KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20

T.Y. B.Sc. I.T. Semester VI

INDEX

KISHINCHAND CHELLARAM COLLEGE, MUMBAI - 20

T.Y. B.Sc. I.T. Semester VI

Sr. No.	Topic	Date	Signature
6	Implement RSA Algorithm	08|12|22	
7	Implement RC4 Algorithm	08|12|22	
8	Implement Blowfish Algorithm	08|12|22	
9	Configure OSPF MD5 authentication and support SSH connections	19|12|22	
10	Configure AAA authentication on routers	08|01|23	
11	Configure a Zone-Based Policy Firewall (ZPF)	19|01|23	

Practical 1 – Implementation of Substitution Techniques

Practical 1A

Aim: To implement Caesar Cipher

Theory:

Julius Caesar is said to have been the first to use the scheme, in which each letter is translated to a letter a fixed number of places after it in alphabet. Caesar used a shift of 3, so that the plaintext letter P_i was enciphered as cipher text letter C_i by the rule

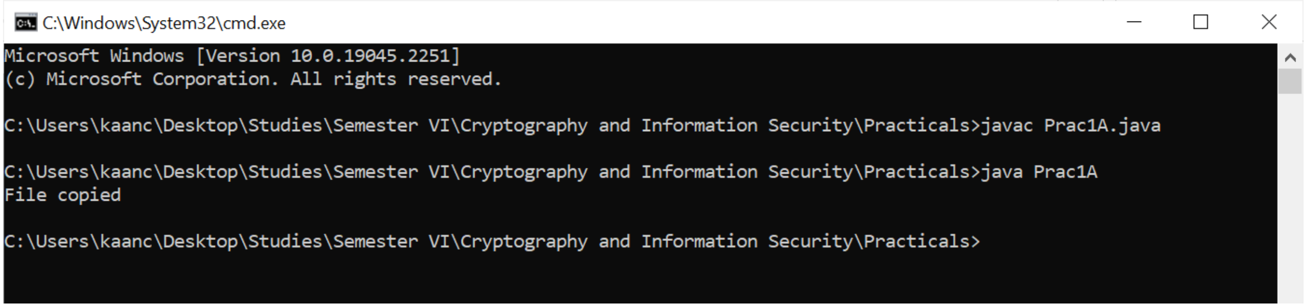
$C_i = E (P_i) = P_i + 3$. The alphabet to be replaced will be the one that is 3 places down the line.

Code:

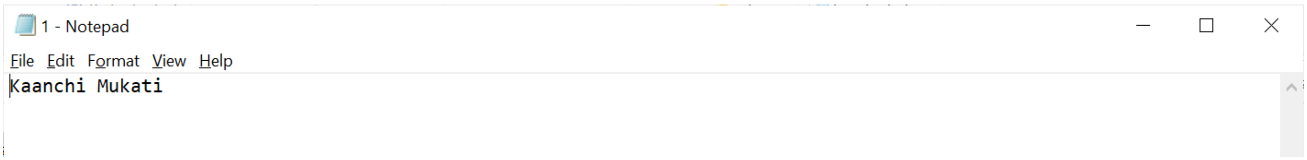
```
import java.io.*;
class Prac1A
{
    public static void main(String args[])
    {
        int i;
        try
        {
            File f1 = new File("1.txt");
            File f2 = new File("2.txt");
            File f3 = new File("3.txt");

            FileInputStream fis = new FileInputStream(f1);
            FileOutputStream fos = new FileOutputStream(f2);
            FileOutputStream os = new FileOutputStream(f3);

            while((i=fis.read()) != -1)
            {
                fos.write(i);
                os.write(i+6);
            }

            System.out.println("File copied");
        }
```

```
                catch(Exception e)
                {

                        e.printStackTrace();
                }
        }
}
```

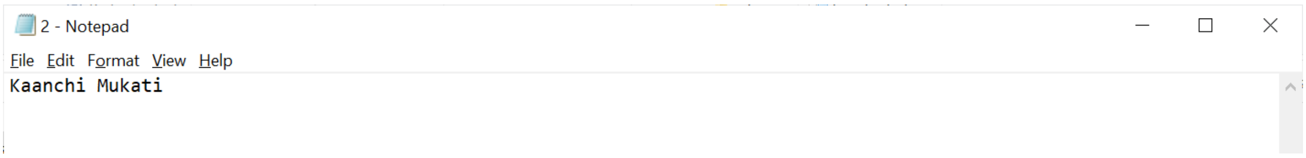
Output:



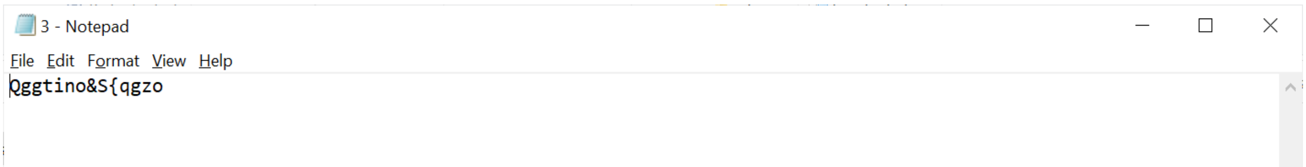
```
Microsoft Windows [Version 10.0.19045.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>javac Prac1A.java

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>java Prac1A
File copied

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>
```



```
Kaanchi Mukati
```



```
Kaanchi Mukati
```



```
Qggtino&S{qgzo
```

Practical 1B

Aim: To implement Modified Caesar Cipher

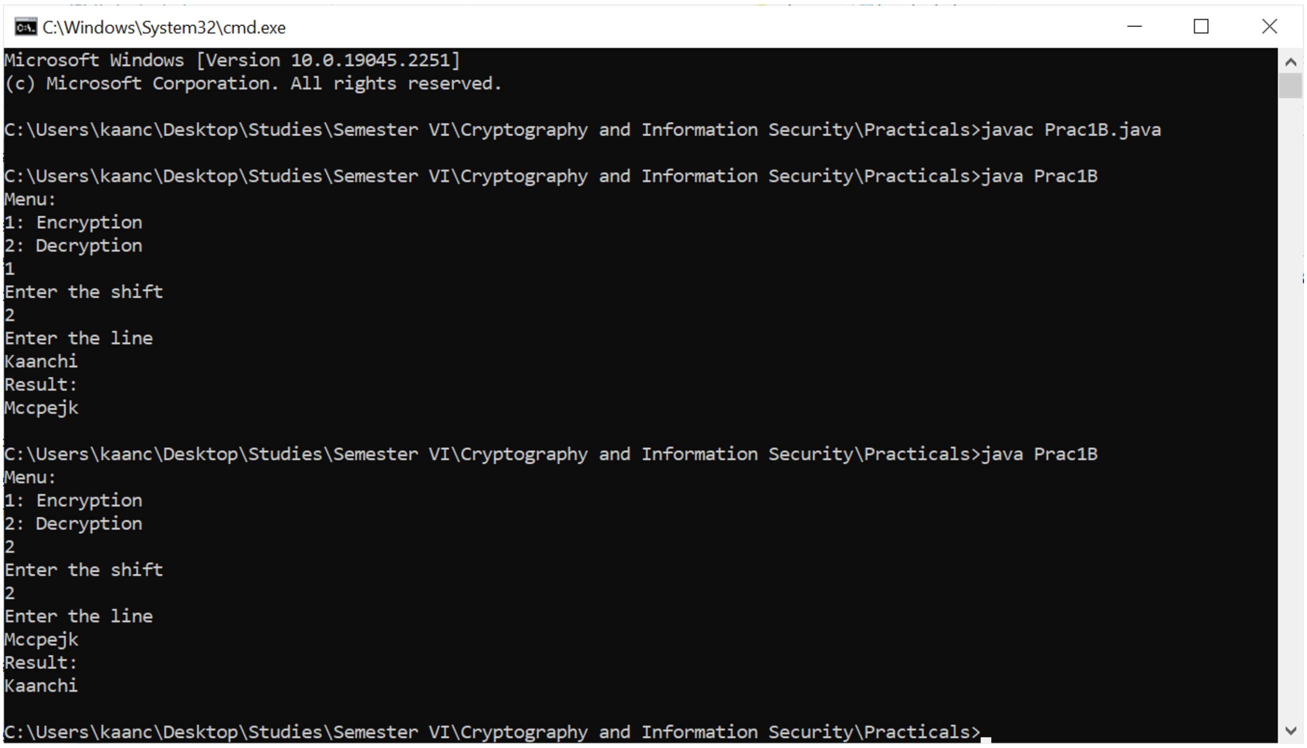
Theory:

In Modified Caesar Cipher the original plain text alphabets may not necessarily be three places down the line, but instead can be any places down the line. Once the replacement scheme is decided, it would be constant and will be used for all the other alphabets in that message.

Code:

```
import java.io.*;
class Prac1B
{
    public String Encrypt(int shift, String line)
    {
        String result = "";
        int offset;
        for(int i=0;i<line.length();i++)
        {
            offset = ((int)line.charAt(i)+shift)%256;
            result += (char)(offset);
        }
        return result;
    }

    public String Decrypt(int shift, String line)
    {
        String result = "";
        int offset;
        for(int i=0;i<line.length();i++)
        {
            offset = ((int)line.charAt(i)-shift)%256;
            if(offset<0)
            {
                offset+= 256;
            }
            result+= (char)(offset);
        }
        return result;
    }
    public static void main(String args[])
    throws IOException
    {
        Prac1B b = new Prac1B();
        BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
        int choice;
        System.out.println("Menu:\n1: Encryption\n2:
Decryption");
        choice = Integer.parseInt(br.readLine());
        System.out.println("Enter the shift");
        int shift = Integer.parseInt(br.readLine());
        System.out.println("Enter the line");
        String line = br.readLine();
        System.out.println("Result: ");
        switch(choice)
        {
            case 1:
            {
                System.out.println(b.Encrypt(shift, line));
                break;
            }

            case 2:
            {
                System.out.println(b.Decrypt(shift, line));
                break;
            }

            default:
            {
                System.out.println("Invalid choice");
            }
        }
    }
}
```

Output:



```
Microsoft Windows [Version 10.0.19045.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>javac Prac1B.java

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>java Prac1B
Menu:
1: Encryption
2: Decryption
1
Enter the shift
2
Enter the line
Kaanchi
Result:
Mccpejk

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>java Prac1B
Menu:
1: Encryption
2: Decryption
2
Enter the shift
2
Enter the line
Mccpejk
Result:
Kaanchi

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>
```

Practical 1C

Aim: To implement Mono-alphabetic Cipher

Theory:

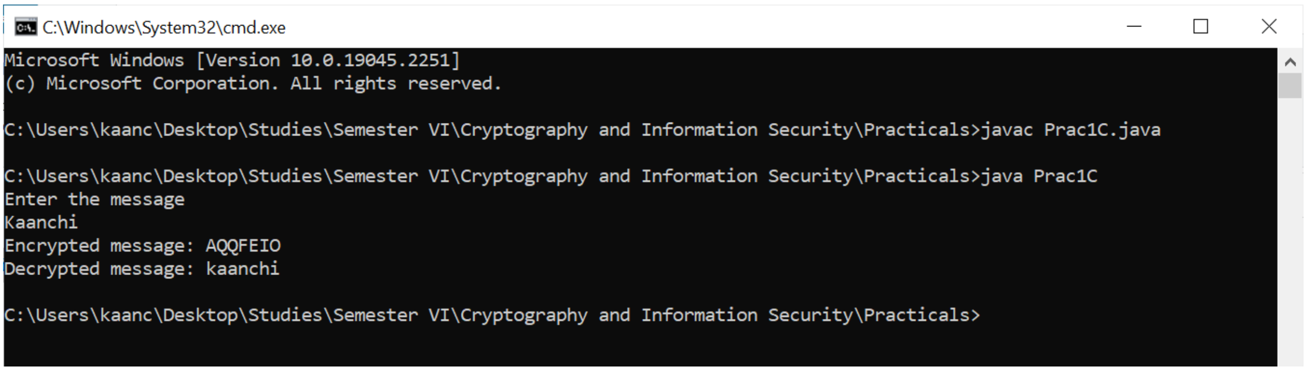
The mono-alphabetic substitution cipher is so called because each plain text letter is substituted by the same cipher text letter throughout the entire message.

Code:

```
import java.util.Scanner;
public class Prac1C
{
     public static char p[] =
{'a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q'
,'r','s','t','u','v','w','x','y','z'};
     public static char ch[] =
{'Q','W','E','R','T','Y','U','I','O','P','A','S','D','F','G','H','J'
,'K','L','Z','X','C','V','B','N','M'};

     public static String doEncryption(String s)
     {
          char c[] = new char[(s.length())];
          for(int i=0;i<s.length();i++)
          {
               for(int j=0;j<26;j++)
               {
                    if(p[j]==s.charAt(i))
                    {
                         c[i] = ch[j];
                         break;
                    }
               }
          }
          return (new String(c));
     }

     public static String doDecryption(String s)
     {
          char p1[] = new char[(s.length())];
          for(int i=0;i<s.length();i++)
          {
               for(int j=0;j<26;j++)
```

```
            {
                if(ch[j] == s.charAt(i))
                {
                    p1[i] = p[j];
                    break;
                }
            }
        }
        return(new String(p1));
    }

    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the message");
        String en = doEncryption(sc.next().toLowerCase());
        System.out.println("Encrypted message: " + en);
        System.out.println("Decrypted message: " +
doDecryption(en));
        sc.close();
    }
}
```

Output:



```
Microsoft Windows [Version 10.0.19045.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>javac Prac1C.java

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>java Prac1C
Enter the message
Kaanchi
Encrypted message: AQQFEIO
Decrypted message: kaanchi

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>
```

Practical 2 – Implementation of Transposition Techniques

Practical 2A

Aim: To implement Rail-Fence Technique

Theory:

Rail fence Technique involves writing plain text as sequence of diagonals and then reading it row-by-row to produce cipher text.

Algorithm:

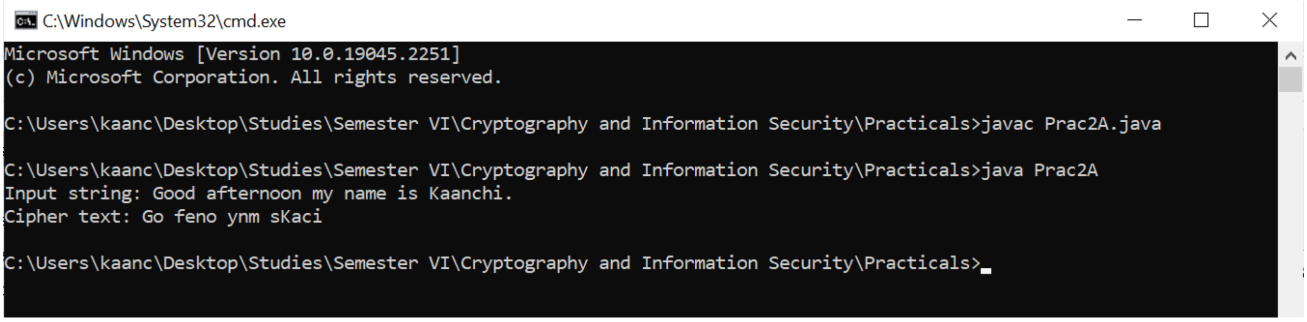
Step 1: Write down the plain text message as a sequence of diagonals.

Step 2: Read the plain text written in step 1 as a sequence of rows.

Code:

```
class Prac2A
{
	public static void main(String args[])
	{
		String input = "Good afternoon my name is Kaanchi.";
		String output = "";
		int len = input.length();
		System.out.println("Input string: " + input);
		for(int i=0;i<len;i+=2)
		{
			output += input.charAt(i);
		}
		System.out.println("Cipher text: " +output);
	}
}
```

Output:



```
Microsoft Windows [Version 10.0.19045.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>javac Prac2A.java

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>java Prac2A
Input string: Good afternoon my name is Kaanchi.
Cipher text: Go feno ynm sKaci

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>
```

Practical 2B

Aim: To implement Vernam Cipher

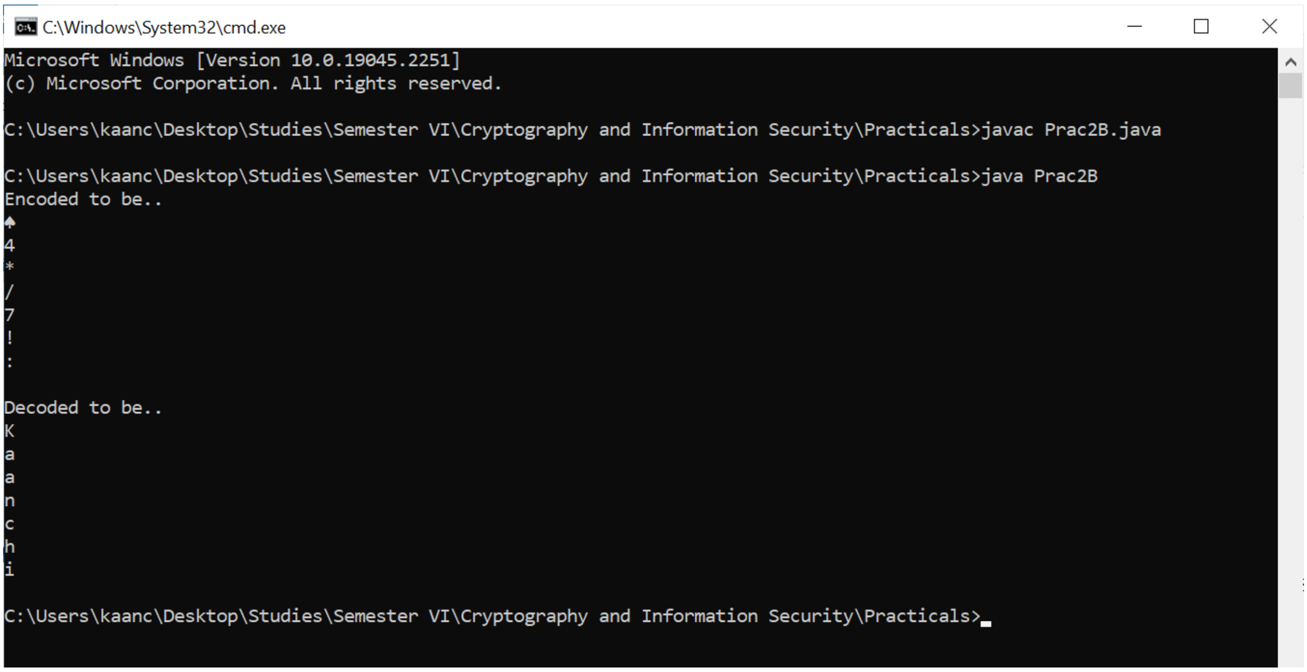
Theory:

Verman Cipher is also known as the one-time-pad. Gilbert Vernam invented and patented his cipher in 1917 while working at AT&T. The teletype had been recently introduced, and along with this the commercial Baudot code. Then, messages were uniformly thought of as streams of zero's and one's. Verman proposed a bit-wise exclusive or of the message stream with a truly random zero-one stream which was shared by sender and recipient.

Code:

```
import java.lang.Math;
public class Prac2B
{
     public static void main(String args[])
     {
          String str = new String("Kaanchi");
          char[] arText = str.toCharArray();
          String cipher = new String("MUKATIS");
          char[] arCipher = cipher.toCharArray();
          char[] encoded = new char[7];
          System.out.println("Encoded to be..");
          for(int i=0;i<arText.length;i++)
          {
               encoded[i] = (char)(arText[i] ^ arCipher[i]);
               System.out.println(encoded[i]);
          }
          System.out.println("\nDecoded to be..");
          for(int i=0;i<encoded.length; i++)
          {
               char tem = (char)(encoded[i] ^ arCipher[i]);
               System.out.println(tem);
          }
     }
}
```

Output:



```
Microsoft Windows [Version 10.0.19045.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>javac Prac2B.java

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>java Prac2B
Encoded to be..
♠
4
*
/
7
!
:

Decoded to be..
K
a
a
n
c
h
i

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>
```

Practical 3

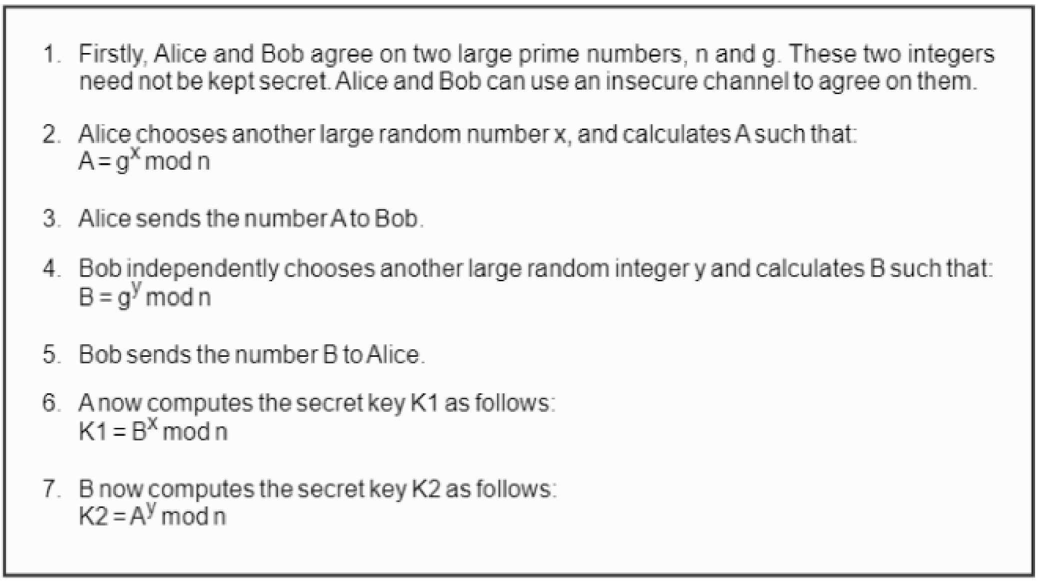
Aim: To implement Diffie-Hellman’s Key Exchange Algorithm

Theory:

- Whitefield Diffie and Martin Hellman devised a solution to the problem of key agreement or key exchange in 1976.
- This solution is called as Diffie-Hellman key exchange / Agreement Algorithm.
- The two parties who want to communicate securely can agree on a symmetric key using this technique.
- This key then is used for encryption and decryption.
- The Diffie-Hellman key exchange algorithm can be used only for key agreement but not for encryption or decryption of messages.

