**CONTENTS**

* 1. Syllabus
  2. Lab Cycles
  3. Instructions to Students
  4. Lab Exercises
  5. Additional Lab Programs
  6. Viva Questions

### Information Security LAB

##### Course Code: IT703PC

**List of Experiments**

##### 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.

1. **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.**

##### Write a Java program to perform encryption and decryption using the following algorithms

**a. Ceaser cipher b. Substitution cipher c. Hill Cipher**

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

1. **Write a C/JAVA program to implement the Blowfish algorithm logic.**

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

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.**

##### Write a Java program to implement RSA algorithm.

1. **Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript.**

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

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

### Department of Information Technology

##### INFORMATION SECURITY LAB

**Year: IV SEM: I**

**LAB CYCLES**

|  |  |
| --- | --- |
| **Cycle 1** | 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.   1. 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. Write a Java program to perform encryption and decryption using the following algorithms   a. Ceaser cipher b. Substitution cipher c. Hill Cipher |
| **Cycle 2** | 1. Write a C/JAVA program to implement the DES algorithm logic. 2. Write a C/JAVA program to implement the Blowfish algorithm logic. 3. Write a C/JAVA program to implement the Rijndael algorithm logic. 4. Write the RC4 logic in Java Using Java cryptography; encrypt the text “Hello world” using   Blowfish. Create your own key using Java key tool. |
| **Cycle 3** | 1. Write a Java program to implement RSA algorithm. 2. Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript. 3. Calculate the message digest of a text using the SHA-1 algorithm in JAVA. 4. Calculate the message digest of a text using the MD5 algorithm in   JAVA. |

### INSTRUCTIONS TO STUDENTS

1. After entering the lab, make an entry in the log book providing the specified details.
2. Observation book and Lab Record should be duly signed by the concerned faculty after the completion of each experiment, failing which marks will not be awarded. All experiments must be completed as per the schedule.
3. After the completion of lab, properly shutdown the system.

**GRADING:**

* The overall lab evaluation is for **100** marks. In that **75** marks for Semester End Examination and remaining **25** marks will be awarded based on **internal evaluation**.
* Internal Evaluation consists of:
  + Lab internal examination: There will be two internal examinations for 10 marks each. The final marks are awarded as the average of two internal examination marks.
  + Continuation evaluation is for 15 marks and distributed for each experiment as:

##### Experimentation 5 marks

* 1. **Observation 5 marks**
  2. **Record 5 marks**

# Lab Cycle I

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.

Aim: To be able to apply XOR operation 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

1. 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: Hello World Hello World Hello World

1. Write a Java program to perform encryption and decryption using the following algorithms
   1. 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);

static BufferedReaderbr = new BufferedReader(new InputStreamReader(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); String decrypted = 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;

}

decrypted += (char) c;

}

return decrypted;

}

}

##### Output:

Enter any String: Hello World Enter the Key: 5

Encrypted String is: MjqqtBtwqi Decrypted String is: Hello World

* 1. Substitution Cipher

import java.io.\*; import java.util.\*;

public class SubstitutionCipher {

static Scanner sc = new Scanner(System.in); static BufferedReaderbr = new

BufferedReader(newInputStreamReader(System.in)); public static void main(String[] args) throws IOException {

// TODO code application logic here 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

* 1. Hill Cipher

import java.io.\*; import java.util.\*; import java.io.\*; public class HillCipher {

static float[][] decrypt = new float[3][1]; static float[][] a = new float[3][3]; static float[][] b = new float[3][3]; static float[][] mes = new float[3][1]; static float[][] res = new float[3][1];

static BufferedReaderbr = new BufferedReader(new InputStreamReader(System.in)); static Scanner sc = new Scanner(System.in); public static void main(String[] args) throws IOException {

// TODO code application logic here getkeymes();

for(int i=0;i<3;i++) for(int j=0;j<1;j++) for(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)); res[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(int i=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:

Enter a 3 letter string: 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

# Lab Cycle II

1. Write a C/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 BufferedReaderbr = 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)

{ StringencryptedString = 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)

{ StringdecryptedText=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("\nEncryptedValue : " +encrypted);

System.out.println("\nDecryptedValue : " +decrypted); System.out.println("");

