 public final static int gValue = 71; public final static int XaValue = 9;
 publicfinalstaticint XbValue= 14;
public static void main(String[] args) throws Exception
{ // TODO code application logic here
BigInteger p = new BigInteger(Integer.toString(pValue));
 BigInteger g = new BigInteger(Integer.toString(gValue));
 BigIntegerXa = new BigInteger(Integer.toString(XaValue))
; BigIntegerXb = new BigInteger(Integer.toString(XbValue));
createKey(); intbitLength = 512; // 512 bits
SecureRandomrnd = new SecureRandom();
p = BigInteger.probablePrime(bitLength, rnd);
 g = BigInteger.probablePrime(bitLength, rnd);

createSpecificKey(p, g);
}
public static void createKey() throws Exception {
KeyPairGeneratorkpg = KeyPairGenerator.getInstance("DiffieHellman");
kpg.initialize(512);
 KeyPairkp = kpg.generateKeyPair();
KeyFactorykfactory = KeyFactory.getInstance("DiffieHellman");
DHPublicKeySpeckspec = (DHPublicKeySpec)
kfactory.getKeySpec(kp.getPublic().DHPublicKeySpec.class);
System.out.println("Public key is: " +kspec);
}
public static void createSpecificKey(BigInteger p, BigInteger g) throws Exception
{
 KeyPairGeneratorkpg = KeyPairGenerator.getInstance("DiffieHellman");
 DHParameterSpecparam = new DHParameterSpec(p, g);
 kpg.initialize(param);
KeyPairkp = kpg.generateKeyPair();
KeyFactorykfactory = KeyFactory.getInstance("DiffieHellman");
DHPublicKeySpeckspec = (DHPublicKeySpec) kfactory.getKeySpec(kp.getPublic(),
DHPublicKeySpec.class);
System.out.println("\nPublic key is : " +kspec);
}
}
```

OUTPUT:

```
Public key is: javax.crypto.spec.DHPublicKeySpec@5afd29 Public key is:
javax.crypto.spec.DHPublicKeySpec@9971a
```

Week 10

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 1.6
ToString = SHA1 Message Digest from SUN, <initialized> SHA1("") =
DA39A3EE5E6B4B0D3255BFEF95601890AFD80709 SHA1("abc") =
A9993E364706816ABA3E25717850C26C9CD0D89D
SHA1("abcdefghijklmnopqrstuvwxyz")=32D10C7B8CF96570CA04CE37F2A19D8424 0D3A89
```

Week 11

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

PROGRAM:

```
import java.security.*;
public class MD5 {
public static void main(String[] a) {
// TODO code application logic here
try {
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  =  SUN
version 1.6
ToString=MD5MessageDigestfromSUN,<initialized>MD5("")=
D41D8CD98F00B204E9800998ECF8427E  MD5("abc")  =
900150983CD24FB0D6963F7D28E17F72     MD5("abcdefghijklmnopqrstuvwxyz")
=  C3FCD3D76192E4007DFB496CCA67E13B
```

2. Write a java program to implement Diffie Hellman Key Exchange

PROGRAM

```
    class Diffie_Hellman
{
        public static void main(String args[])
        {
                Scanner sc=new Scanner(System.in);
                System.out.println("Enter modulo(p)");
                int p=sc.nextInt();
                System.out.println("Enter primitive root of "+p);
                int g=sc.nextInt();
                System.out.println("Choose 1st secret no(Alice)");
                int a=sc.nextInt();
                System.out.println("Choose 2nd secret no(BOB)");
                int b=sc.nextInt();

                int A = (int)Math.pow(g,a)%p;
                int B = (int)Math.pow(g,b)%p;

                int S_A = (int)Math.pow(B,a)%p;
                int S_B =(int)Math.pow(A,b)%p;

                if(S_A==S_B)
                {
                        System.out.println("ALice and Bob can communicate with each other!!!");
                        System.out.println("They share a secret no = "+S_A);
                }

                else
                {
                        System.out.println("ALice and Bob cannot communicate with each other!!!");
                }
        }
}
```