Algorithm:

Pg 66 of Chap 2



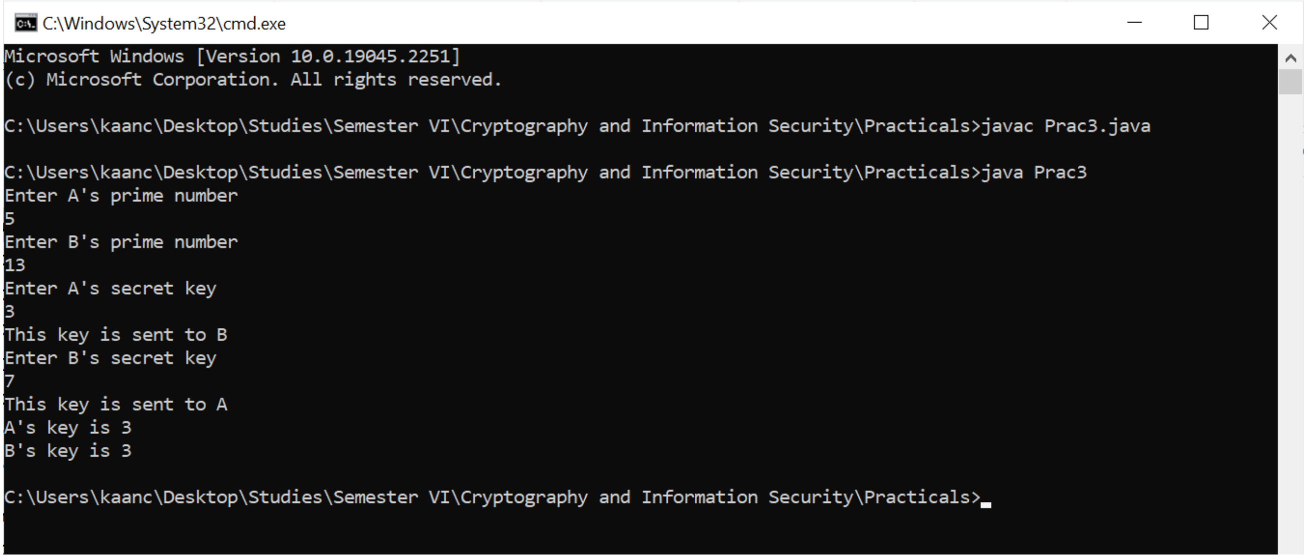
1. Firstly, Alice and Bob agree on two large prime numbers, n and g. These two integers need not be kept secret. Alice and Bob can use an insecure channel to agree on them.
2. Alice chooses another large random number x, and calculates A such that:
   $A = g^x \bmod n$
3. Alice sends the number A to Bob.
4. Bob independently chooses another large random integer y and calculates B such that:
   $B = g^y \bmod n$
5. Bob sends the number B to Alice.
6. A now computes the secret key K1 as follows:
   $K1 = B^x \bmod n$
7. B now computes the secret key K2 as follows:
   $K2 = A^y \bmod n$

Fig. 2.49 *Diffie–Hellman key exchange algorithm*

Code:

```
import java.math.BigInteger;
import java.security.KeyFactory;
import java.security.KeyPair;
import java.util.*;
import java.math.BigInteger.*;
import java.math.*;

class Prac3
{
	public static void main(String args[])
	{
		BigInteger a,b,k1,k2,x,y,g,n;
		Scanner s = new Scanner(System.in);

		System.out.println("Enter A's prime number");
		n = s.nextBigInteger();

		System.out.println("Enter B's prime number");
		g = s.nextBigInteger();

		System.out.println("Enter A's secret key");
		x = s.nextBigInteger();

		System.out.println("This key is sent to B");

		System.out.println("Enter B's secret key");
		y = s.nextBigInteger();

		System.out.println("This key is sent to A");

		a = g.modPow(x,n);
		b = g.modPow(y,n);
		k1 = b.modPow(x,n);
		k2 = a.modPow(y,n);

		System.out.println("A's key is " + k1);
		System.out.println("B's key is " + k2);
	}
}
```

Output:



```
Microsoft Windows [Version 10.0.19045.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>javac Prac3.java

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>java Prac3
Enter A's prime number
5
Enter B's prime number
13
Enter A's secret key
3
This key is sent to B
Enter B's secret key
7
This key is sent to A
A's key is 3
B's key is 3

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>
```

Practical 4

Aim: To implement Data Encryption Standard Algorithm

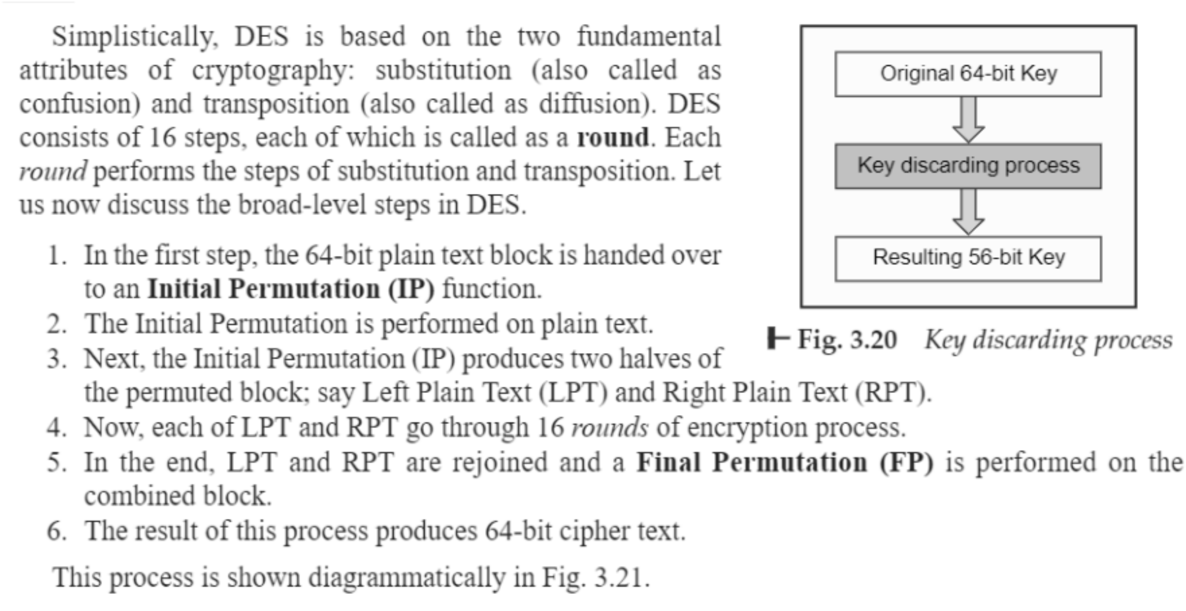
Theory:

The algorithm derives its strength from the repeated applications of these two techniques one on the top of the other, for a total of 16 cycles. The algorithm begins by encrypting plaintext as a block of 64-bits. The key is 64- bit long, but it can be any 56-bit number. The extra 8-bits are often used as check digit. The key can be changed by user security purpose.

In DES algorithm the techniques Substitution provides the Confusion and Transposition provides Diffusion. DES uses only standard arithmetic and logical operations on number up to 64-bit long. So, it can be used in current computers. The algorithm is repetitive, making it suitable for implementation on a single purpose chip. The chip can be used as basic component in devices that uses DES encryption in application.

Algorithm:

Pg 102 of chap 3



Original 64-bit Key
Key discarding process
Resulting 56-bit Key

Fig. 3.20 *Key discarding process*

Simplistically, DES is based on the two fundamental attributes of cryptography: substitution (also called as confusion) and transposition (also called as diffusion). DES consists of 16 steps, each of which is called as a **round**. Each *round* performs the steps of substitution and transposition. Let us now discuss the broad-level steps in DES.

1. In the first step, the 64-bit plain text block is handed over to an **Initial Permutation (IP)** function.
2. The Initial Permutation is performed on plain text.
3. Next, the Initial Permutation (IP) produces two halves of the permuted block; say Left Plain Text (LPT) and Right Plain Text (RPT).
4. Now, each of LPT and RPT go through 16 *rounds* of encryption process.
5. In the end, LPT and RPT are rejoined and a **Final Permutation (FP)** is performed on the combined block.
6. The result of this process produces 64-bit cipher text.

This process is shown diagrammatically in Fig. 3.21.

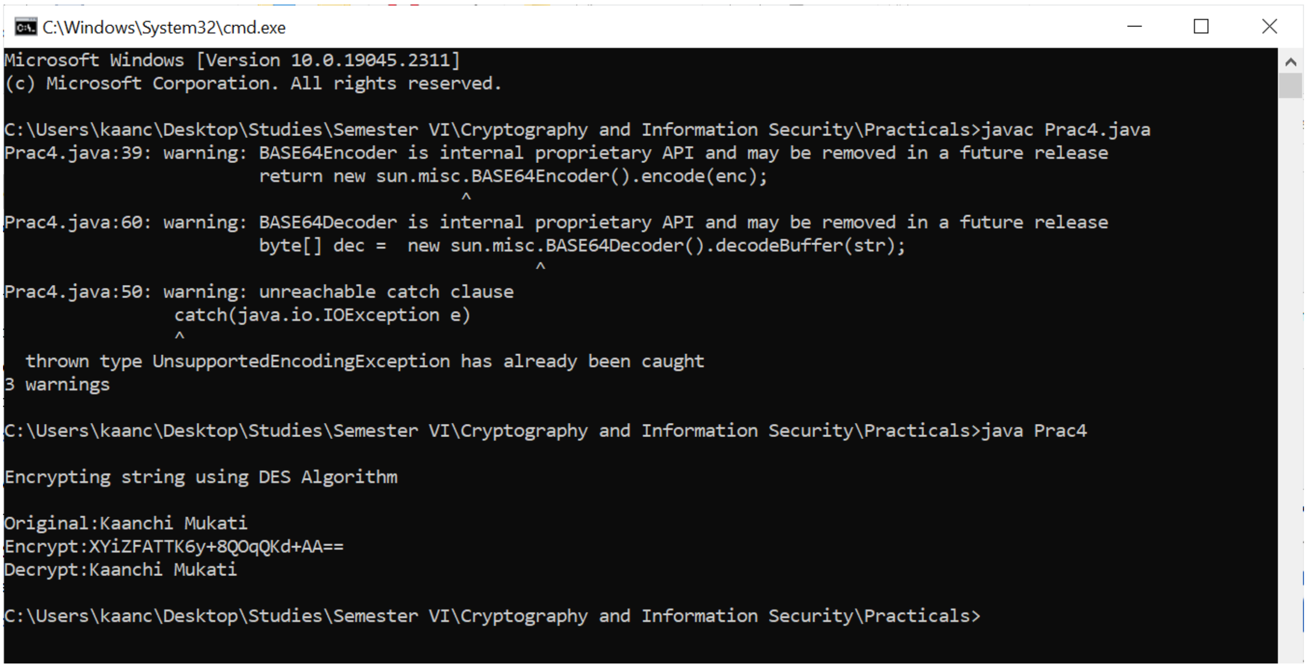
Code:

```
import javax.crypto.*;
import java.io.*;
import java.security.InvalidAlgorithmParameterException;
import java.security.spec.*;
import java.lang.*;
import javax.crypto.spec.IvParameterSpec;

public class Prac4
{
      Cipher ecipher;
      Cipher dcipher;

      Prac4(SecretKey key)
      {
            try
            {
                  ecipher = Cipher.getInstance("DES");
                  dcipher = Cipher.getInstance("DES");
                  ecipher.init(Cipher.ENCRYPT_MODE,key);
                  dcipher.init(Cipher.DECRYPT_MODE,key);
            }
            catch(javax.crypto.NoSuchPaddingException e)
            {
            }
            catch(java.security.NoSuchAlgorithmException e)
            {
            }
            catch(java.security.InvalidKeyException e)
            {
            }
      }

      public String Encrypt(String str)
      {
            try
            {
                  byte[] utf8 = str.getBytes("UTF8");
                  byte[] enc = ecipher.doFinal(utf8);
                  return new sun.misc.BASE64Encoder().encode(enc);
            }
            catch(javax.crypto.BadPaddingException e)
```

```
          {
          }
          catch(IllegalBlockSizeException e)
          {
          }
          catch(UnsupportedEncodingException e)
          {
          }
          catch(java.io.IOException e)
          {
          }
          return null;
     }

     public String Decrypt(String str)
     {
          try
          {
               byte[] dec =  new 
sun.misc.BASE64Decoder().decodeBuffer(str);
               byte[] utf8 = dcipher.doFinal(dec);
               return new String(utf8,"UTF8");
          }
          catch(javax.crypto.BadPaddingException e)
          {
          }
          catch(IllegalBlockSizeException e)
          {
          }
          catch(UnsupportedEncodingException e)
          {
          }
          catch(java.io.IOException e)
          {
          }
          return null;
     }

     public static void main(String args[])
     {
          System.out.println();
          System.out.println("Encrypting string using DES 
Algorithm");
          System.out.println();
```

```
		try
		{
			SecretKey key = 
KeyGenerator.getInstance("DES").generateKey();
			Prac4 encrypter = new Prac4(key);
			String s = "Kaanchi Mukati";
			String d = "Hello";

			String encrypted = encrypter.Encrypt(s);

			String decrypted = encrypter.Decrypt(encrypted);

			System.out.println("Original:" + s);

			System.out.println("Encrypt:" + encrypted);
			System.out.println("Decrypt:" + decrypted);
		}

		catch(Exception e)
		{
		}
	}
}
```

Output:



```
Microsoft Windows [Version 10.0.19045.2311]
(c) Microsoft Corporation. All rights reserved.

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>javac Prac4.java
Prac4.java:39: warning: BASE64Encoder is internal proprietary API and may be removed in a future release
                        return new sun.misc.BASE64Encoder().encode(enc);
                                         ^
Prac4.java:60: warning: BASE64Decoder is internal proprietary API and may be removed in a future release
                        byte[] dec =  new sun.misc.BASE64Decoder().decodeBuffer(str);
                                                ^
Prac4.java:50: warning: unreachable catch clause
                catch(java.io.IOException e)
                ^
  thrown type UnsupportedEncodingException has already been caught
3 warnings

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>java Prac4

Encrypting string using DES Algorithm

Original:Kaanchi Mukati
Encrypt:XYiZFATTK6y+8QOqQKd+AA==
Decrypt:Kaanchi Mukati

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>
```

Practical 5

Aim: To implement Advanced Encrypting Standard Algorithm

Theory:

The AES was adopted for used by the U.S government in December, 2001 and become Federal Information Processing Standard 197. The AES algorithm has strong mathematical foundation. It primarily uses the Substitution, Transposition and the Shift, Exclusive OR and additional operations. AES uses repeat cycles. There are 9, 11 or 13 cycles for 128, 192 or 256 bits key.

Algorithm:

The AES algorithm consist of cycles, each cycle consists of four steps.

Step 1: Byte Substitution:

This step uses the substitution box structure similar to DES, Substituting each byte of a 128-bit block according to the substitution table. This is straight confusion operation.

Step 2: Shift Row:

It is a transposition step. For 128 and 192 bits size block, row n is shifted left circular (n-1) bytes and for 256-bit block, row 2 is shifted by 1 byte and row 3 and 4 are shifted by 3 and 4 bytes.

Step 3: Mix Column:

This step involves shifting left and exclusive OR ing bits with themselves. The operation provides both Confusion and Diffusion.

Step 4: Add Subkey:

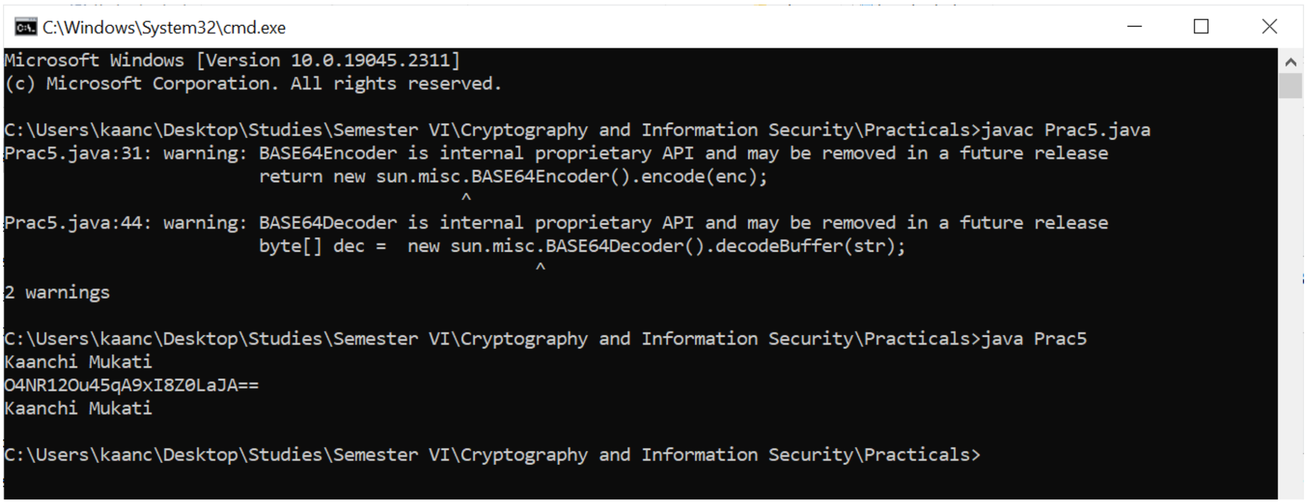
Here portion of key unique to this cycle is exclusive OR ed with cycle Results. This operation provides Confusion and incorporates the key.

Code:

```
import java.security.*;
import javax.crypto.*;
import java.io.*;

public class Prac5
{
	Cipher ecipher;
	Cipher dcipher;

	Prac5(SecretKey key)
	{
		try
		{
			ecipher = Cipher.getInstance("AES");
			dcipher = Cipher.getInstance("AES");
			ecipher.init(Cipher.ENCRYPT_MODE, key);
			dcipher.init(Cipher.DECRYPT_MODE, key);
		}
		catch(Exception e)
		{
			System.out.println(e.getMessage());
		}
	}

	public String Encrypt(String str)
	{
		try
		{
			byte[] utf8 = str.getBytes("UTF8");
			byte[] enc = ecipher.doFinal(utf8);
			return new sun.misc.BASE64Encoder().encode(enc);
		}
		catch(Exception e)
		{
			e.printStackTrace();
		}
		return null;
	}

	public String Decrypt(String str)
	{
		try
		{
			byte[] dec =  new
sun.misc.BASE64Decoder().decodeBuffer(str);
			byte[] utf8 = dcipher.doFinal(dec);
			return new String(utf8,"UTF8");
		}

		catch(Exception e)
```

```
                {

                        System.out.println(e.getMessage());
                }

                return null;
        }

        public static void main(String[] args)
        {
                SecretKey key = null;
                try
                {
                        KeyGenerator keyGen = 
KeyGenerator.getInstance("AES");
                        key = keyGen.generateKey();
                }

                catch(Exception e)
                {
                        e.printStackTrace();
                }

                Prac5 dese = new Prac5(key);
                String o = "Kaanchi Mukati" ;
                String e = dese.Encrypt(o);
                String d = dese.Decrypt(e);

                System.out.println(o);
                System.out.println(e);
                System.out.println(d);
        }
}
```

Output:



```
Microsoft Windows [Version 10.0.19045.2311]
(c) Microsoft Corporation. All rights reserved.

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>javac Prac5.java
Prac5.java:31: warning: BASE64Encoder is internal proprietary API and may be removed in a future release
                        return new sun.misc.BASE64Encoder().encode(enc);
                                           ^
Prac5.java:44: warning: BASE64Decoder is internal proprietary API and may be removed in a future release
                        byte[] dec =  new sun.misc.BASE64Decoder().decodeBuffer(str);
                                                  ^
2 warnings

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>java Prac5
Kaanchi Mukati
O4NR12Ou45qA9xI8Z0LaJA==
Kaanchi Mukati

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>
```

Practical 6

Aim: To implement RSA (Rivest Shamir Adleman) Algorithm

Theory:

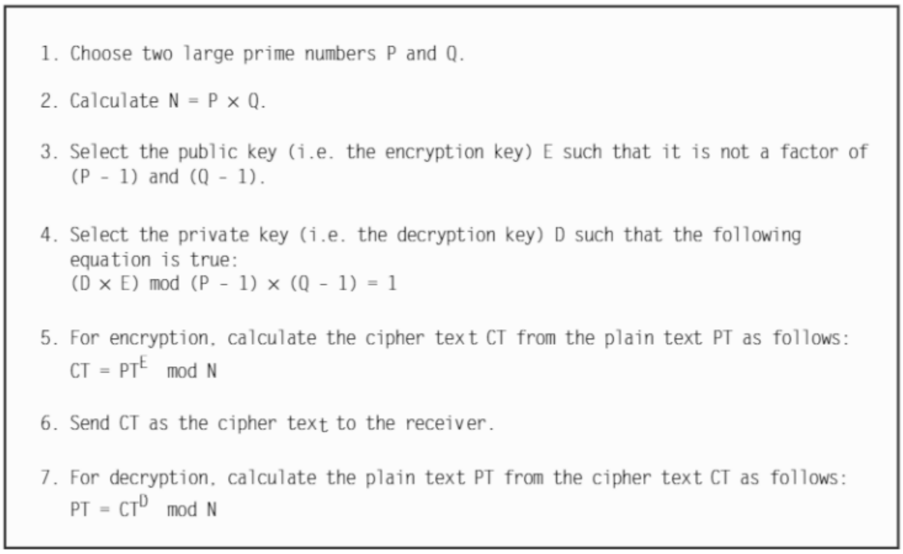
RSA is similar to other methods such as Merkle-Hellman, in that solving the encryption amounts to finding terms that add to a particular sum or multiply to a particular product. It relies on an area of mathematics known as number theory, in which mathematicians study properties of numbers such as their prime factors. The RSA encryption algorithm combines results from number theory with the degree of difficulty in determining the prime factor of a given number. The RSA algorithm also operates with arithmetic mod n.