}

}

##### OUTPUT:

Enter the string: Welcome String To Encrypt: Welcome

Encrypted Value : BPQMwc0wKvg= Decrypted Value : Welcome

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

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“

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

import java.security.\*; import javax.crypto.\*; import javax.crypto.spec.\*; import java.io.\*;

public class AES {

public static String asHex (byte buf[]) { StringBufferstrbuf = 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

KeyGeneratorkgen = KeyGenerator.getInstance("AES"); kgen.init(128); // 192 and 256 bits may not be available

// Generate the secret key specs. SecretKeyskey = kgen.generateKey(); byte[] raw = skey.getEncoded();

SecretKeySpecskeySpec = 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 HAI Encrypted text: 3ooo&&(\*&\*4r4 Decrypted text: Hello HAI

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

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-initialise the cipher to be in decrypt mode 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" + "\nDecrypted text: " + new String(decrypted));

System.exit(0);

} }

##### OUTPUT:

Input your message: Hello world Encrypted text: 3ooo&&(\*&\*4r4 Decrypted text: Hello world

# Lab Cycle III

1. Write a Java program to implement RSA algorithm.

import java.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: ");

BigInteger p = sc.nextBigInteger(); // Here's one prime number..System.out.print("Enter another prime number: "); BigInteger q = sc.nextBigInteger(); // ..and another.

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("Encryption keys are: " + e + ", " + n); System.out.println("Decryption keys are: " + d + ", " + n);

}

public static BigIntegergenerateE(BigIntegerfiofn) { int y, intGCD;

BigInteger e; BigIntegergcd;

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

Enter another prime number: 11 Encryption keys are: 33, 55

Decryption keys are: 17, 55

1. Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript**.**

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; public final static int 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@9971ad

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

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")=32D10C7B8CF96570CA04CE37F2A19D84 240D3A89

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

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+"\") = "

+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]); } return buf.toString(); } }

Output:

Message digest object info:

Algorithm = MD5

Provider = SUN version 1.6

ToString = MD5 Message Digest from SUN, <initialized> MD5("") = D41D8CD98F00B204E9800998ECF8427E MD5("abc") =

900150983CD24FB0D6963F7D28E17F72 MD5("abcdefghijklmnopqrstuvwxyz")

= C3FCD3D76192E4007DFB496CCA67E13B

# Additional Lab Programs

#### Key Generation using Java

**AIM:** Key generation (public and private key pair) can be performed using java. Write a program which can do this.

**PROGRAM:**

import java.security.KeyPair;

import java.security.KeyPairGenerator; import java.security.PrivateKey; import java.security.PublicKey;

public class DSAKeyGen {

public static void main(String[] argv) throws Exception {

// TODO code application logic here

// Generate a 1024-bit Digital Signature Algorithm (DSA) key pair KeyPairGeneratorkeyGen = KeyPairGenerator.getInstance("DSA"); keyGen.initialize(1024);

KeyPairkeypair = keyGen.genKeyPair(); PrivateKeyprivateKey = keypair.getPrivate(); System.out.println(privateKey); PublicKeypublicKey = keypair.getPublic(); System.out.println(publicKey);

}

}

**OUTPUT:**

Sun DSA Private Key parameters:

p:

fd7f5381 1d751229 52df4a9c 2eece4e7 f611b752 3cef4400 c31e3f80 b6512669 455d4022 51fb593d 8d58fabf c5f5ba30 f6cb9b55 6cd7813b 801d346f f26660b7 6b9950a5 a49f9fe8 047b1022 c24fbba9 d7feb7c6 1bf83b57 e7c6a8a6 150f04fb 83f6d3c5 1ec30235 54135a16 9132f675 f3ae2b61 d72aeff2 2203199d d14801c7

q:

9760508f 15230bcc b292b982 a2eb840b f0581cf5

G NATA RAJU M.Tech (CNIS) – II Sem Roll No: 11031D6427

Program No: Page No: 25 g:

f7e1a085 d69b3dde cbbcab5c 36b857b9 7994afbb fa3aea82 f9574c0b 3d078267

5159578e bad4594f e6710710 8180b449 167123e8 4c281613 b7cf0932 8cc8a6e1

3c167a8b 547c8d28 e0a3ae1e 2bb3a675 916ea37f 0bfa2135 62f1fb62 7a01243b cca4f1be a8519089 a883dfe1 5ae59f06 928b665e 807b5525 64014c3b fecf492a x: 5e84c621 72f648fd 172b9ef9 1f3d3259 225bd0d4