3. Write a java program to implement AES ALGORITHM

PROGRAM

```
 import javax.crypto.Cipher;
 import javax.crypto.SecretKey;
import javax.crypto.SecretKeyFactory;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.PBEKeySpec;
import javax.crypto.spec.SecretKeySpec;
import java.nio.charset.StandardCharsets;
import    java.security.InvalidAlgorithmParameterException;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.security.spec.InvalidKeySpecException;
import java.security.spec.KeySpec;
import java.util.Base64;
import javax.crypto.BadPaddingException;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
public class AESExample
        {
  /* Private variable declaration */
  private static final String SECRET_KEY = "123456789";
  private static final String SALTVALUE = "abcdefg";

  /* Encryption Method */
  public static String encrypt(String strToEncrypt)
  {
  try
  {
  /* Declare a byte array. */
  byte[] iv = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
  IvParameterSpec ivspec = new IvParameterSpec(iv);
  /* Create factory for secret keys. */
  SecretKeyFactory factory = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA256");
  /* PBEKeySpec class implements KeySpec interface. */
  KeySpec spec = new PBEKeySpec(SECRET_KEY.toCharArray(), SALTVALUE.getBytes(), 65536, 256);
  SecretKey tmp = factory.generateSecret(spec);
  SecretKeySpec secretKey = new SecretKeySpec(tmp.getEncoded(), "AES");
  Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
  cipher.init(Cipher.ENCRYPT_MODE, secretKey, ivspec);
  /* Retruns encrypted value. */
  return Base64.getEncoder()
  .encodeToString(cipher.doFinal(strToEncrypt.getBytes(StandardCharsets.UTF_8)));
  }
  catch (InvalidAlgorithmParameterException | InvalidKeyException | NoSuchAlgorithmException | InvalidKeySpecExcepti
  on | BadPaddingException | IllegalBlockSizeException | NoSuchPaddingException e)
  {
  System.out.println("Error occured during encryption: " + e.toString());
  }
}
return null;

/* Decryption Method */
public static String decrypt(String strToDecrypt)
{
try
{
/* Declare a byte array. */
byte[] iv = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
IvParameterSpec ivspec = new IvParameterSpec(iv);
/* Create factory for secret keys. */
SecretKeyFactory factory = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA256");
/* PBEKeySpec class implements KeySpec interface. */
KeySpec spec = new PBEKeySpec(SECRET_KEY.toCharArray(), SALTVALUE.getBytes(), 65536, 256);
SecretKey tmp = factory.generateSecret(spec);
SecretKeySpec secretKey = new SecretKeySpec(tmp.getEncoded(), "AES");
Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5PADDING");
cipher.init(Cipher.DECRYPT_MODE, secretKey, ivspec);
/* Retruns decrypted value. */
return new String(cipher.doFinal(Base64.getDecoder().decode(strToDecrypt)));
}
catch (InvalidAlgorithmParameterException | InvalidKeyException | NoSuchAlgorithmException | InvalidKeySpecExcepti
on |
BadPaddingException | IllegalBlockSizeException | NoSuchPaddingException e)
{
System.out.println("Error occured during decryption: " + e.toString());
}
return null;
}
/* Driver Code */
public static void main(String[] args)
{
/* Message to be encrypted. */
String originalval = "AES Encryption";
/* Call the encrypt() method and store result of encryption. */
String encryptedval = encrypt(originalval);
/* Call the decrypt() method and store result of decryption. */
String decryptedval = decrypt(encryptedval);
/* Display the original message, encrypted message and decrypted message on the console. */
System.out.println("Original value: " + originalval);
System.out.println("Encrypted value: " + encryptedval);
System.out.println("Decrypted value: " + decryptedval);
}
}
```

14. Write a java program for Knapsack using Dynamic Programming based solution

PROGRAM:

```
// A Dynamic Programming based solution for 0-1 Knapsack problem
class Knapsack {

    // A utility function that returns maximum of two integers
    static int max(int a, int b)
{ return (a > b) ? a : b; }

    // Returns the maximum value that can be put in a knapsack
    // of capacity W
    static int knapSack(int W, int wt[], int val[], int n)
    {
        int i, w;
        int K[][] = new int[n + 1][W + 1];

        // Build table K[][] in bottom up manner
        for (i = 0; i<= n; i++) {
            for (w = 0; w<= W; w++) {
                if (i == 0 || w == 0)
                    K[i][w] = 0;
                else if (wt[i - 1]<= w)
                    K[i][w] = max(val[i - 1] + K[i - 1][w - wt[i - 1]], K[i - 1][w]);
                else
                    K[i][w] = K[i - 1][w];
            }
        }

        return K[n][W];
    }

    // Driver program to test above function
    public static void main(String args[])
    {
        int val[] = new int[] { 60, 100, 120 };
        int wt[] = new int[] { 10,  20,  30  };
        int W = 50;
        int n = val.length;
        System.out.println(knapSack(W,  wt,  val,  n));
    }
}
```

OUTPUT:

```
220
```

SET 1

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 Java program to perform encryption and decryption using the following algorithms:

a) Ceaser Cipher

b) Substitution Cipher
c) 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

SET 2

1. Write the RC4 logic in Java Using Java Cryptography, encrypt the text “Hello world”using Blowfish. Create your own key using Java key tool.

2. Write a Java program to implement RSA Algorithm.

3. Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript.
4. Calculate the message digest of a text using the SHA-1 algorithm in JAVA.
5. Calculate the message digest of a text using the MD5 algorithm in JAVA.

```
Original value: AES SET2SE

Encrypted value: V5E9I52IxhMaW4+hJh156g==
Decrypted
```

Viva questions

1. Define Cryptography and its benefits?
2. What are the few major applications of cryptography in the modern world?
3. What is decryption? What is its need?
4. What do you mean by Secret Key Cryptography and Public Key Cryptography? How they are different from one another?
5. What type of information can be secured with Cryptography?
6. What exactly do you know about RSA?
7. What is the Digital Signature Algorithm?
8. Differentiate symmetric and asymmetric encryption?
9. What is the Caesar cipher?
10. What is plaintext?
11. What is cipher text?
12. What are the mathematical algorithms used in symmetric cryptography?
13. What are the mathematical algorithms used in asymmetric cryptography?
14. What is the difference between a private key and a public key?
15. What is a block cipher?
16. What is Transposition Ciphers?
17. What is the International Data Encryption Algorithm (IDEA)?
18. How is a Key Distribution Center (KDC) used?
19. What are the specific components of the Public Key Infrastructure (PKI)?
20. List down some Hashing Algorithms