The two keys used in RSA, d and e, are used for decryption and encryption. They are actually interchangeable: Either can be chosen as a public key but, having chosen one, you must keep the other one private. For simplicity, we call the encryption key e and the decryption key d. Also, because of the nature of the RSA algorithm, the key can be applied in either order:

$P = E(D(P)) = D(E(P))$

Algorithm:

Pg 157 of chap 3



1. Choose two large prime numbers P and Q.
2. Calculate N = P × Q.
3. Select the public key (i.e. the encryption key) E such that it is not a factor of (P - 1) and (Q - 1).
4. Select the private key (i.e. the decryption key) D such that the following equation is true:
   (D × E) mod (P - 1) × (Q - 1) = 1
5. For encryption, calculate the cipher text CT from the plain text PT as follows:
   CT = PT^E mod N
6. Send CT as the cipher text to the receiver.
7. For decryption, calculate the plain text PT from the cipher text CT as follows:
   PT = CT^D mod N

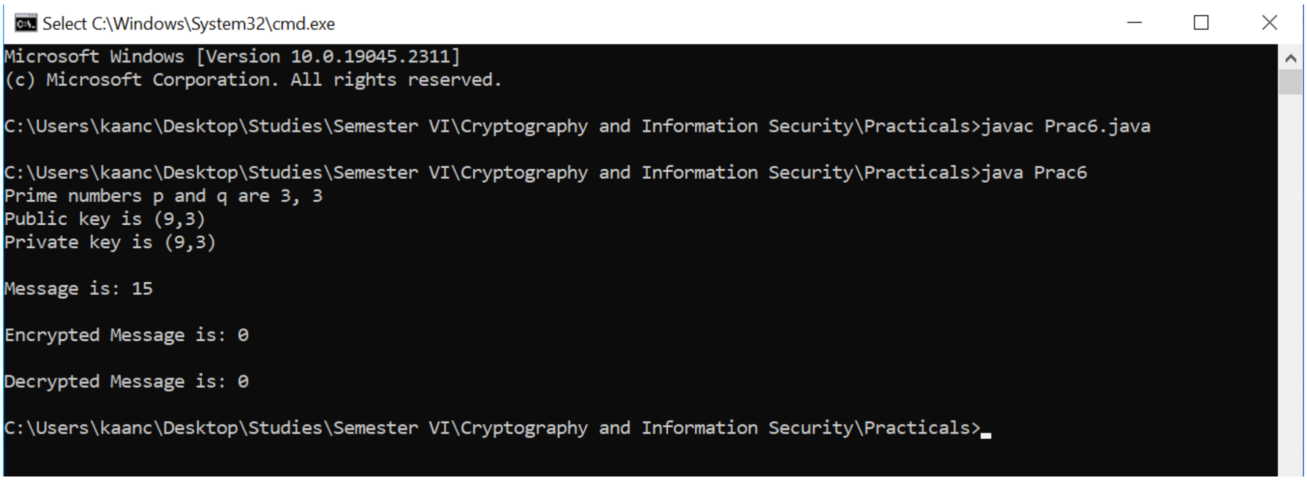
Fig. 4.3 *The RSA algorithm*

Code:

```
import java.math.*;
import java.security.*;

public class Prac6
{
      BigInteger p,q,n,d,e,ph,t;
      SecureRandom r;

      public Prac6()
      {
                  r = new SecureRandom();
                  p = new BigInteger(2,0,r);
                  q = new BigInteger(2,0,r);
                  System.out.println("Prime numbers p and q are " +
p.intValue() + ", " + q.intValue());
                  n = p.multiply(q);
                  ph = (p.subtract(new BigInteger("1")));
                  ph = ph.multiply(q.subtract(new BigInteger("1")));
                  e = new BigInteger("2");

                  while(ph.gcd(e).intValue()>1 || e.compareTo(ph)!=-1)
                  {
```

```
                           e = e.add(new BigInteger("1"));
                           d = e.modInverse(ph);

                           System.out.println("Public key is ("+
n.intValue() + "," + e.intValue()+")");
                      System.out.println("Private key is ("+
n.intValue() + "," + d.intValue()+")");

                           BigInteger msg = new BigInteger("15");
                           System.out.println("\nMessage is: " +
msg);

                           BigInteger enmsg = encrypt(msg,e,n);
                           System.out.println("\nEncrypted Message
is: " + enmsg.intValue());

                           BigInteger demsg = decrypt(msg,e,n);
                           System.out.println("\nDecrypted Message
is: " + demsg.intValue());
                }
     }

     BigInteger encrypt(BigInteger msg, BigInteger e, BigInteger n)
     {
                return msg.modPow(e,n);
     }
     BigInteger decrypt(BigInteger msg, BigInteger d, BigInteger n)
     {
                return msg.modPow(d,n);
     }
     public static void main(String args[])
     {
                Prac6 r = new Prac6();
     }
}
```

Output:



```
Microsoft Windows [Version 10.0.19045.2311]
(c) Microsoft Corporation. All rights reserved.

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>javac Prac6.java

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>java Prac6
Prime numbers p and q are 3, 3
Public key is (9,3)
Private key is (9,3)

Message is: 15

Encrypted Message is: 0

Decrypted Message is: 0

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>
```

Practical 7

Aim: To implement RC4 (Rivest Cipher 4) Algorithm

Theory:

RC4 was designed by Ron Rivest of RSA Security in 1987; while it is officially termed "Rivest Cipher 4", the RC acronym is alternatively understood to stand for "Ron's Code".

RC4 is a stream cipher, symmetric key algorithm. The same algorithm is used for both encryption and decryption as the data stream is simply XORed with the generated key sequence. The key stream is completely independent of the plaintext used. It uses a variable length key from 1 to 256 bit to initialize a 256-bit state table. The state table is used for subsequent generation of pseudo-random bits and then to generate a pseudo-random stream which is XORed with the plaintext to give the cipher text. The algorithm can be broken into two stages: initialization, and operation. In the initialization stage the 256- bit state table, S is populated, using the key, K as a seed. Once the state table is setup, it continues to be modified in a regular pattern as data is encrypted.

Algorithm:

Step 1: Get the data to be encrypted and the selected key.

Step 2: Create two string arrays.

Step 3: Initiate one array with numbers from 0 to 255.

Step 4: Fill the other array with the selected key.

Step 5: Randomize the first array depending on the array of the key.

Step 6: Randomize the first array within itself to generate the final key stream.

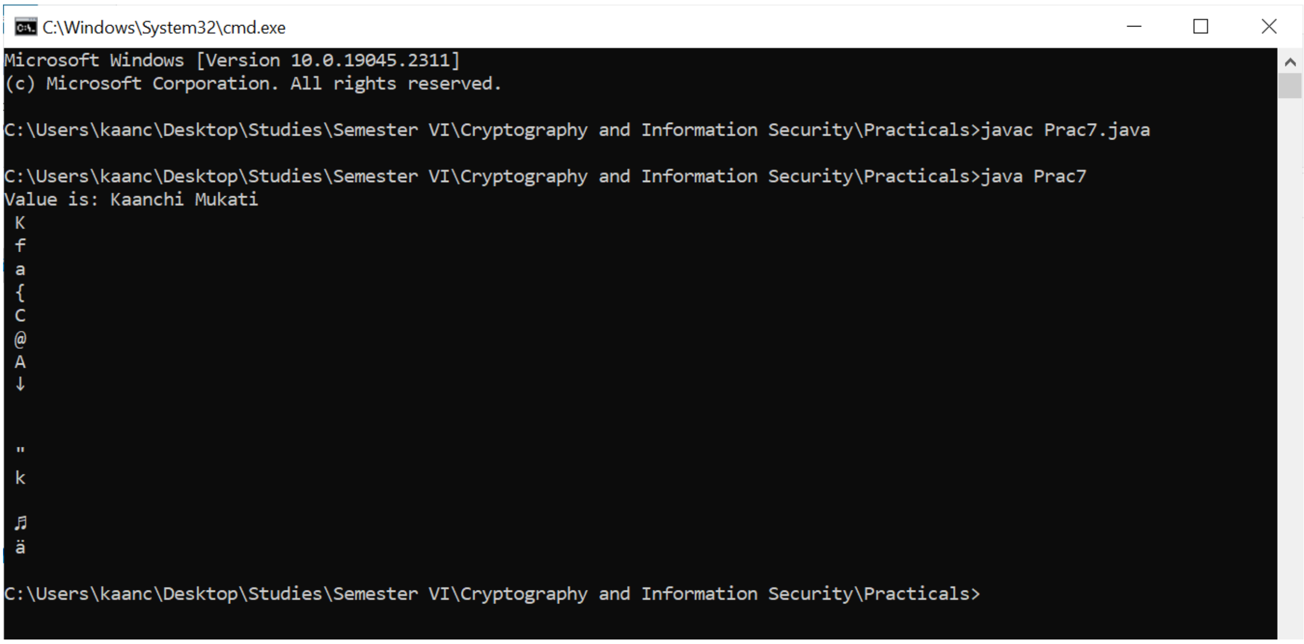
Step 7: XOR final key stream with the data to be encrypted to give cipher text.

Code:

```
class Prac7
{
      String pt;
      static char cipher[];

      Prac7(String pt, int[] key)
      {
```

```
		this.pt = pt;
		int s[] = new int[255];
		cipher = new char[pt.length()];

		for(int i=0;i<s.length;i++)
		{
			s[i] = i;
		}

		int i = 0;
		int j = 0;
		for(int k=0;k<pt.length();k++)
		{
			int modk = (k%key.length);
			int kc = key[modk];
			j = (s[i]+j+kc)%256+1;

			int temp = s[i];
			s[i] = s[j];
			s[j] = temp;
			int sc = (s[i]+s[j])%256;
			int ck = s[sc];
			cipher[k] = (char)(ck^(int)pt.charAt(k));
			i = i+1;
		}
	}

	public static void main(String args[])
	{
		int k[] = {1,2,3,4,5};
		String original = "Kaanchi Mukati";
		System.out.println("Value is: " + original);
		new Prac7(original, k);
		for(int i=0;i<cipher.length;i++)
		{
			System.out.println(" " + cipher[i]);
		}
	}
}
```

Output:



```
Microsoft Windows [Version 10.0.19045.2311]
(c) Microsoft Corporation. All rights reserved.

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>javac Prac7.java

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>java Prac7
Value is: Kaanchi Mukati
 K
 f
 a
 {
 C
 @
 A
 ↓


 "
 k

 ♬
 ä

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>
```

Practical 8

Aim: To implement Blowfish Algorithm

Theory:

Blowfish was developed by Bruce Schneier and has the reputation of being a very strong symmetric key cryptographic algorithm.

Blowfish encrypts 64-bit blocks with a variable-length key. It contains two parts:

1) Subkey generation – This process coverts the key up to 448 bits long to sub-keys totaling 4168 bits.
2) Data encryption – This process involves the iteration of a simple function 16 times. Each round contains a key-dependent permutation and key-and data-dependent substitution.

Algorithm:

Step 1: Generation of subkeys:

18 subkeys {P[0]...P[17]} are needed in both encryption as well as decryption process and the same subkeys are used for both the processes.

These 18 subkeys are stored in a P-array with each array element being a 32-bit entry.

It is initialized with the digits of pi(?).

Step 2: Initialize Substitution Boxes:

4 Substitution boxes(S-boxes) are needed{S[0]...S[4]} in both encryption as well as decryption process with each S-box having 256 entries{S[i][0]...S[i][255], 0&lei&le4} where each entry is 32-bit.

It is initialized with the digits of pi(?) after initializing the P-array.

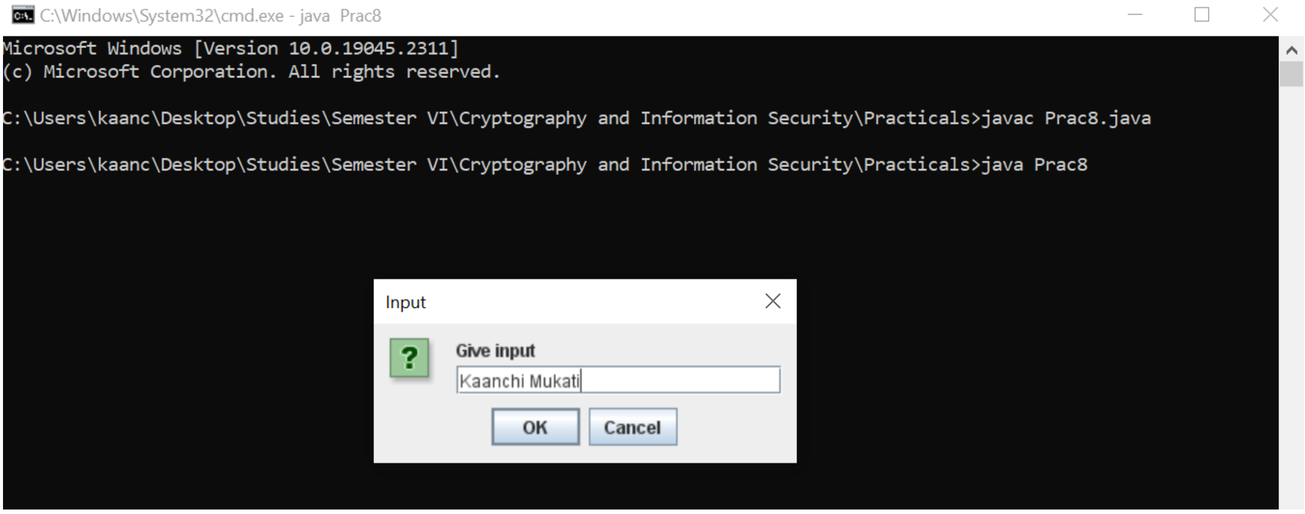
Step 3: Encryption:

a) The encryption function consists of two parts:
   Rounds: The encryption consists of 16 rounds with each round (Ri) taking inputs the plain Text (P.T.) from previous round and corresponding subkey (Pi).
b) Post-processing: The output after the 16 rounds is processed

Code:

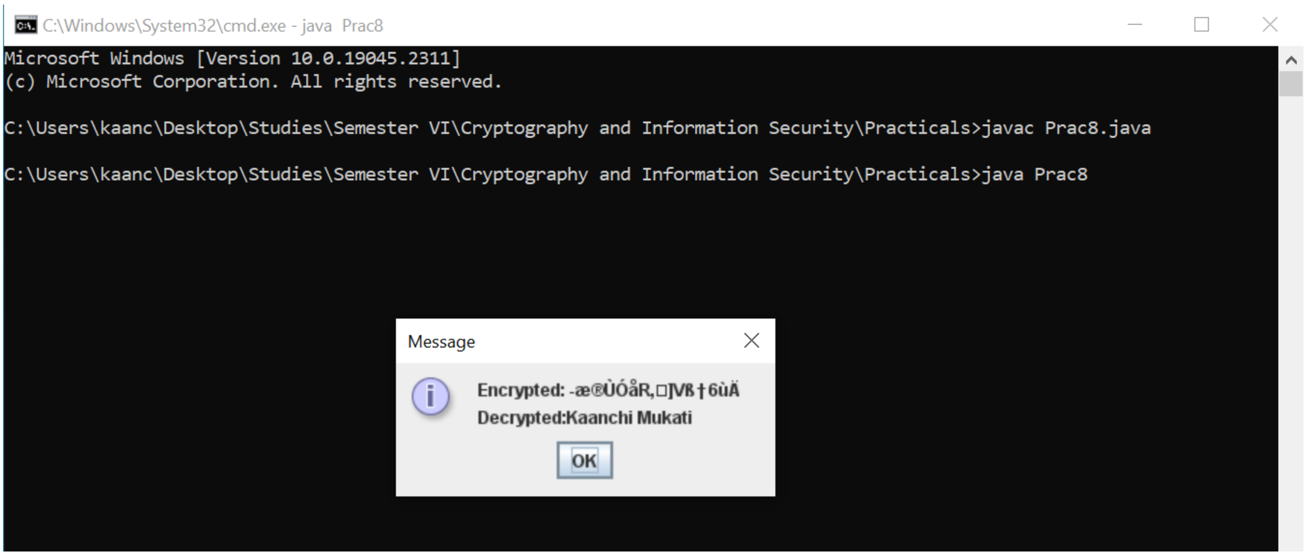
```
import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.swing.JOptionPane;

public class Prac8
{
     public static void main(String args[])
     throws Exception
     {
           KeyGenerator keygen =
KeyGenerator.getInstance("Blowfish");
           SecretKey secretkey = keygen.generateKey();
           Cipher cip = Cipher.getInstance("Blowfish");
           cip.init(Cipher.ENCRYPT_MODE, secretkey);

           String inputText = JOptionPane.showInputDialog("Give
input");
           byte[] encrypted = cip.doFinal(inputText.getBytes());
           cip.init(Cipher.DECRYPT_MODE, secretkey);
           byte[] decrypted = cip.doFinal(encrypted);
           JOptionPane.showMessageDialog(JOptionPane.getRootFrame(),
"Encrypted: " + new String(encrypted) +"\n" + "Decrypted:" + new
String(decrypted));
           System.exit(0);
     }
}
```

Output:



```
Microsoft Windows [Version 10.0.19045.2311]
(c) Microsoft Corporation. All rights reserved.

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>javac Prac8.java

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>java Prac8
```

Kaanchi Mukati



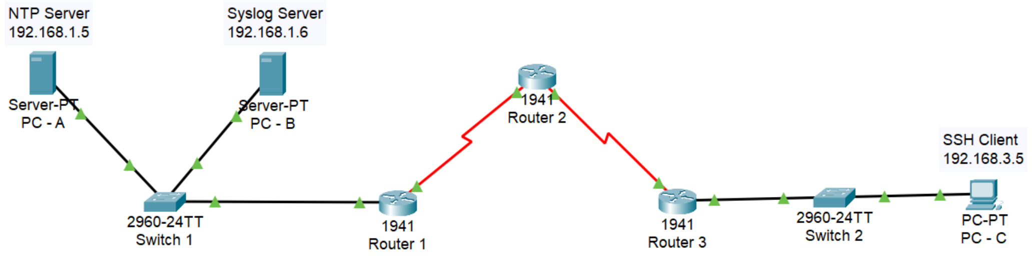
```
Microsoft Windows [Version 10.0.19045.2311]
(c) Microsoft Corporation. All rights reserved.

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>javac Prac8.java

C:\Users\kaanc\Desktop\Studies\Semester VI\Cryptography and Information Security\Practicals>java Prac8
```

Encrypted: -æ®ÙÓåR,□]Vß†6ùÄ
Decrypted:Kaanchi Mukati

Practical 9 – Configure OSPF MD5 authentication and support SSH connections.

A) Topology

1) Drag and drop the required icons that are 3 1941 Routers; 1 PT-PC; 2 2960-24TT Switches; 2 PT-Servers and rename them accordingly.
2) Define the respective IP configuration by going to Desktop and then to IP Configuration for PC - A, PC - B(Server) and PC - C(Server).
3) Add serial ports to all the three routers by going to Physical of Router 1, switching the router off and adding HWIC-2T to both the empty slots and turn the switch on, do the same for Router 2 and Router 3.
4) Connect the devices with the respective wired connection.