Sun DSA Public Key Parameters:

p:

fd7f5381 1d751229 52df4a9c 2eece4e7 f611b752 3cef4400 c31e3f80 b6512669 455d4022 51fb593d 8d58fabf c5f5ba30 f6cb9b55 6cd7813b 801d346f f26660b7 6b9950a5 a49f9fe8 047b1022 c24fbba9 d7feb7c6 1bf83b57 e7c6a8a6 150f04fb 83f6d3c5 1ec30235 54135a16 9132f675 f3ae2b61 d72aeff2 2203199d d14801c7

q:

9760508f 15230bcc b292b982 a2eb840b f0581cf5 g:

f7e1a085 d69b3dde cbbcab5c 36b857b9 7994afbb fa3aea82 f9574c0b 3d078267 5159578e bad4594f e6710710 8180b449 167123e8 4c281613 b7cf0932 8cc8a6e1

3c167a8b 547c8d28 e0a3ae1e 2bb3a675 916ea37f 0bfa2135 62f1fb62 7a01243b cca4f1be a8519089 a883dfe1 5ae59f06 928b665e 807b5525 64014c3b fecf492a y:

72047306 ba96a791 1dae6991 f31c1728 40a1c799 8c5be82d 2e458eff e20b3f02 b09e5e39 c0bf385c be4f559b 2ddfedd4 7100c7e4 869267d2 cdda96e0 36fdacca 1bd6f482 56aa5503 690ae509 3b488d03 c3609eb5 0604f60f b43c77b1 ec11c04d db1ff501 ecb03763 6cf014d5 044ad7d5 cd987ee5 2e803278 52638378 b4bb844

#### Program to perform Digital Signature on text

**AIM:** Write a program in java, which performs a digital signature on a given text.

**PROGRAM:**

import java.security.KeyPair;

import java.security.KeyPairGenerator; import java.security.Signature;

import sun.misc.BASE64Encoder; public class DigSign {

public static void main(String[] args) throws Exception {

// TODO code application logic here

KeyPairGeneratorkpg = KeyPairGenerator.getInstance("RSA"); kpg.initialize(1024);

KeyPairkeyPair = kpg.genKeyPair();

byte[] data = "Sample Text".getBytes("UTF8"); Signature sig = Signature.getInstance("MD5WithRSA"); sig.initSign(keyPair.getPrivate());

sig.update(data);

byte[] signatureBytes = sig.sign();

System.out.println("Signature: \n" + new BASE64Encoder().encode(signatureBytes));

sig.initVerify(keyPair.getPublic()); sig.update(data); System.out.println(sig.verify(signatureBytes));

}

}

**OUTPUT:**

Signature: imwaKe99tkM6H6hiiP0rubmb/MrYJZLiwLdRSjslF2KlA5B23az5M2LKftQFCB+NH Ce5F5/YfN8OsNSNLtucrrZTah0SrdWSzdGCOfYLdUZmPQ72j1SkLhYspsTsUb/U6 FPSYT4QebNSYobDtjKujkHdRimHI9TO4lLuqVQRdWU= true

# Viva Questions

##### Define Information Security ? Answer :

**Information security -** is the protection of information and its critical elements, including the systems and hardware that use, store, and transmit the information.

##### What Are The Critical Characteristics Of Information? Answer :

o Availability

o Accuracy

o Authenticity

o Confidentiality

o Integrity

o Utility

o Possession

##### What Is An Attack? Answer :

An attack is the deliberate act that exploits vulnerability

It is accomplished by a threat-agent to damage or steal an organization's information or physical asset

An exploit is a technique to compromise a system

A vulnerability is an identified weakness of a controlled system whose controls are not present or are no longer effective

An attack is then the use of an exploit to achieve the compromise of a controlled system.

##### What Is Distributed Denial-of-service (ddos)? Answer :

DDoS is an attack in which a coordinated stream of requests is launched against a target from many locations at the same time

##### Define Man-in-the-middle ? Answer :

Man-in-the-middle is an attacker sniffs packets from the network, modifies them, and inserts them back into the network.