B) Configure Router

Router 1

```
Router>en
Router#conf t
Router(config)#host r1

r1(config)#int g0/1
r1(config-if)# ip address 192.168.1.1 255.255.255.0
r1(config-if)#no shut
```

```
r1(config-if)#int s0/0/0
r1(config-if)#ip address 10.1.1.1 255.255.255.252
r1(config-if)#no shut
```

Router 2

```
Router>en
Router#conf t
Router(config)#host r2

r2(config)#int s0/0/0
r2(config-if)#ip address 10.1.1.2 255.255.255.252
r2(config-if)#no shut

r2(config-if)#int s0/0/1
r2(config-if)#ip address 10.2.2.2 255.255.255.252
r2(config-if)#no shut
```

Router 3

```
Router>en
Router#conf t
Router(config)#host r3

r3(config)#int g0/1
r3(config-if)#ip address 192.168.3.1 255.255.255.0
r3(config-if)#no shut

r3(config-if)#int s0/0/1
r3(config-if)#ip address 10.2.2.1 255.255.255.252
r3(config-if)#no shut
```

C) Configure OSPF

Router 1

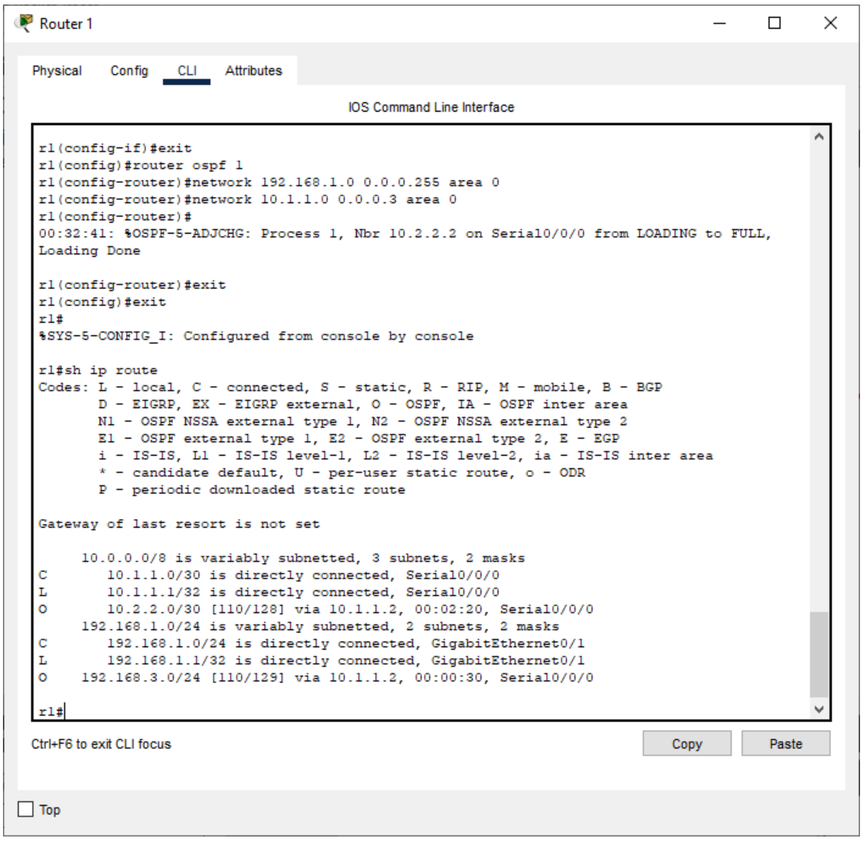
```
r1(config)#router ospf 1
r1(config-router)#network 192.168.1.0 0.0.0.255 area 0
r1(config-router)#network 10.1.1.0 0.0.0.3 area 0
r1(config-router)#exit
r1(config)#exit
r1#sh ip route
```

Router 2

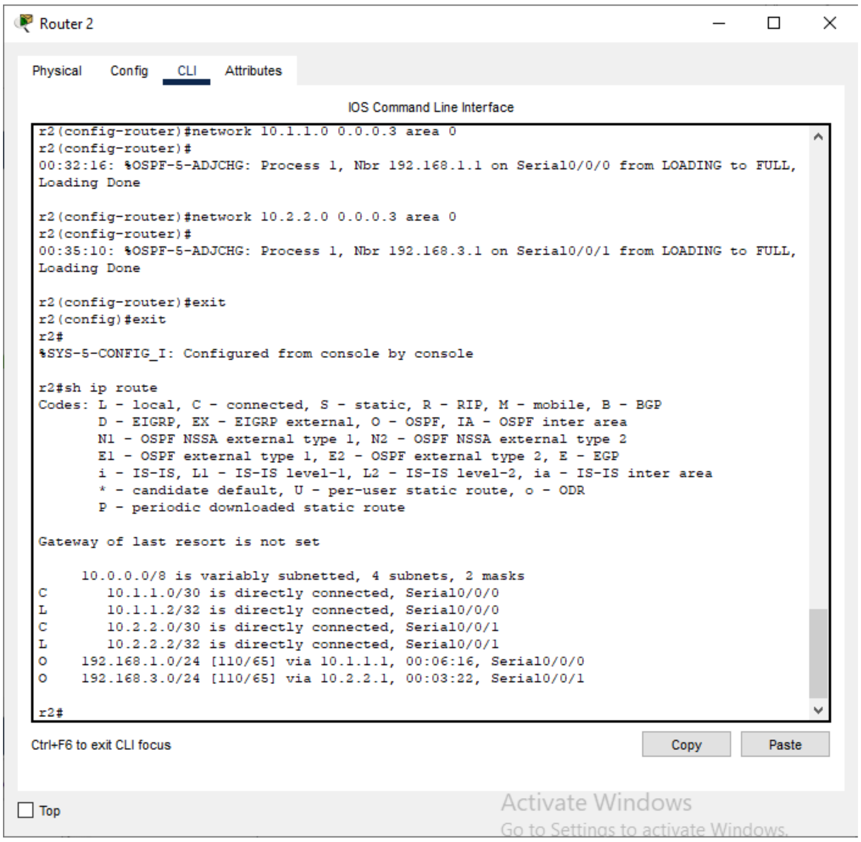
```
r2(config)#router ospf 1
r2(config-router)#network 10.1.1.0 0.0.0.3 area 0
r2(config-router)#network 10.2.2.0 0.0.0.3 area 0
r2(config-router)#exit
r2(config)#exit
r2#sh ip route
```

Router 3

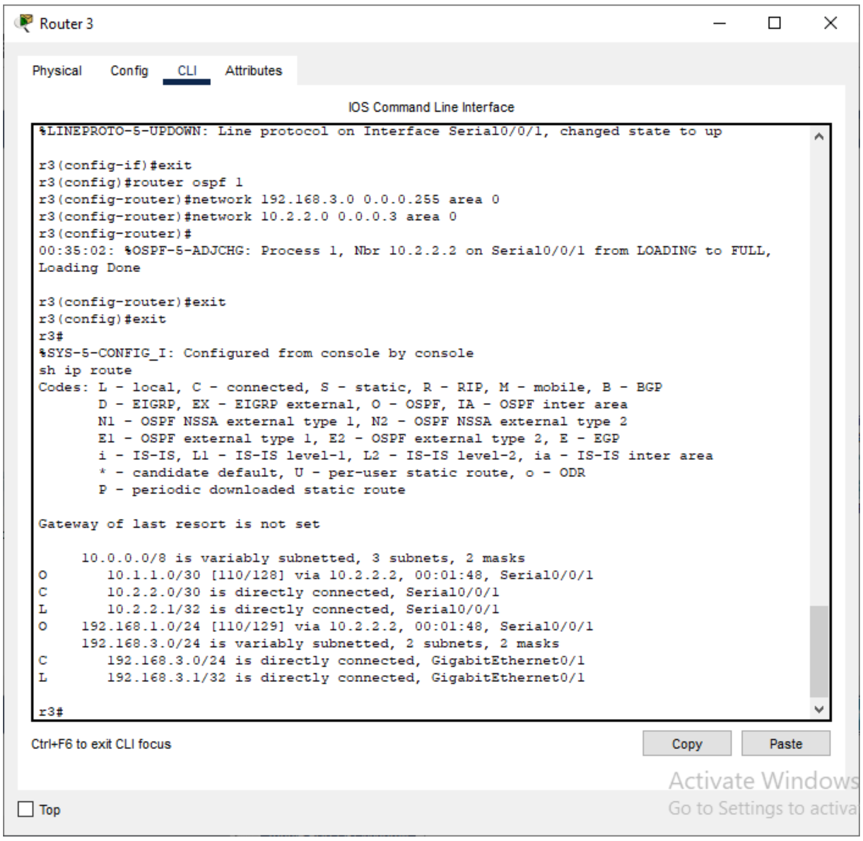
```
r3(config)#router ospf 1
r3(config-router)#network 192.168.3.0 0.0.0.255 area 0
r3(config-router)#network 10.2.2.0 0.0.0.3 area 0
r3(config-router)#exit
r3(config)#exit
r3#sh ip route
```



```
r1(config-if)#exit
r1(config)#router ospf 1
r1(config-router)#network 192.168.1.0 0.0.0.255 area 0
r1(config-router)#network 10.1.1.0 0.0.0.3 area 0
r1(config-router)#
00:32:41: %OSPF-5-ADJCHG: Process 1, Nbr 10.2.2.2 on Serial0/0/0 from LOADING to FULL,
Loading Done

r1(config-router)#exit
r1(config)#exit
r1#
%SYS-5-CONFIG_I: Configured from console by console

r1#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C       10.1.1.0/30 is directly connected, Serial0/0/0
L       10.1.1.1/32 is directly connected, Serial0/0/0
O       10.2.2.0/30 [110/128] via 10.1.1.2, 00:02:20, Serial0/0/0
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, GigabitEthernet0/1
L       192.168.1.1/32 is directly connected, GigabitEthernet0/1
O    192.168.3.0/24 [110/129] via 10.1.1.2, 00:00:30, Serial0/0/0

r1#
```


Router 2
Physical
Config
CLI
Attributes
IOS Command Line Interface
Ctrl+F6 to exit CLI focus
Copy
Paste
□ Top
Activate Windows
Go to Settings to activate Windows.

```
r2(config-router)#network 10.1.1.0 0.0.0.3 area 0
r2(config-router)#
00:32:16: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on Serial0/0/0 from LOADING to FULL,
Loading Done

r2(config-router)#network 10.2.2.0 0.0.0.3 area 0
r2(config-router)#
00:35:10: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.3.1 on Serial0/0/1 from LOADING to FULL,
Loading Done

r2(config-router)#exit
r2(config)#exit
r2#
%SYS-5-CONFIG_I: Configured from console by console

r2#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C       10.1.1.0/30 is directly connected, Serial0/0/0
L       10.1.1.2/32 is directly connected, Serial0/0/0
C       10.2.2.0/30 is directly connected, Serial0/0/1
L       10.2.2.2/32 is directly connected, Serial0/0/1
O    192.168.1.0/24 [110/65] via 10.1.1.1, 00:06:16, Serial0/0/0
O    192.168.3.0/24 [110/65] via 10.2.2.1, 00:03:22, Serial0/0/1

r2#
```



```
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/1, changed state to up

r3(config-if)#exit
r3(config)#router ospf 1
r3(config-router)#network 192.168.3.0 0.0.0.255 area 0
r3(config-router)#network 10.2.2.0 0.0.0.3 area 0
r3(config-router)#
00:35:02: %OSPF-5-ADJCHG: Process 1, Nbr 10.2.2.2 on Serial0/0/1 from LOADING to FULL,
Loading Done

r3(config-router)#exit
r3(config)#exit
r3#
%SYS-5-CONFIG_I: Configured from console by console
sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
O       10.1.1.0/30 [110/128] via 10.2.2.2, 00:01:48, Serial0/0/1
C       10.2.2.0/30 is directly connected, Serial0/0/1
L       10.2.2.1/32 is directly connected, Serial0/0/1
O    192.168.1.0/24 [110/129] via 10.2.2.2, 00:01:48, Serial0/0/1
     192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.3.0/24 is directly connected, GigabitEthernet0/1
L       192.168.3.1/32 is directly connected, GigabitEthernet0/1

r3#
```

The O proves that OSPF was successful.

D) Configure MD5 Authentication

Router 1

r1(config)#router ospf 1
r1(config-router)#int s0/0/0
r1(config-router)#ip ospf message-digest-key 1 md5 ty22
r1(config-router)#exit
r1(config)#do sh ip ospf int

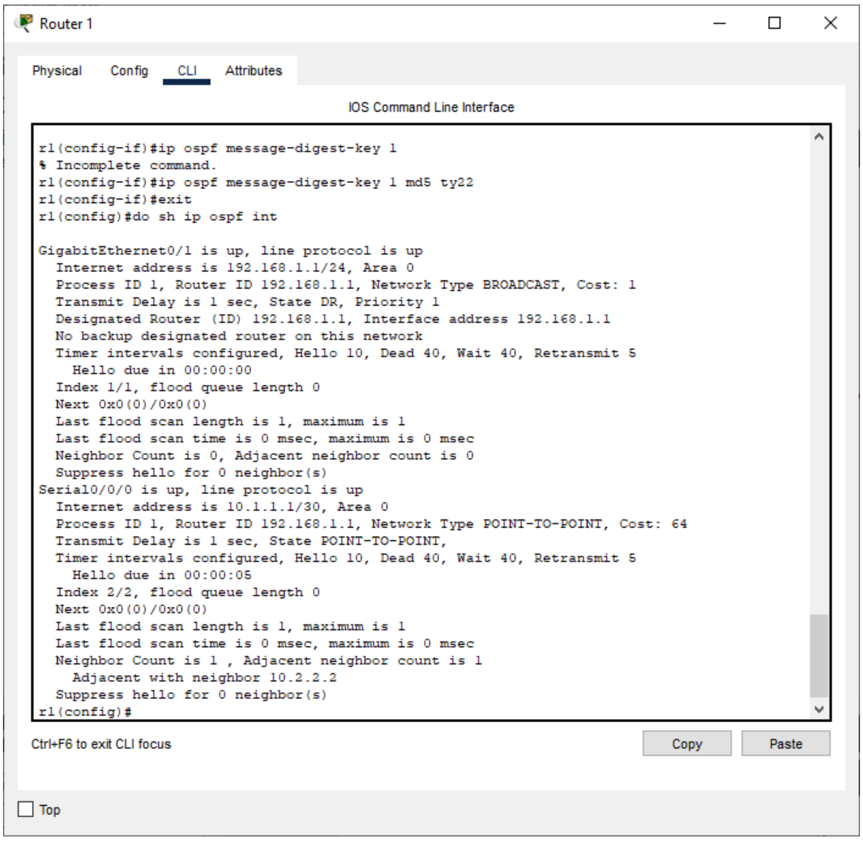
Router 2

r2(config)#router ospf 1
r2(config-router)#int s0/0/0

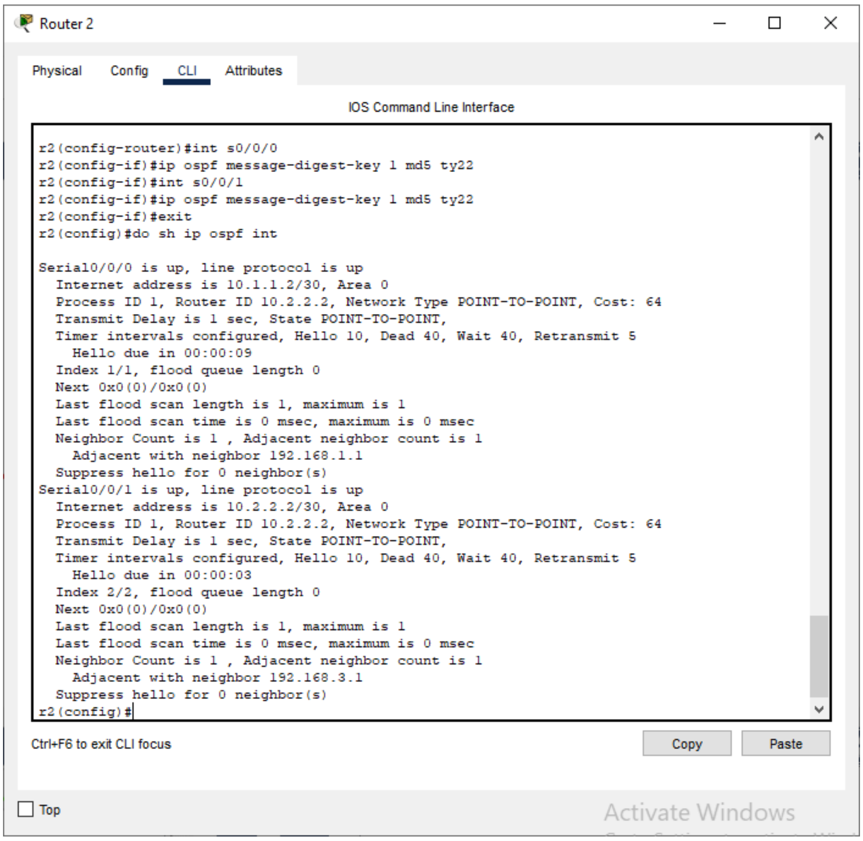
```
r2(config-router)#ip ospf message-digest-key 1 md5 ty22
r2(config-router)#int s0/0/1
r2(config-router)#ip ospf message-digest-key 1 md5 ty22
r2(config-router)#exit
r2(config)#do sh ip ospf int
```

Router 3

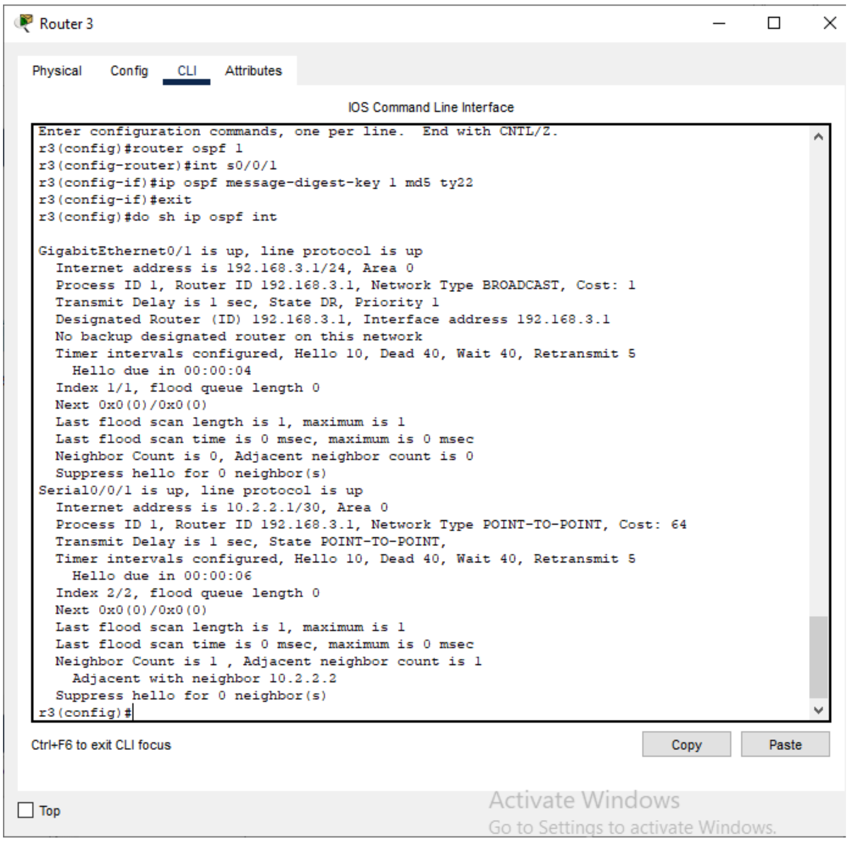
```
r3(config)#router ospf 1
r3(config-router)#int s0/0/1
r3(config-router)#ip ospf message-digest-key 1 md5 ty22
r3(config-router)#exit
r3(config)#do sh ip ospf int
```



```
r1(config-if)#ip ospf message-digest-key 1
% Incomplete command.
r1(config-if)#ip ospf message-digest-key 1 md5 ty22
r1(config-if)#exit
r1(config)#do sh ip ospf int

GigabitEthernet0/1 is up, line protocol is up
  Internet address is 192.168.1.1/24, Area 0
  Process ID 1, Router ID 192.168.1.1, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 192.168.1.1, Interface address 192.168.1.1
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:00
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 0, Adjacent neighbor count is 0
  Suppress hello for 0 neighbor(s)
Serial0/0/0 is up, line protocol is up
  Internet address is 10.1.1.1/30, Area 0
  Process ID 1, Router ID 192.168.1.1, Network Type POINT-TO-POINT, Cost: 64
  Transmit Delay is 1 sec, State POINT-TO-POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:05
  Index 2/2, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1 , Adjacent neighbor count is 1
    Adjacent with neighbor 10.2.2.2
  Suppress hello for 0 neighbor(s)
r1(config)#
```



```
r2(config-router)#int s0/0/0
r2(config-if)#ip ospf message-digest-key 1 md5 ty22
r2(config-if)#int s0/0/1
r2(config-if)#ip ospf message-digest-key 1 md5 ty22
r2(config-if)#exit
r2(config)#do sh ip ospf int

Serial0/0/0 is up, line protocol is up
  Internet address is 10.1.1.2/30, Area 0
  Process ID 1, Router ID 10.2.2.2, Network Type POINT-TO-POINT, Cost: 64
  Transmit Delay is 1 sec, State POINT-TO-POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:09
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1 , Adjacent neighbor count is 1
    Adjacent with neighbor 192.168.1.1
  Suppress hello for 0 neighbor(s)
Serial0/0/1 is up, line protocol is up
  Internet address is 10.2.2.2/30, Area 0
  Process ID 1, Router ID 10.2.2.2, Network Type POINT-TO-POINT, Cost: 64
  Transmit Delay is 1 sec, State POINT-TO-POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:03
  Index 2/2, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1 , Adjacent neighbor count is 1
    Adjacent with neighbor 192.168.3.1
  Suppress hello for 0 neighbor(s)
r2(config)#
```


Router 3
Physical
Config
CLI
Attributes
IOS Command Line Interface
Ctrl+F6 to exit CLI focus
Copy
Paste
□ Top
Activate Windows
Go to Settings to activate Windows.

```
Enter configuration commands, one per line.  End with CNTL/Z.
r3(config)#router ospf 1
r3(config-router)#int s0/0/1
r3(config-if)#ip ospf message-digest-key 1 md5 ty22
r3(config-if)#exit
r3(config)#do sh ip ospf int

GigabitEthernet0/1 is up, line protocol is up
  Internet address is 192.168.3.1/24, Area 0
  Process ID 1, Router ID 192.168.3.1, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 192.168.3.1, Interface address 192.168.3.1
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:04
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 0, Adjacent neighbor count is 0
  Suppress hello for 0 neighbor(s)
Serial0/0/1 is up, line protocol is up
  Internet address is 10.2.2.1/30, Area 0
  Process ID 1, Router ID 192.168.3.1, Network Type POINT-TO-POINT, Cost: 64
  Transmit Delay is 1 sec, State POINT-TO-POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:06
  Index 2/2, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1 , Adjacent neighbor count is 1
    Adjacent with neighbor 10.2.2.2
  Suppress hello for 0 neighbor(s)
r3(config)#
```

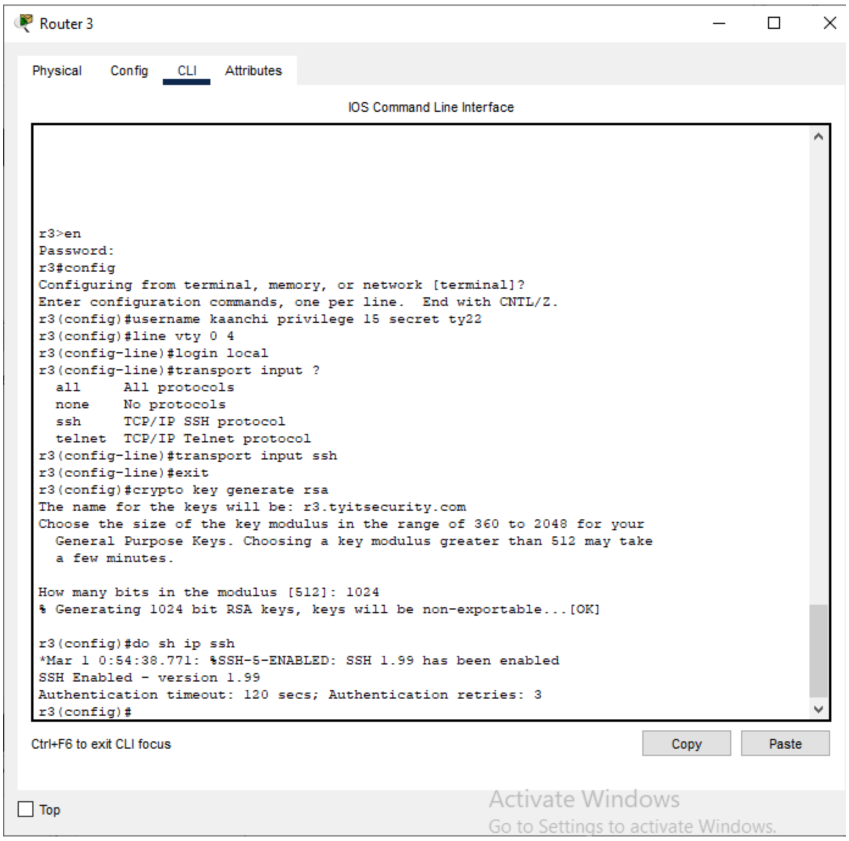
E) Configure Router 3 to support SSH connections

r3(config)#ip domain-name tyitsecurity.com

r3(config)#enable secret ty22

Exit the connections and enable the router again. It will ask for password, enter ty22, it will not be visible but the system will accept it. Once you press enter, it will take you to config mode of the router.

r3(config)#username kaanchi privilege 15 secret ty22
r3(config)#line vty 0 4
r3(config-line)#login local
r3(config-line)#transport input ssh
r3(config-line)#exit
r3(config)#crypto key generate rsa
How many bits in the modulus? 1024
r3(config)#do sh ip ssh


Router 3
Physical
Config
CLI
Attributes
IOS Command Line Interface
Ctrl+F6 to exit CLI focus
Copy
Paste
□ Top
Activate Windows
Go to Settings to activate Windows.

```
r3>en
Password:
r3#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
r3(config)#username kaanchi privilege 15 secret ty22
r3(config)#line vty 0 4
r3(config-line)#login local
r3(config-line)#transport input ?
  all     All protocols
  none    No protocols
  ssh     TCP/IP SSH protocol
  telnet  TCP/IP Telnet protocol
r3(config-line)#transport input ssh
r3(config-line)#exit
r3(config)#crypto key generate rsa
The name for the keys will be: r3.tyitsecurity.com
Choose the size of the key modulus in the range of 360 to 2048 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

r3(config)#do sh ip ssh
*Mar 1 0:54:38.771: %SSH-5-ENABLED: SSH 1.99 has been enabled
SSH Enabled - version 1.99
Authentication timeout: 120 secs; Authentication retries: 3
r3(config)#
```

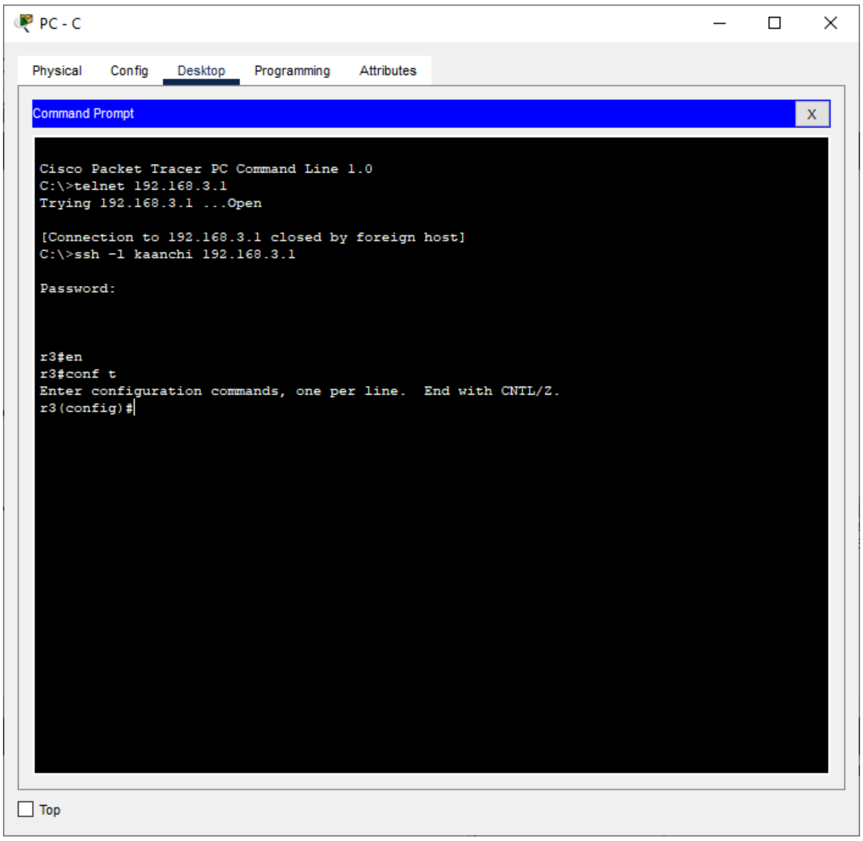
F) Check results

Open Command Prompt of PC – C.

Enter telnet 192.168.3.1, the connection will open.

Enter ssh -l kaanchi 192.168.3.1

Password will be asked type ty22 which will not be visible but will be accepted by the system. Router 3 configuration is now accessible from PC - C.



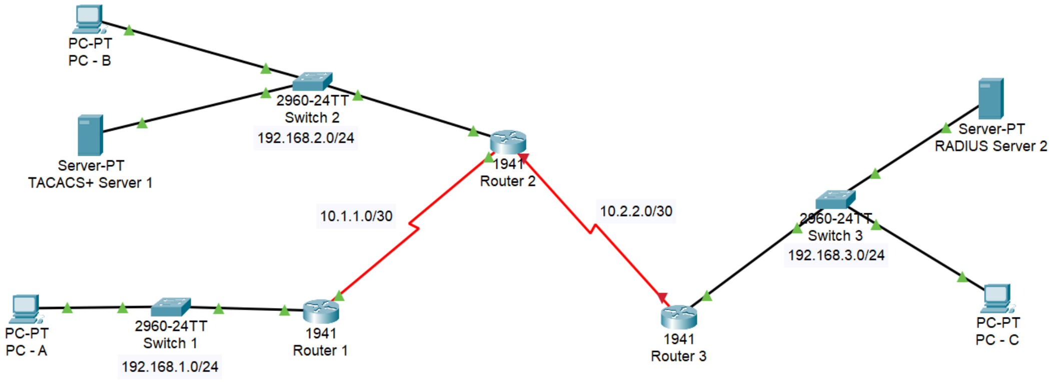
```
Cisco Packet Tracer PC Command Line 1.0
C:\>telnet 192.168.3.1
Trying 192.168.3.1 ...Open

[Connection to 192.168.3.1 closed by foreign host]
C:\>ssh -l kaanchi 192.168.3.1

Password:



r3#en
r3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
r3(config)#
```

Practical 10 – Configure AAA authentication on routers.

A) Topology

Drag and drop the required icons; rename them and change the IP configuration and then connect them.

5) Drag and drop the required icons that are 3 1941 Routers; 3 PT-PCs'; 3 2960-24TT Switches; 2 PT-Servers and rename them accordingly.
6) Define the respective IP configuration by going to Desktop and then to IP Configuration for PC - A, PC – B, PC – C, TACACS+ Server and RADIUS Server.
7) Add serial ports to all the three routers by going to Physical of Router 1, switching the router off and adding HWIC-2T to both the empty slots and turn the switch on, do the same for Router 2 and Router 3.
8) Connect the devices with the respective wired connection.



B) Configure Router

Router 1

```
Router>en
Router#conf t
Router(config)#host r1
```

```
r1(config)#int g0/1
r1(config-if)# ip address 192.168.1.1 255.255.255.0
r1(config-if)#no shut

r1(config-if)#int s0/0/0
r1(config-if)#ip address 10.1.1.2 255.255.255.252
r1(config-if)#no shut
```

Router 2

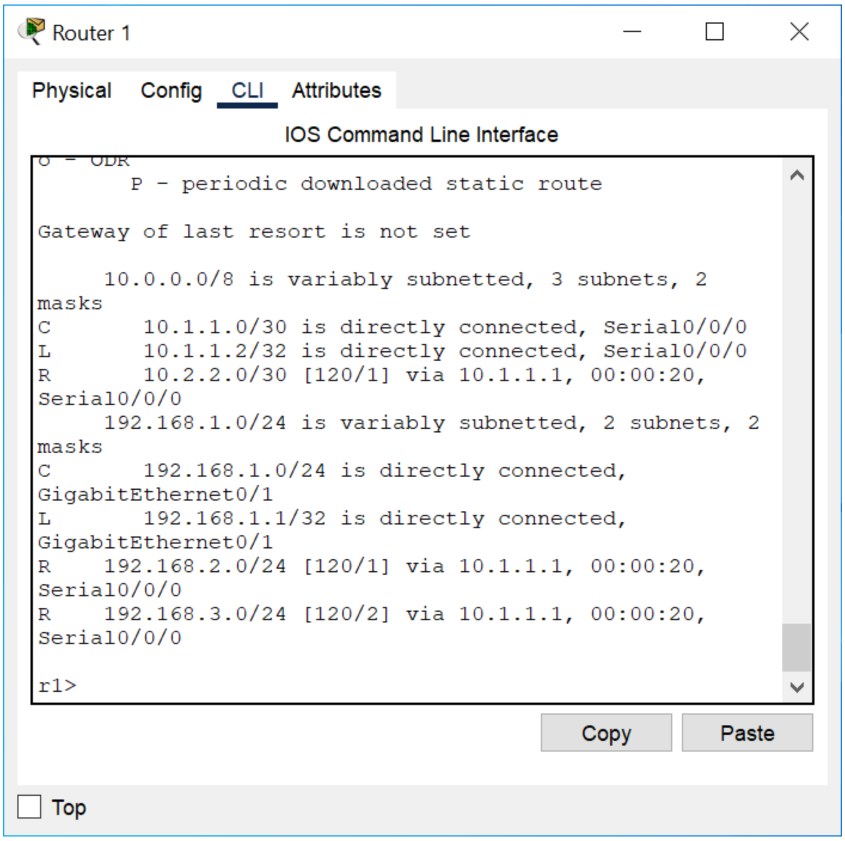
```
Router>en
Router#conf t
Router(config)#host r2

r2(config)#int g0/0
r2(config-if)#ip address 192.168.2.1 255.255.255.0
r2(config-if)#no shut

r2(config-if)#int s0/0/0
r2(config-if)#ip address 10.1.1.1 255.255.255.252
r2(config-if)#no shut

r2(config-if)#int s0/0/1
r2(config-if)#ip address 10.2.2.1 255.255.255.252
r2(config-if)#no shut
```

Router 3

```
Router>en
Router#conf t
Router(config)#host r3

r3(config)#int g0/1
r3(config-if)#ip address 192.168.3.1 255.255.255.0
r3(config-if)#no shut

r3(config-if)#int s0/0/1
r3(config-if)#ip address 10.2.2.2 255.255.255.252
r3(config-if)#no shut
```

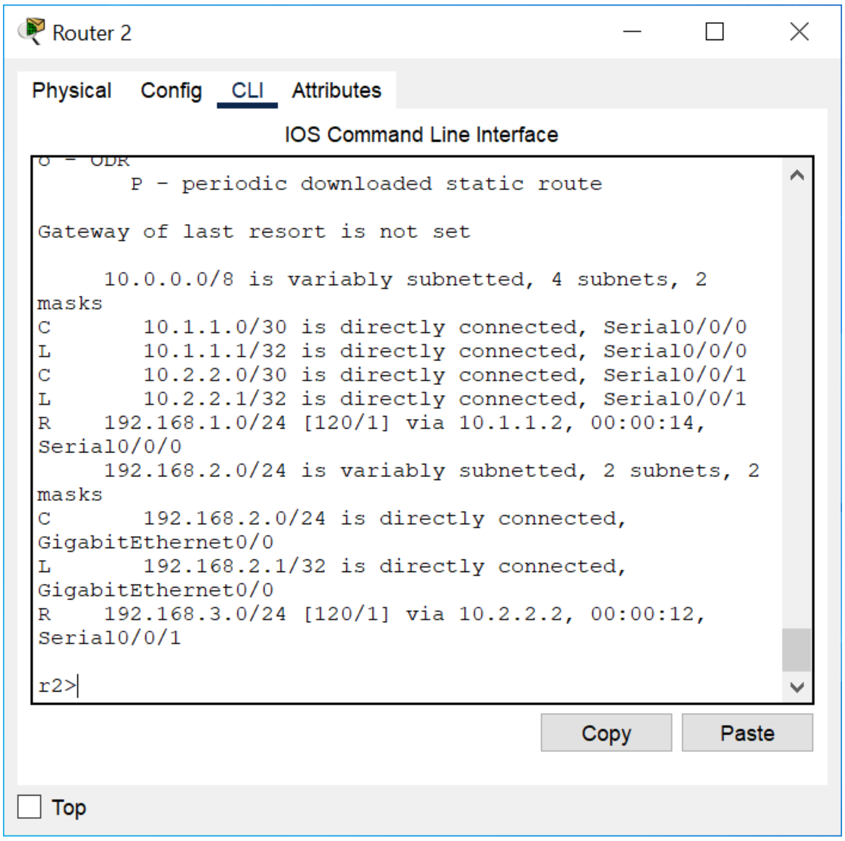
C) Configure RIP on routers

Router 1
r1(config)#router rip
r1(config-router)#network 192.168.1.0
r1(config-router)#network 10.1.1.0
r1(config-router)#exit
r1>sh ip route



```
o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 3 subnets, 2
masks
C       10.1.1.0/30 is directly connected, Serial0/0/0
L       10.1.1.2/32 is directly connected, Serial0/0/0
R       10.2.2.0/30 [120/1] via 10.1.1.1, 00:00:20,
Serial0/0/0
     192.168.1.0/24 is variably subnetted, 2 subnets, 2
masks
C       192.168.1.0/24 is directly connected,
GigabitEthernet0/1
L       192.168.1.1/32 is directly connected,
GigabitEthernet0/1
R    192.168.2.0/24 [120/1] via 10.1.1.1, 00:00:20,
Serial0/0/0
R    192.168.3.0/24 [120/2] via 10.1.1.1, 00:00:20,
Serial0/0/0

r1>
```

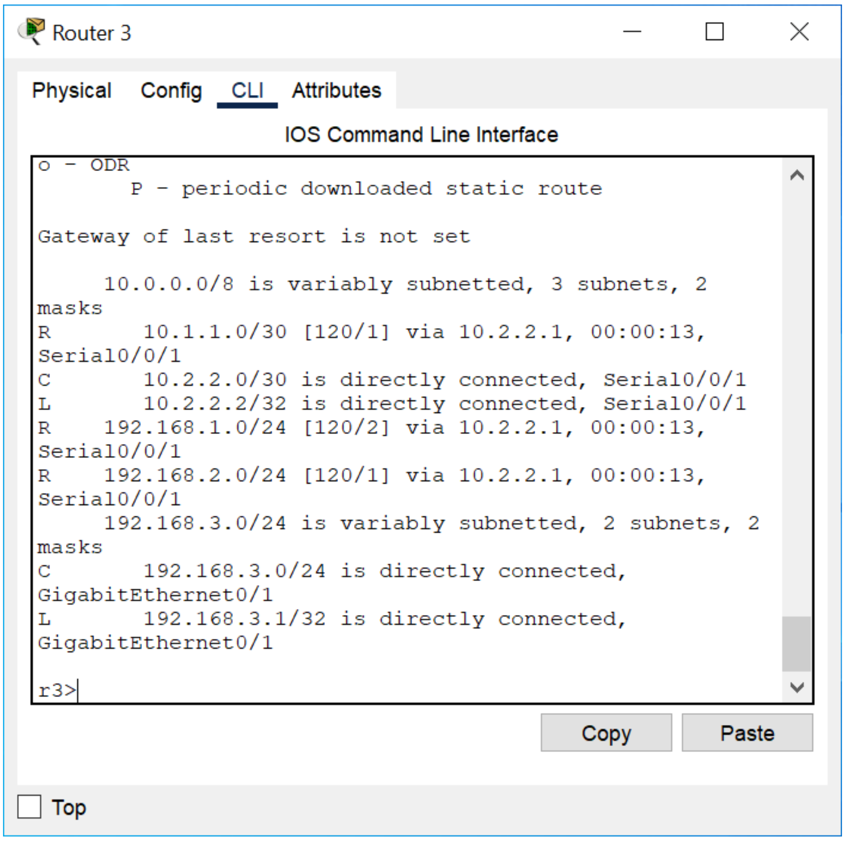
Router 2
r2(config)#router rip
r2(config-router)#network 10.1.1.0
r2(config-router)#network 192.168.2.0
r2(config-router)#network 10.2.2.0
r2(config-router)#exit
r2>sh ip route



```
o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 4 subnets, 2
masks
C       10.1.1.0/30 is directly connected, Serial0/0/0
L       10.1.1.1/32 is directly connected, Serial0/0/0
C       10.2.2.0/30 is directly connected, Serial0/0/1
L       10.2.2.1/32 is directly connected, Serial0/0/1
R    192.168.1.0/24 [120/1] via 10.1.1.2, 00:00:14,
Serial0/0/0
     192.168.2.0/24 is variably subnetted, 2 subnets, 2
masks
C       192.168.2.0/24 is directly connected,
GigabitEthernet0/0
L       192.168.2.1/32 is directly connected,
GigabitEthernet0/0
R    192.168.3.0/24 [120/1] via 10.2.2.2, 00:00:12,
Serial0/0/1

r2>
```