##### Which is the better security measure, HTTPS, or SSL?

A: HTTPS (Hypertext Transfer Protocol Secure) is HTTP combined with SSL, encrypting a user’s browsing activity and making it safer. SSL (Secure Sockets Layer) is a protocol that protects Internet conversations between two or more parties. Though it’s close, SSL wins out in terms of sheer security, though any of these are valuable things to know for the purposes of [web development.](https://www.simplilearn.com/full-stack-web-developer-mean-stack-certification-training)

##### What is decryption? What is its need?

Cryptography has two important modules and they are encryption and second is decryption. Encryption is basically an approach that converts information into secret codes. It is also known as encoding. It is done to make the information secure. On the other side decryption is a process that is opposite to it i.e. converting the coded information back to its actual form. only the receiver knows the protocols to decode that information.

##### What do you mean by Secret Key Cryptography and Public Key Cryptography? How they are different from one another

Both these are the algorithms of encryption and contribute to data security. Secret Key Cryptography can be used for both encryptions as well as decryption. It is also considered an asymmetric approach and contains only one key. On the other hand, Public Key cryptography is basically an asymmetric approach. There are two keys in this approach and one is basically the public key and any user can access the information. The other key is private and can only be accessed by the administrator.

##### What exactly do you know about RSA?

It is basically a public key cryptography approach that is based on encryption, as well as authentication. It was first used in the year 1977 and is based on prime number logic. It is basically a fast approach that can handle multiple operations at a time. However, if the key size is small, it generally performs its operation slower.

##### What is the Digital Signature Algorithm?

The digital signature algorithm was implemented for the authentication of data in the year 1994. It offers quick signature generation and in addition to this, it ensures better verification of information it is dealing with. It is actually based on computing discrete algorithms and its security actually depends on the size of the key. It can handle key size up to 1024 bits presently.

##### What is the One-Way function?

It is basically an approach that is used to assure faster information processing for cryptography when data needs to be sent in one way only and no reverse action is possible. Functions can be computed in one direction within a very short span of time but performing the opposite task can be extremely daunting.

##### What is the Fast Data Encipherment Algorithm?

It is basically a cryptosystem that has a 64-bit block size. Its prime aim is to assure the performance of the software in which it is used. However, due to its insecure approach, it is not so common in the present scenario.

##### What is the Caesar cipher?

In the Caesar methodology, each letter of the text or the written message is substituted with another letter of the alphabet which is so many spaces or letters later in the alphabet. This is probably the simplest form of encryption, because each letter in plain text message is literally substituted by another letter, thus forming the ciphertext. This methodology (which was said to be used by Julius Caesar) is probably the most-cited type of algorithm in academic literature.

##### What are symmetric and asymmetric key systems?

A symmetric key system uses only the private key, and the asymmetric key system makes use of both the public key and the private key. The latter used primarily in what is known as a Public Key Infrastructure, or PKI for short. It will be discussed in more detail later on.

1. What are the mathematical algorithms used in symmetric cryptography? They are as follows:

The Needham-Schroder algorithm

The Digital Encryption Standard algorithm (DES)

The Triple Digit Encryption Standard algorithm (3DES)

The International Data Encryption Algorithm (IDEA) The Advanced Encryption Standard algorithm (AES)

1. What is the hashing function?

The hashing function is a one-way mathematical function. This means that it can be used to encode data, but it cannot decode data. Its primary purpose is not to encrypt the ciphertext; rather, its primary purpose is to prove that the message in the ciphertext has not changed in any way, shape or form. This is also referred to as “message integrity.” If the mathematical function has changed in any way, the message has then changed.

##### What are the different types of Ciphers? Ans

* Mono-alphabetic Ciphers
* Polyalphabetic Ciphers
* Transpositions and Grills
* Steganography
* Codes
* Voice Scramblers
* Modern Ciphers

##### What is Block Cipher?

**Ans:** Block cipher is a method of encrypting data using cryptographic keys and algorithms to apply to a block or chunks of the message simultaneously rather than individually. The transposition cipher is an example of Block cipher.

##### What is Stream Cipher?

**Ans:** In this cipher, the cryptographic algorithm is used to encrypt or decrypt a message one bit or character at a time. The Caesar cipher is an example of the stream cipher.

##### What is the International Data Encryption Algorithm (IDEA)?

**Ans:** International Data Encryption Algorithm (IDEA) algorithm is a symmetric-key block cipher that operates on 64-bit blocks using a 128-bit key.