Router 3

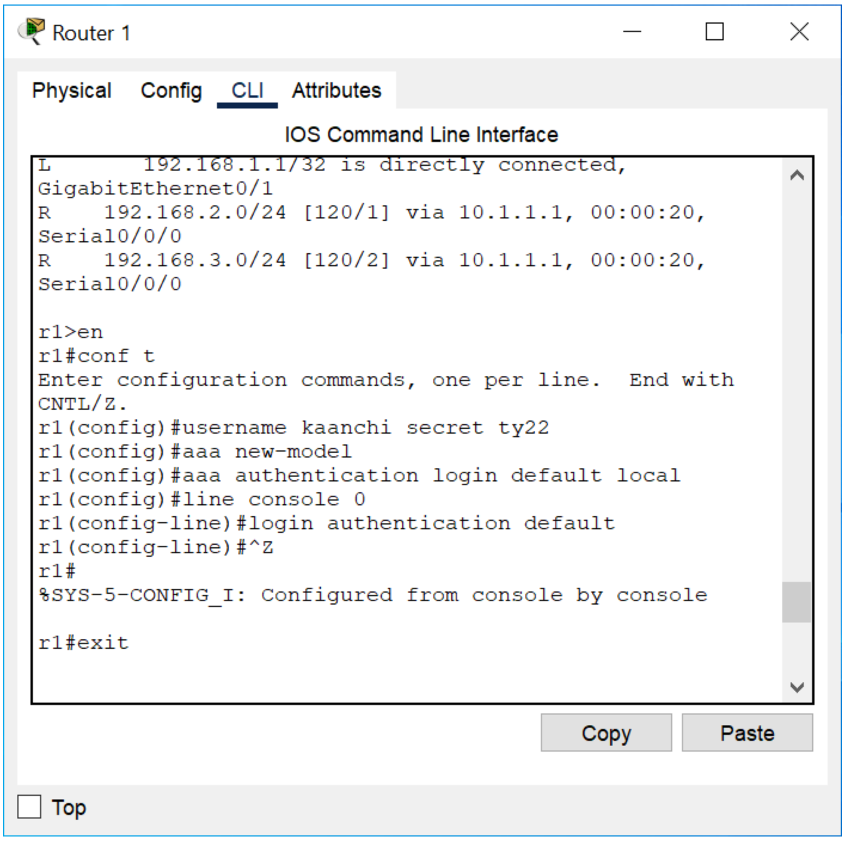
```
r3(config)#router rip
r3(config-router)#network 192.168.3.0
r3(config-router)#network 10.2.2.0
r3(config-router)#exit
r3>sh ip route
```



```
o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 3 subnets, 2
masks
R       10.1.1.0/30 [120/1] via 10.2.2.1, 00:00:13,
Serial0/0/1
C       10.2.2.0/30 is directly connected, Serial0/0/1
L       10.2.2.2/32 is directly connected, Serial0/0/1
R    192.168.1.0/24 [120/2] via 10.2.2.1, 00:00:13,
Serial0/0/1
R    192.168.2.0/24 [120/1] via 10.2.2.1, 00:00:13,
Serial0/0/1
     192.168.3.0/24 is variably subnetted, 2 subnets, 2
masks
C       192.168.3.0/24 is directly connected,
GigabitEthernet0/1
L       192.168.3.1/32 is directly connected,
GigabitEthernet0/1

r3>
```

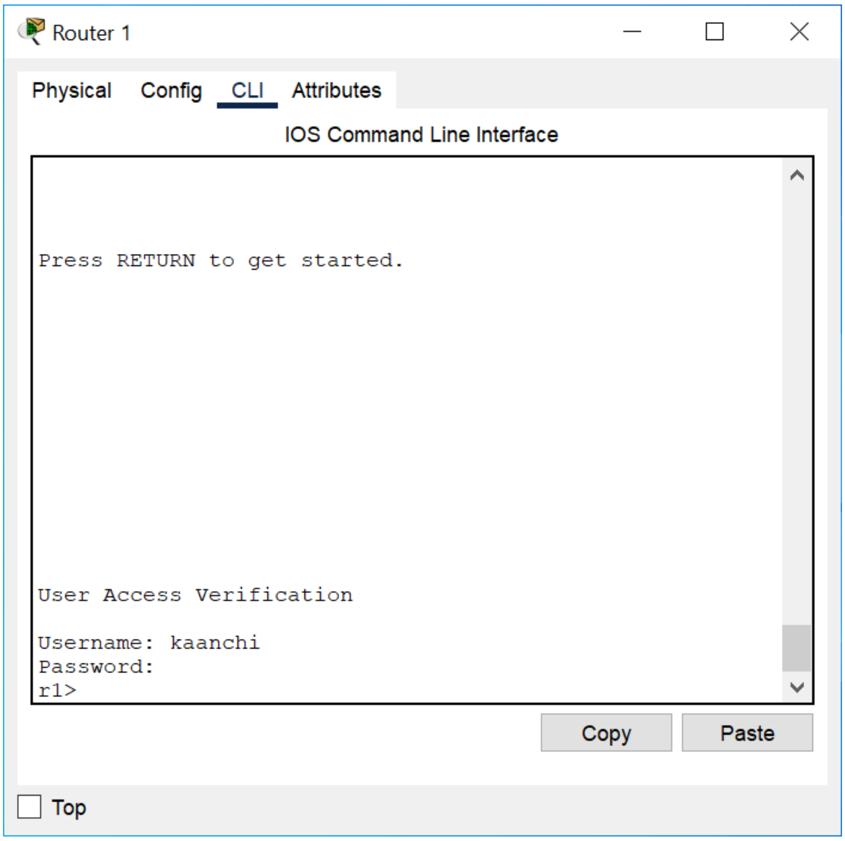
The R proves that RIP was successful. (Routing Information Protocol)

D) Configure local AAA authentication on Router 1



```
L        192.168.1.1/32 is directly connected,
GigabitEthernet0/1
R    192.168.2.0/24 [120/1] via 10.1.1.1, 00:00:20,
Serial0/0/0
R    192.168.3.0/24 [120/2] via 10.1.1.1, 00:00:20,
Serial0/0/0

r1>en
r1#conf t
Enter configuration commands, one per line.  End with
CNTL/Z.
r1(config)#username kaanchi secret ty22
r1(config)#aaa new-model
r1(config)#aaa authentication login default local
r1(config)#line console 0
r1(config-line)#login authentication default
r1(config-line)#^Z
r1#
%SYS-5-CONFIG_I: Configured from console by console

r1#exit
```

```
r1>en
r1#conf t
r1(config)#username kaanchi secret ty22
r1(config)#aaa new-model
r1(config)#aaa authentication login default local
r1(config)#line console 0
r1(config-line)#login authentication default
```

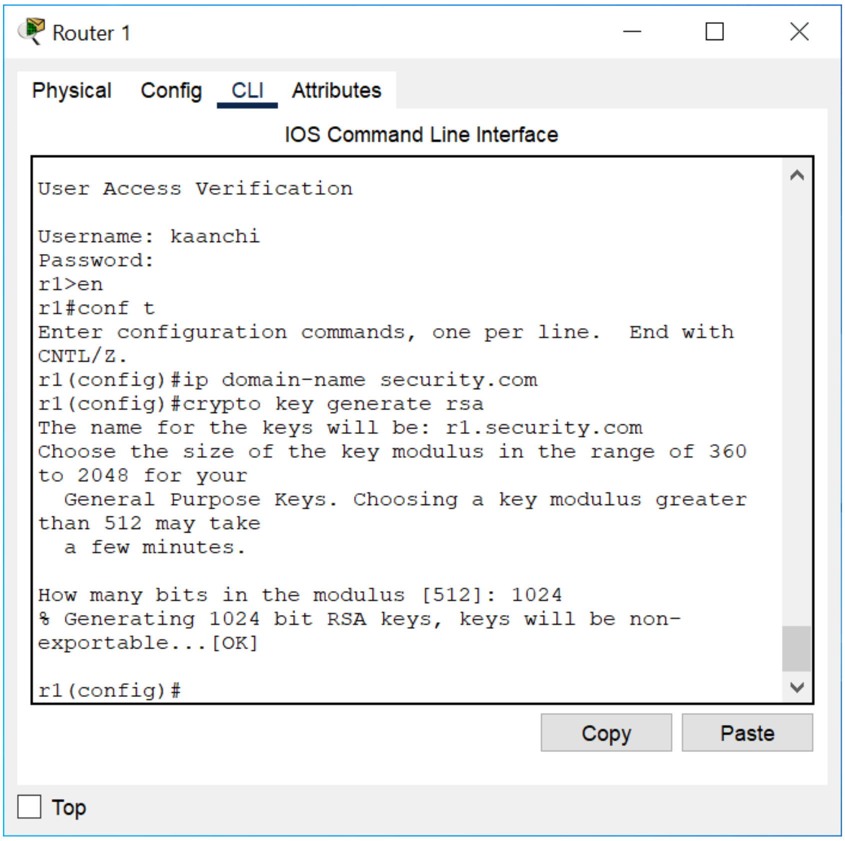


```
Press RETURN to get started.




User Access Verification

Username: kaanchi
Password:
r1>
```

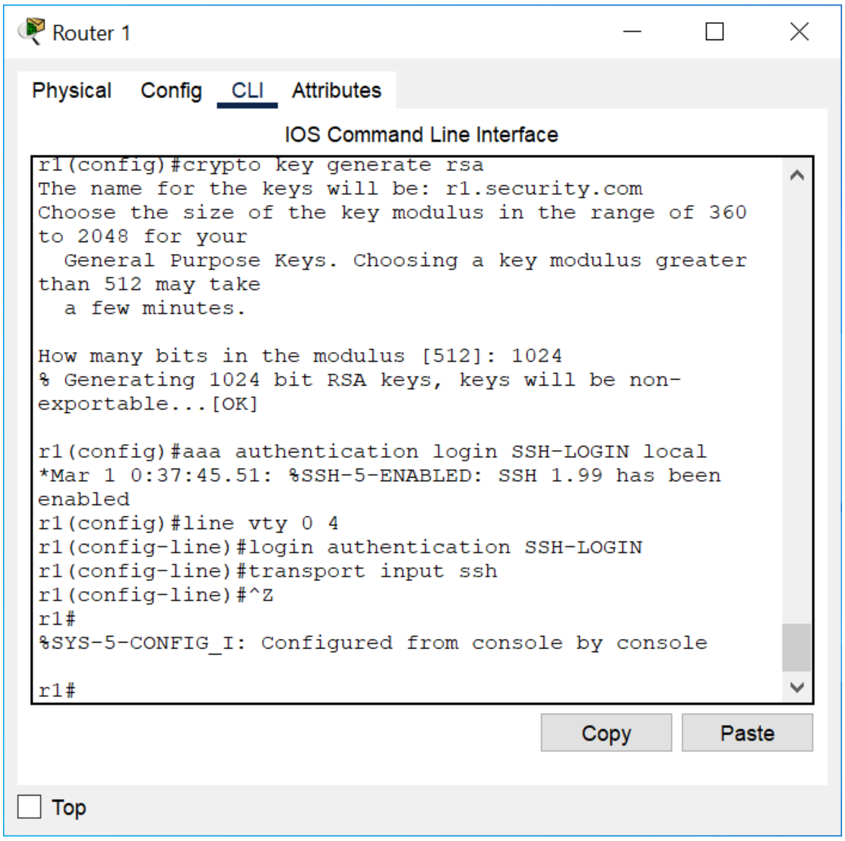
Username is Kaanchi and password is ty22.

E) Configure local AAA authentication for vty lines on Router 1

From security.com it is



```
User Access Verification

Username: kaanchi
Password:
r1>en
r1#conf t
Enter configuration commands, one per line.  End with
CNTL/Z.
r1(config)#ip domain-name security.com
r1(config)#crypto key generate rsa
The name for the keys will be: r1.security.com
Choose the size of the key modulus in the range of 360
to 2048 for your
  General Purpose Keys. Choosing a key modulus greater
than 512 may take
  a few minutes.

How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-
exportable...[OK]

r1(config)#
```



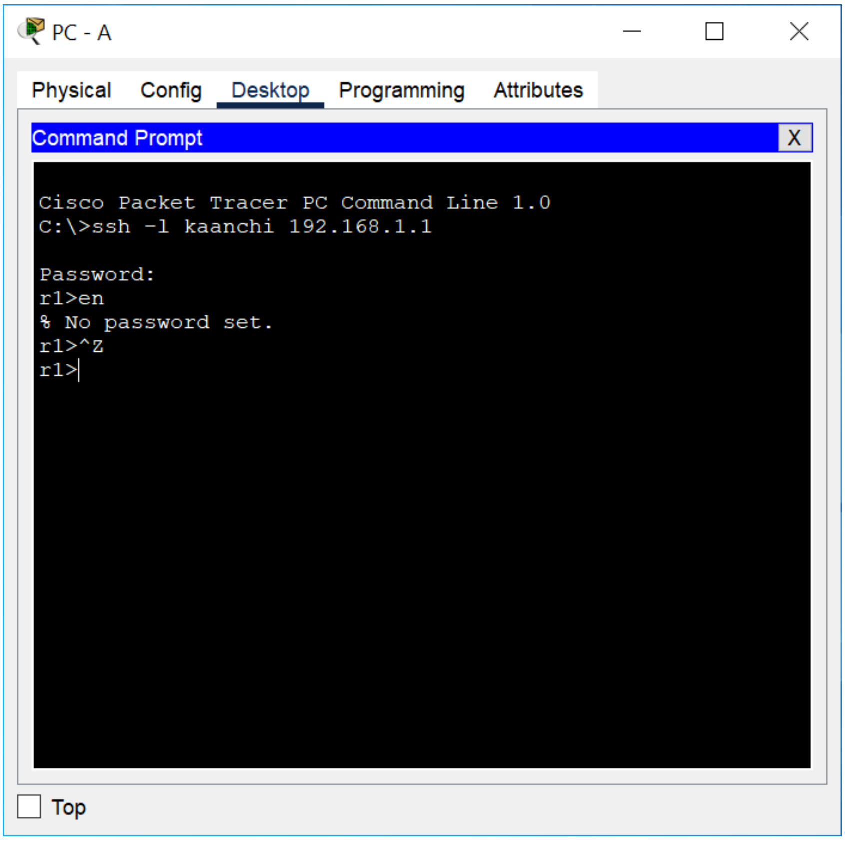
```
r1(config)#crypto key generate rsa
The name for the keys will be: r1.security.com
Choose the size of the key modulus in the range of 360
to 2048 for your
  General Purpose Keys. Choosing a key modulus greater
than 512 may take
  a few minutes.

How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-
exportable...[OK]

r1(config)#aaa authentication login SSH-LOGIN local
*Mar 1 0:37:45.51: %SSH-5-ENABLED: SSH 1.99 has been
enabled
r1(config)#line vty 0 4
r1(config-line)#login authentication SSH-LOGIN
r1(config-line)#transport input ssh
r1(config-line)#^Z
r1#
%SYS-5-CONFIG_I: Configured from console by console

r1#
```

```
r1>en
r1#conf t
r1(config)#ip domain-name security.com
r1(config)#crypto key generate rsa
```

How many bits in the modulus will be asked, enter 1024.

```
r1(config)#aaa authentication login SSH-LOGIN local
r1(config)#line vty 0 4
r1(config-line)#login authentication SSH-LOGIN
r1(config-line)#transport input ssh
r1(config-line)#exit
```

Go to PC – A Command Prompt and enter ssh -l Kaanchi 192.168.1.1. It will ask for password, enter ty22.

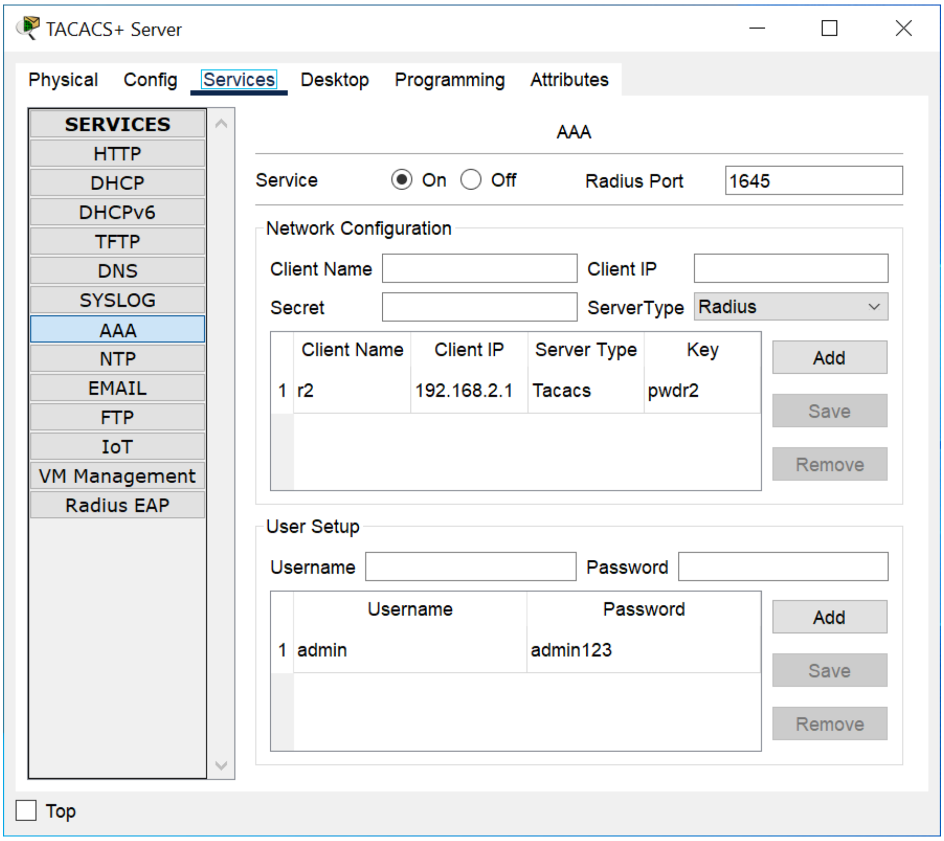


```
Cisco Packet Tracer PC Command Line 1.0
C:\>ssh -l kaanchi 192.168.1.1

Password:
r1>en
% No password set.
r1>^Z
r1>
```

add a black text box over the no password set.

F) Configure service-based AAA authentication using TACACS+ on Router 2

Go to services of TACACS+ Server and then select AAA. Switch on the service and enter the client IP and password. Setup username and password as well.



AAA

Service ● On ○ Off Radius Port 1645

Network Configuration

Client Name Client IP

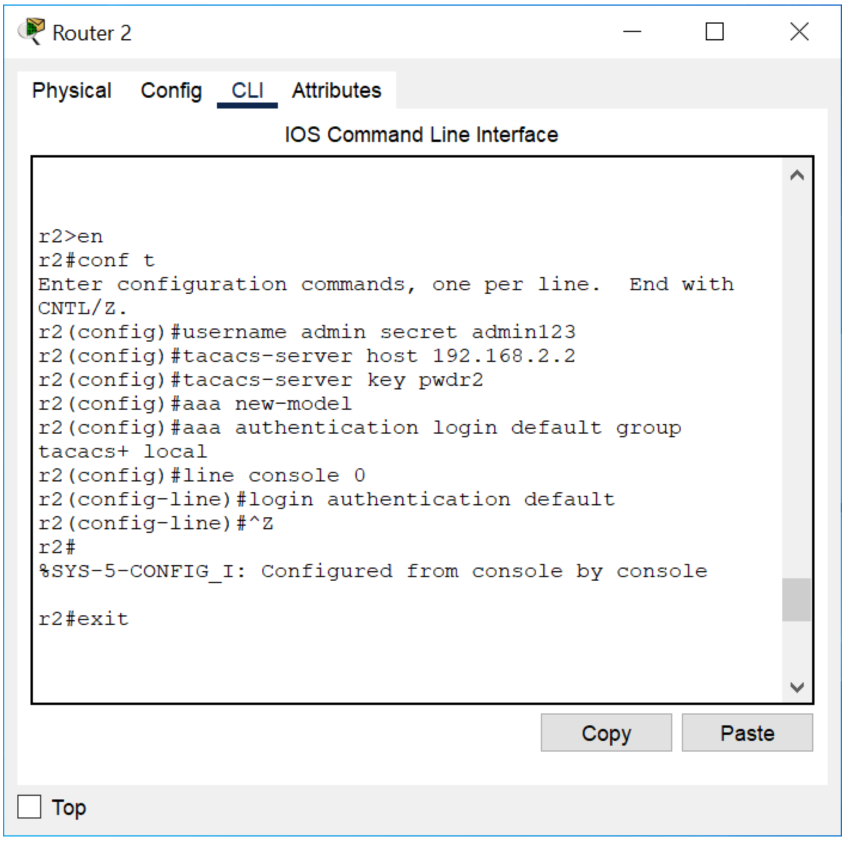
Secret ServerType Radius

	Client Name	Client IP	Server Type	Key
1	r2	192.168.2.1	Tacacs	pwdr2

User Setup

Username Password

	Username	Password
1	admin	admin123

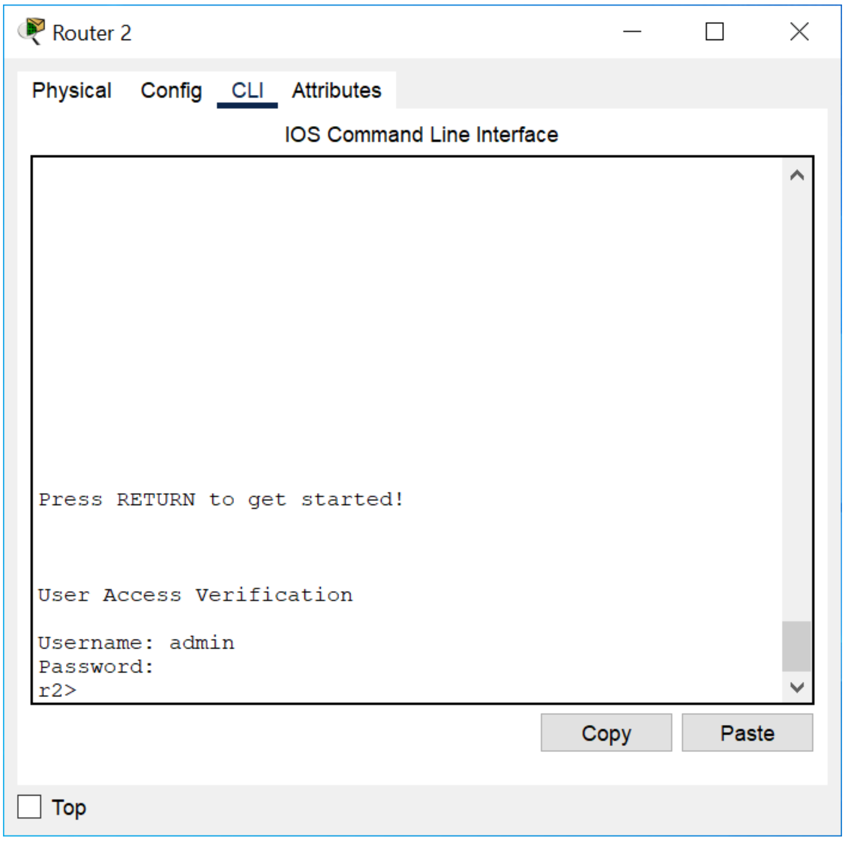


```
r2>en
r2#conf t
Enter configuration commands, one per line.  End with
CNTL/Z.
r2(config)#username admin secret admin123
r2(config)#tacacs-server host 192.168.2.2
r2(config)#tacacs-server key pwdr2
r2(config)#aaa new-model
r2(config)#aaa authentication login default group
tacacs+ local
r2(config)#line console 0
r2(config-line)#login authentication default
r2(config-line)#^Z
r2#
%SYS-5-CONFIG_I: Configured from console by console

r2#exit
```

WRITE FROM HERE!

```
r2>en
r2#conf t
r2(config)#username admin secret admin123
r2(config)#tacacs-server host 192.168.2.2
r2(config)#tacacs-server key pwdr2
r2(config)#aaa new-model
r2(config)#aaa authentication login default group tacacs+ local
r2(config)#line console 0
r2(config-line)#login authentication default
```

User Access Verification is asked enter the credentials of the TACACS+ Server.

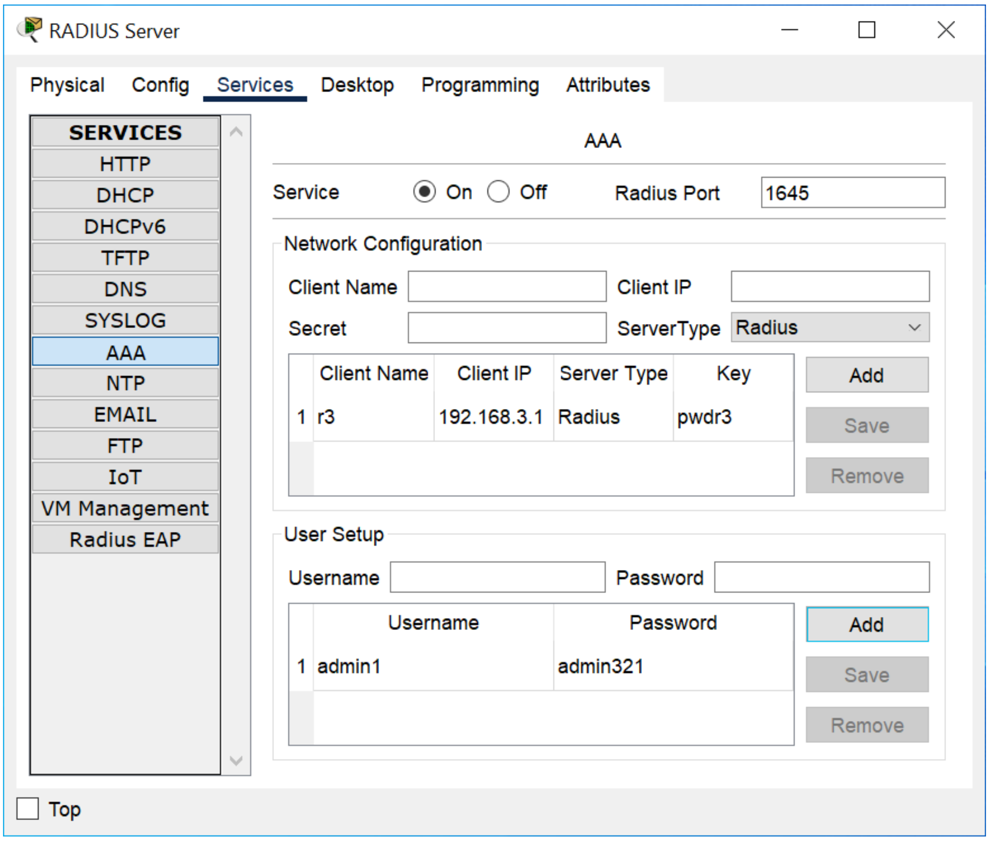


```
Press RETURN to get started!


User Access Verification

Username: admin
Password:
r2>
```

Username is admin and password is admin123 which is also the username and password set for TACACS Server.

G) Configure server-based AAA authentication using RADIUS on Router 3

Go to services of RADIUS Server and then select AAA. Switch on the service and enter the client IP and password. Setup username and password as well.



AAA

Service ● On ○ Off Radius Port 1645

Network Configuration

Client Name: | Client IP:

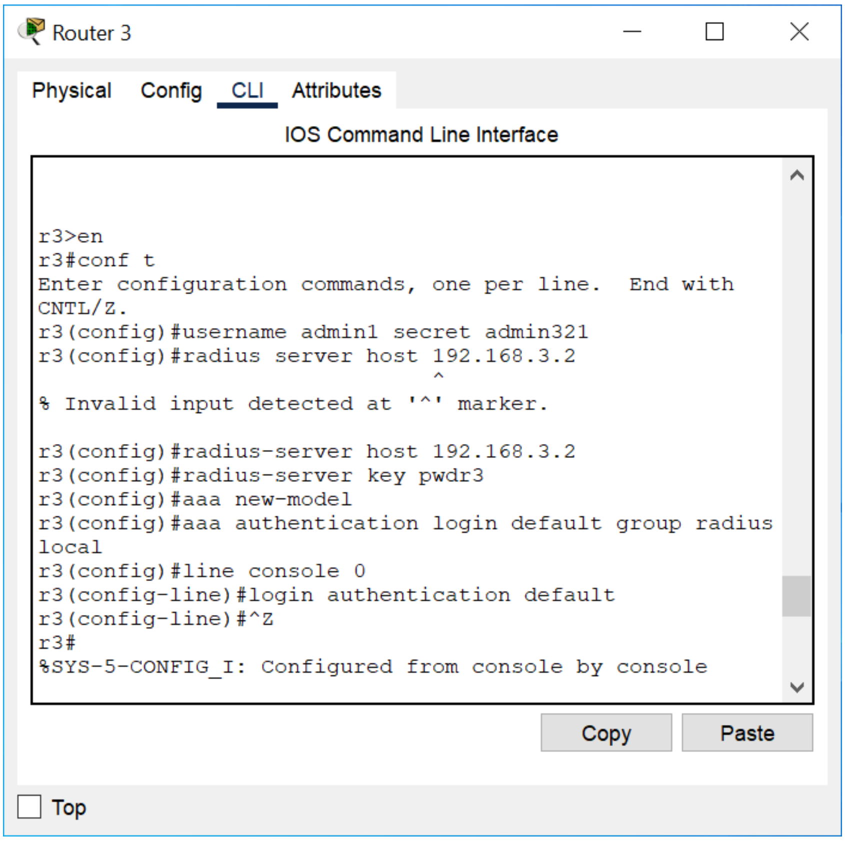
Secret: | ServerType: Radius

	Client Name	Client IP	Server Type	Key
1	r3	192.168.3.1	Radius	pwdr3

User Setup

Username: | Password:

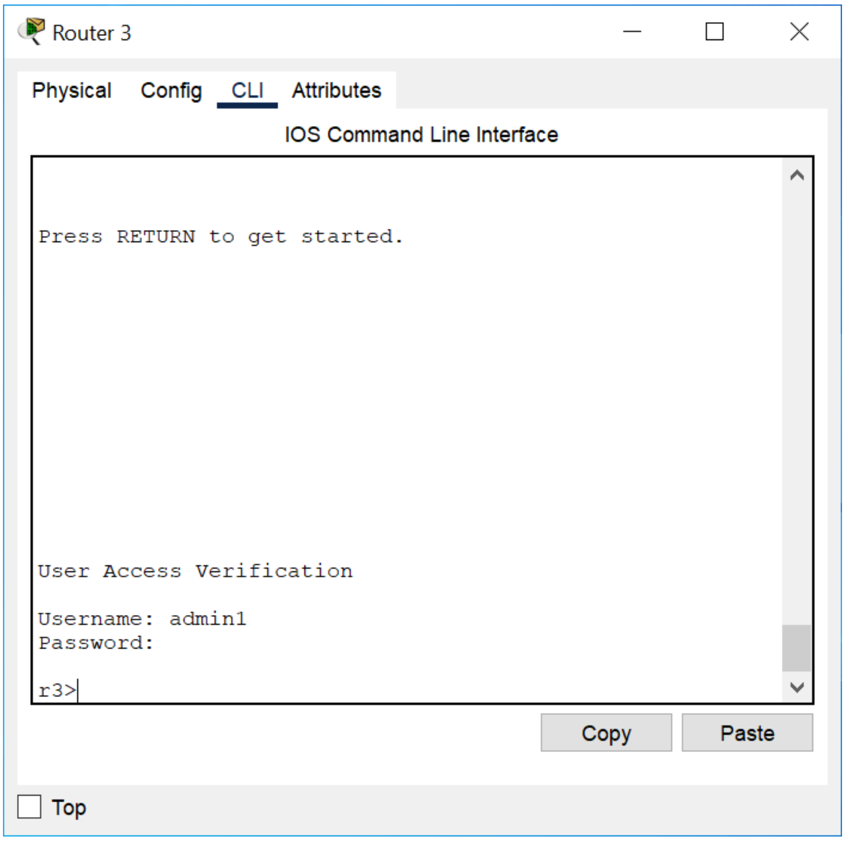
	Username	Password
1	admin1	admin321



```
r3>en
r3#conf t
Enter configuration commands, one per line.  End with
CNTL/Z.
r3(config)#username admin1 secret admin321
r3(config)#radius server host 192.168.3.2
                              ^
% Invalid input detected at '^' marker.

r3(config)#radius-server host 192.168.3.2
r3(config)#radius-server key pwdr3
r3(config)#aaa new-model
r3(config)#aaa authentication login default group radius
local
r3(config)#line console 0
r3(config-line)#login authentication default
r3(config-line)#^Z
r3#
%SYS-5-CONFIG_I: Configured from console by console
```

r3>en
r3#conf t
r3(config)#username admin1 secret admin321
r3(config)#radius-server host 192.168.3.2
r3(config)#radius-server key pwdr3
r3(config)#aaa new-model
r3(config)#aaa authentication login default group radius local
r3(config)#line console 0
r3(config-line)#login authentication default

User Access Verification is asked enter the credentials of the RADIUS Server.



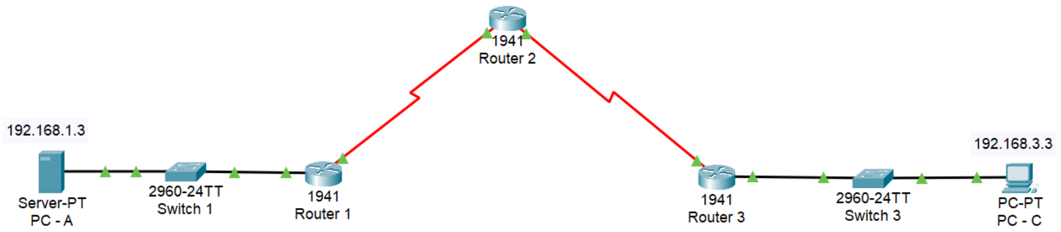
```
Press RETURN to get started.






User Access Verification

Username: admin1
Password:

r3>
```

Username is admin1 and password is admin321 which is also set in radius server.

Practical 11 – Configure a Zone-Based Policy Firewall (ZPF).

A) Topology

9) Drag and drop the required icons that are 3 1941 Routers; 1 PT-PC; 2 2960-24TT Switches; 1 PT-Server and rename them accordingly.
10) Define the respective IP configuration by going to Desktop and then to IP Configuration for PC - A(Server) and PC - C.
11) Add serial ports to all the three routers by going to Physical of Router 1, switching the router off and adding HWIC-2T to both the empty slots and turn the switch on, do the same for Router 2 and Router 3.
12) Connect the devices with the respective wired connection.



B) Configure Router

Router 1

```
Router>en
Router#conf t
Router(config)#host r1

r1(config)#int g0/1
r1(config-if)# ip address 192.168.1.1 255.255.255.0
r1(config-if)#no shut

r1(config-if)#int s0/0/0
r1(config-if)#ip address 10.1.1.1 255.255.255.252
```

```
r1(config-if)#no shut
```

Router 2

```
Router>en
Router#conf t
Router(config)#host r2

r2(config)#int s0/0/0
r2(config-if)#ip address 10.1.1.2 255.255.255.252
r2(config-if)#no shut

r2(config-if)#int s0/0/1
r2(config-if)#ip address 10.2.2.2 255.255.255.252
r2(config-if)#no shut
```

Router 3

```
Router>en
Router#conf t
Router(config)#host r3

r3(config)#int g0/1
r3(config-if)#ip address 192.168.3.1 255.255.255.0
r3(config-if)#no shut

r3(config-if)#int s0/0/1
r3(config-if)#ip address 10.2.2.1 255.255.255.252
r3(config-if)#no shut
```

C) Configure Static Routing

Go to Config of Router1 → Static under Routing and add the details that are

Network: 0.0.0.0 Mask: 0.0.0.0 Next Hop: 10.1.1.2

```
r1#show ip route
```

Similarly, for Router2:

Network: 192.168. 1.0 Mask: 255.255.255.0 Next Hop: 10.1.1.1

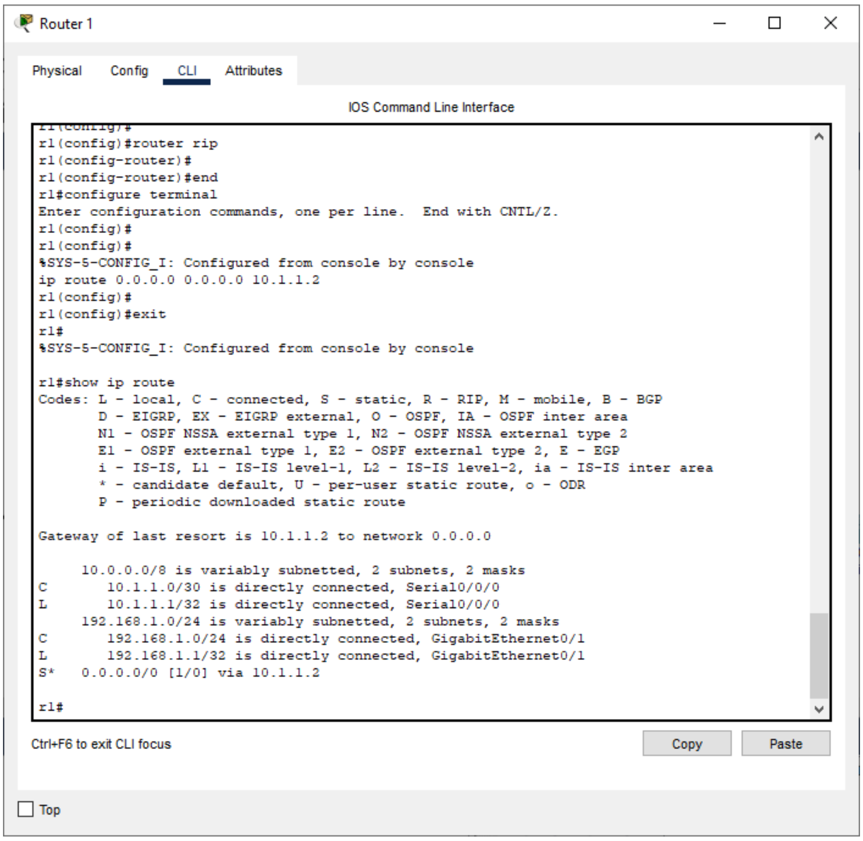
Network: 192.168.3.0 Mask: 255.255.255.0 Next Hop: 10.2.2.1

r2#show ip route

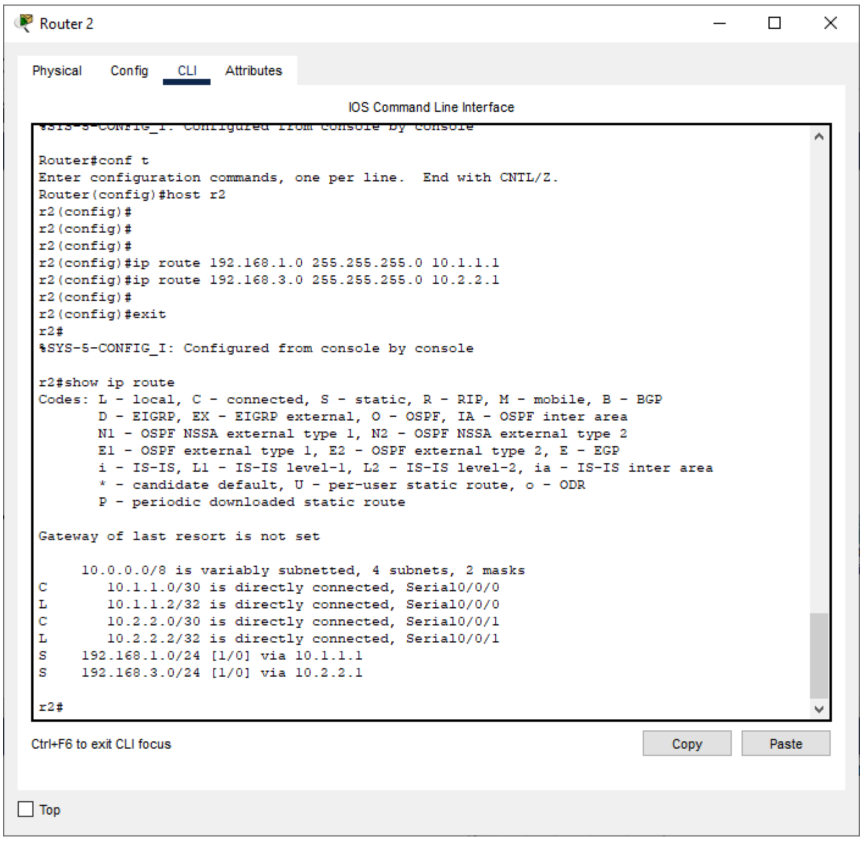
Router3 as well:

Network: 0.0.0.0 Mask: 0.0.0.0 Next Hop: 10.2.2.2

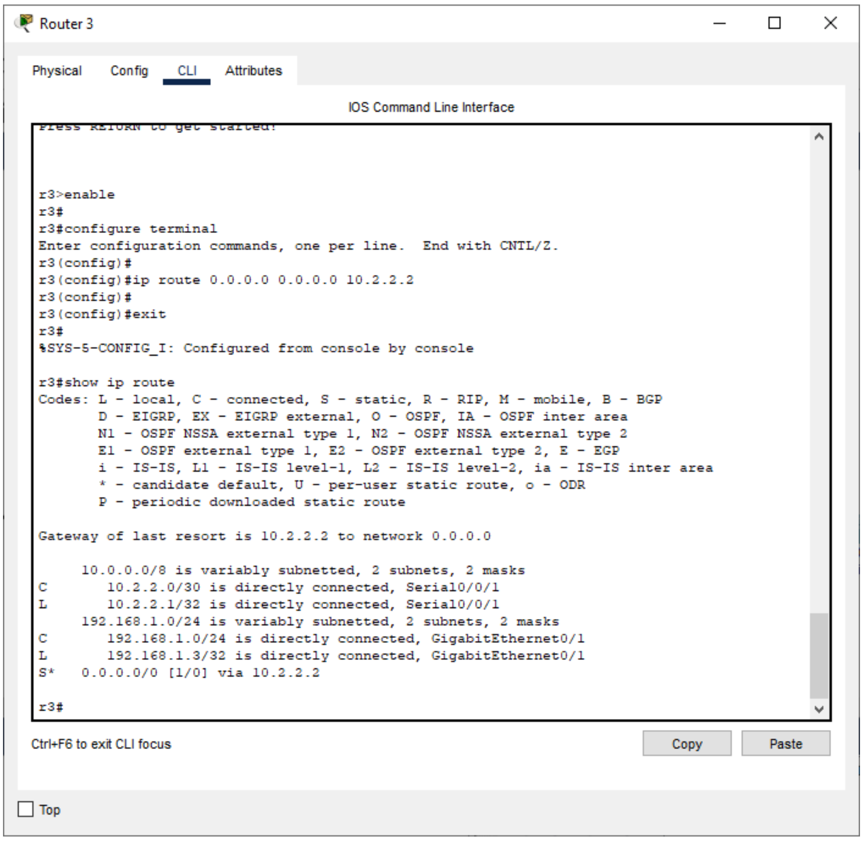
r3#show ip route



```
r1(config)#
r1(config)#router rip
r1(config-router)#
r1(config-router)#end
r1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
r1(config)#
r1(config)#
%SYS-5-CONFIG_I: Configured from console by console
ip route 0.0.0.0 0.0.0.0 10.1.1.2
r1(config)#
r1(config)#exit
r1#
%SYS-5-CONFIG_I: Configured from console by console

r1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 10.1.1.2 to network 0.0.0.0

     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.1.1.0/30 is directly connected, Serial0/0/0
L       10.1.1.1/32 is directly connected, Serial0/0/0
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, GigabitEthernet0/1
L       192.168.1.1/32 is directly connected, GigabitEthernet0/1
S*   0.0.0.0/0 [1/0] via 10.1.1.2

r1#
```



```
%SYS-5-CONFIG_I: Configured from console by console

Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#host r2
r2(config)#
r2(config)#
r2(config)#
r2(config)#ip route 192.168.1.0 255.255.255.0 10.1.1.1
r2(config)#ip route 192.168.3.0 255.255.255.0 10.2.2.1
r2(config)#
r2(config)#exit
r2#
%SYS-5-CONFIG_I: Configured from console by console

r2#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C       10.1.1.0/30 is directly connected, Serial0/0/0
L       10.1.1.2/32 is directly connected, Serial0/0/0
C       10.2.2.0/30 is directly connected, Serial0/0/1
L       10.2.2.2/32 is directly connected, Serial0/0/1
S    192.168.1.0/24 [1/0] via 10.1.1.1
S    192.168.3.0/24 [1/0] via 10.2.2.1

r2#
```



```
Press RETURN to get started!


r3>enable
r3#
r3#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
r3(config)#
r3(config)#ip route 0.0.0.0 0.0.0.0 10.2.2.2
r3(config)#
r3(config)#exit
r3#
%SYS-5-CONFIG_I: Configured from console by console

r3#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 10.2.2.2 to network 0.0.0.0

     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.2.2.0/30 is directly connected, Serial0/0/1
L       10.2.2.1/32 is directly connected, Serial0/0/1
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, GigabitEthernet0/1
L       192.168.1.3/32 is directly connected, GigabitEthernet0/1
S*   0.0.0.0/0 [1/0] via 10.2.2.2

r3#
```

The S proves that static routing is successful.

D) Password Authentication of Router

Router1:

```
r1(config)#enable secret ty22
r1(config)#line console 0
r1(config-line)#password ty22
r1(config-line)#login
r1(config-line)#exit
```

Exit from the connection.

Under User Access Verification password is asked, enter ty22, it will not be seen but the system will accept it.

Connection to the router will open again.

In Router1:

```
r1#conf t
r1(config)#username admin1 secret admin123
r1(config)#aaa new-model
r1(config)#aaa authentication login default local
r1(config)#line console 0
r1(config-line)#login authentication default
r1(config-line)#end
```

End the connection completely.
Again, User Access Verification will come, enter Username as admin1 and Password as admin123. After enabling the router if again a password is asked enter ty22.

```
r1(config)#ip domain-name security.com
r1(config)#crypto key generate rsa
```

How many bits in the modulus will be asked, enter 1024.

```
r1(config)#aaa authentication login sshlogin local
r1(config)#line vty 0 4
```

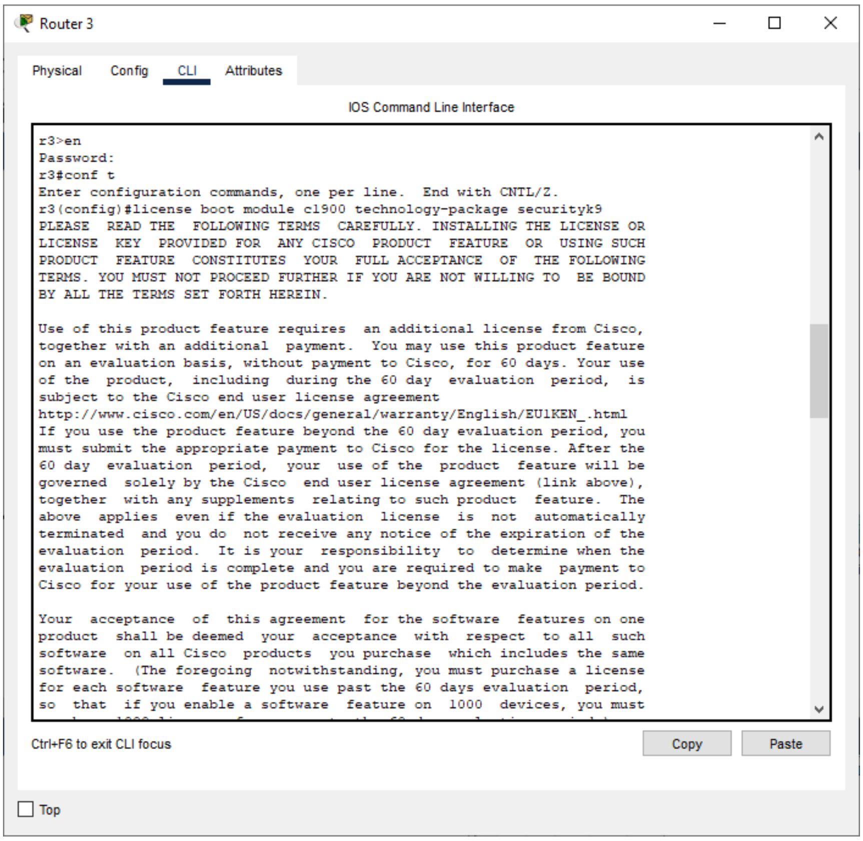
r1(config-line)#login authentication ssh login
r1(config-line)#transport input ssh
r1(config-line)#end

E) Enable security technology package on Router3

In Router3 CLI:

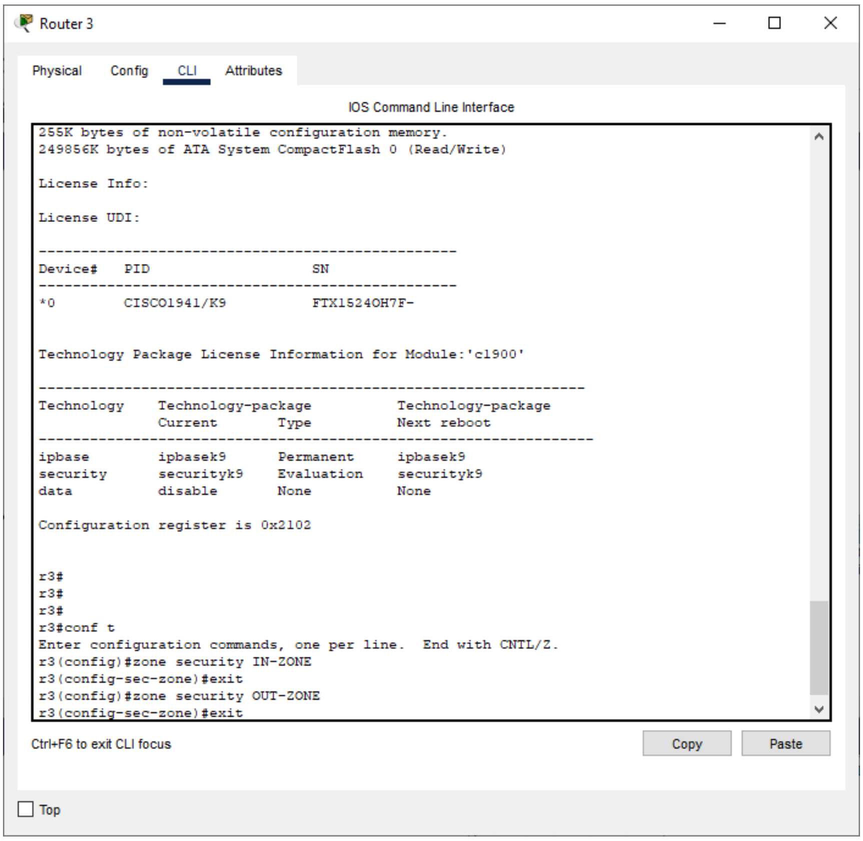
r3(config)#license boot module c1900 technology-package securityk9

Accept the agreement by typing yes.



```
r3>en
Password:
r3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
r3(config)#license boot module c1900 technology-package securityk9
PLEASE  READ THE  FOLLOWING TERMS  CAREFULLY. INSTALLING THE LICENSE OR
LICENSE  KEY  PROVIDED FOR  ANY CISCO  PRODUCT  FEATURE  OR  USING SUCH
PRODUCT  FEATURE  CONSTITUTES  YOUR  FULL ACCEPTANCE  OF  THE FOLLOWING
TERMS. YOU MUST NOT PROCEED FURTHER IF YOU ARE NOT WILLING TO  BE BOUND
BY ALL THE TERMS SET FORTH HEREIN.

Use of this product feature requires  an additional license from Cisco,
together with an additional  payment.  You may use this product feature
on an evaluation basis, without payment to Cisco, for 60 days. Your use
of the  product,  including  during the 60 day  evaluation  period,  is
subject to the Cisco end user license agreement
http://www.cisco.com/en/US/docs/general/warranty/English/EU1KEN_.html
If you use the product feature beyond the 60 day evaluation period, you
must submit the appropriate payment to Cisco for the license. After the
60 day  evaluation  period,  your  use of the  product  feature will be
governed  solely by the Cisco  end user license agreement (link above),
together  with any supplements  relating to such product  feature.  The
above  applies  even if the evaluation  license  is  not  automatically
terminated  and you do  not receive any notice of the expiration of the
evaluation  period.  It is your  responsibility  to  determine when the
evaluation  period is complete and you are required to make  payment to
Cisco for your use of the product feature beyond the evaluation period.

Your  acceptance  of  this agreement  for the software  features on one
product  shall be deemed  your  acceptance  with  respect  to all  such
software  on all Cisco  products  you purchase  which includes the same
software.  (The foregoing  notwithstanding, you must purchase a license
for each software  feature you use past the 60 days evaluation  period,
so  that  if you enable a software  feature on  1000  devices, you must
```

r3#reload
Confirm the reload.
r3#show version



```
255K bytes of non-volatile configuration memory.
249856K bytes of ATA System CompactFlash 0 (Read/Write)

License Info:

License UDI:

-------------------------------------------------
Device#   PID                   SN
-------------------------------------------------
*0        CISCO1941/K9          FTX1524OH7F-


Technology Package License Information for Module:'c1900'

----------------------------------------------------------------
Technology    Technology-package           Technology-package
              Current       Type           Next reboot
-----------------------------------------------------------------
ipbase        ipbasek9      Permanent      ipbasek9
security      securityk9    Evaluation     securityk9
data          disable       None           None

Configuration register is 0x2102


r3#
r3#
r3#
r3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
r3(config)#zone security IN-ZONE
r3(config-sec-zone)#exit
r3(config)#zone security OUT-ZONE
r3(config-sec-zone)#exit
```

F) Create firewall zone on Router3

r3(config)#zone security IN-ZONE
r3(config-sec-zone)#exit
r3(config)#zone security OUT-ZONE
r3(config-sec-zone)#exit

G) Identify traffic using a class-map

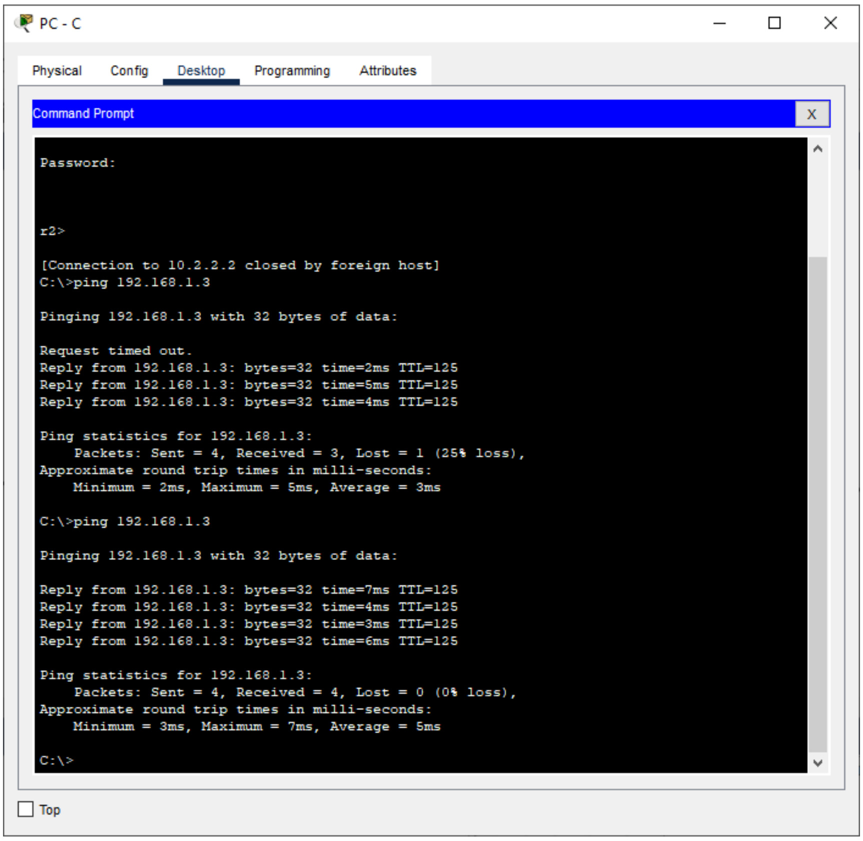
r3(config)# access-list 101 permit ip 192.168.3.0 0.0.0.255 any
r3(config)# class-map type inspect match-all IN-NET-CLASS-MAP
r3(config-cmap)# match access-group 101
r3(config-cmap)# exit

H) Firewall policies

```
r3(config)# policy-map type inspect IN-2-OUT-PMAP
r3(config-pmap)# class type inspect IN-NET-CLASS-MAP
r3(config-pmap-c)# inspect
r3(config-pmap-c)# exit
r3(config-pmap)# exit
r3(config)# zone-pair security IN-2-OUT-ZPAIR source IN-ZONE destination OUT-ZONE
r3(config-sec-zone-pair)# service-policy type inspect IN-2-OUT-PMAP
r3(config-sec-zone-pair)# exit
r3(config)# interface g0/1
r3(config-if)# zone-member security IN-ZONE
r3(config-if)# exit
r3(config)# interface s0/0/1
r3(config-if)# zone-member security OUT-ZONE
r3(config-if)# exit
```

I) Test firewall functionality from IN-ZONE to OUT-ZONE

Open Command Prompt of PC – C and ping 192.168.1.3 that is PC – A (Server). It should be successful.

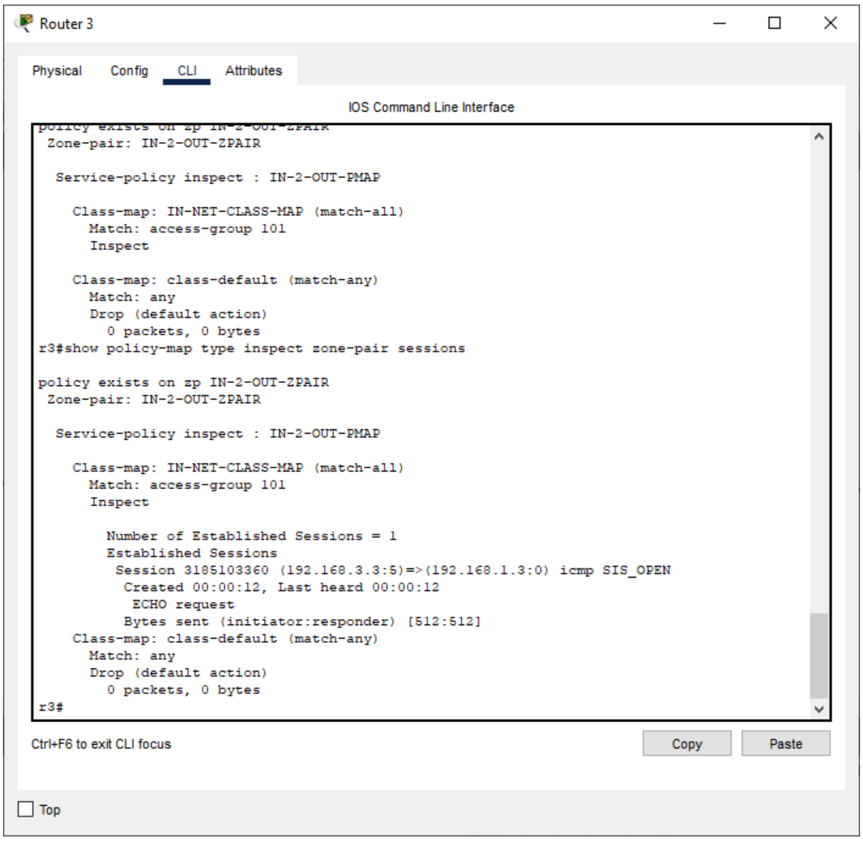


```
Password:


r2>

[Connection to 10.2.2.2 closed by foreign host]
C:\>ping 192.168.1.3

Pinging 192.168.1.3 with 32 bytes of data:

Request timed out.
Reply from 192.168.1.3: bytes=32 time=2ms TTL=125
Reply from 192.168.1.3: bytes=32 time=5ms TTL=125
Reply from 192.168.1.3: bytes=32 time=4ms TTL=125

Ping statistics for 192.168.1.3:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 5ms, Average = 3ms

C:\>ping 192.168.1.3

Pinging 192.168.1.3 with 32 bytes of data:

Reply from 192.168.1.3: bytes=32 time=7ms TTL=125
Reply from 192.168.1.3: bytes=32 time=4ms TTL=125
Reply from 192.168.1.3: bytes=32 time=3ms TTL=125
Reply from 192.168.1.3: bytes=32 time=6ms TTL=125

Ping statistics for 192.168.1.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 7ms, Average = 5ms

C:\>
```

In CLI mode of Router 3 type,

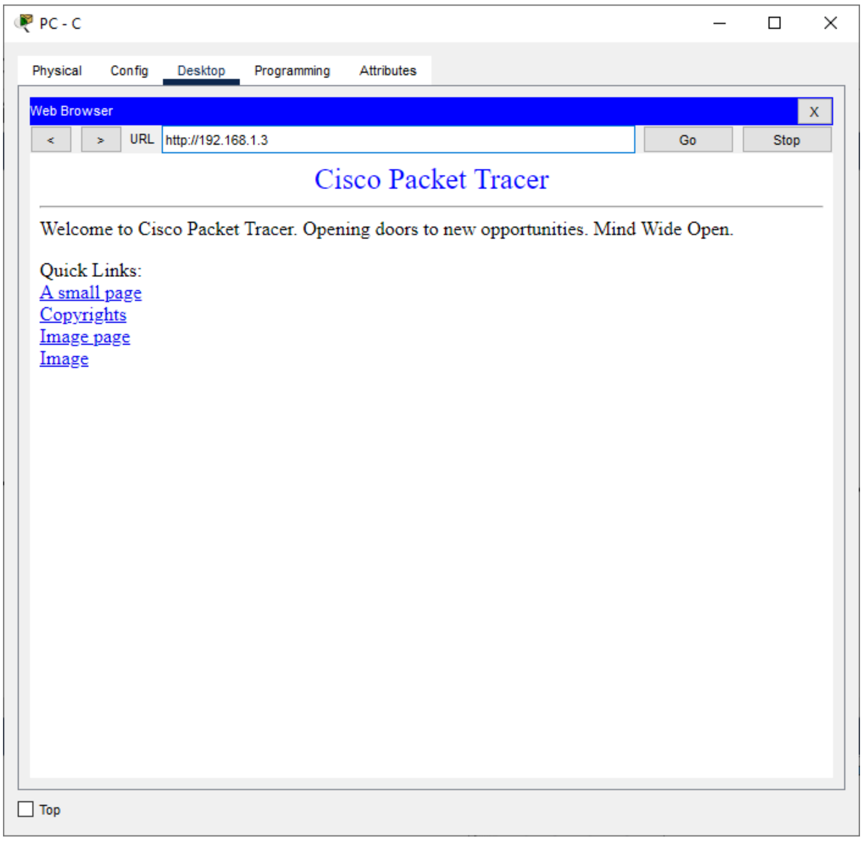
r3#show policy-map type inspect zone-pair sessions

The details regarding the policy should be displayed.



```
policy exists on zp IN-2-OUT-ZPAIR
 Zone-pair: IN-2-OUT-ZPAIR

  Service-policy inspect : IN-2-OUT-PMAP

    Class-map: IN-NET-CLASS-MAP (match-all)
      Match: access-group 101
      Inspect

    Class-map: class-default (match-any)
      Match: any
      Drop (default action)
        0 packets, 0 bytes
r3#show policy-map type inspect zone-pair sessions

policy exists on zp IN-2-OUT-ZPAIR
 Zone-pair: IN-2-OUT-ZPAIR

  Service-policy inspect : IN-2-OUT-PMAP

    Class-map: IN-NET-CLASS-MAP (match-all)
      Match: access-group 101
      Inspect

        Number of Established Sessions = 1
        Established Sessions
         Session 3185103360 (192.168.3.3:5)=>(192.168.1.3:0) icmp SIS_OPEN
          Created 00:00:12, Last heard 00:00:12
           ECHO request
          Bytes sent (initiator:responder) [512:512]
    Class-map: class-default (match-any)
      Match: any
      Drop (default action)
        0 packets, 0 bytes
r3#
```

Go to Web Browser of PC - C enter http://192.168.1.3 and click on Go, Cisco Packet Tracer page will be displayed.



URL: http://192.168.1.3

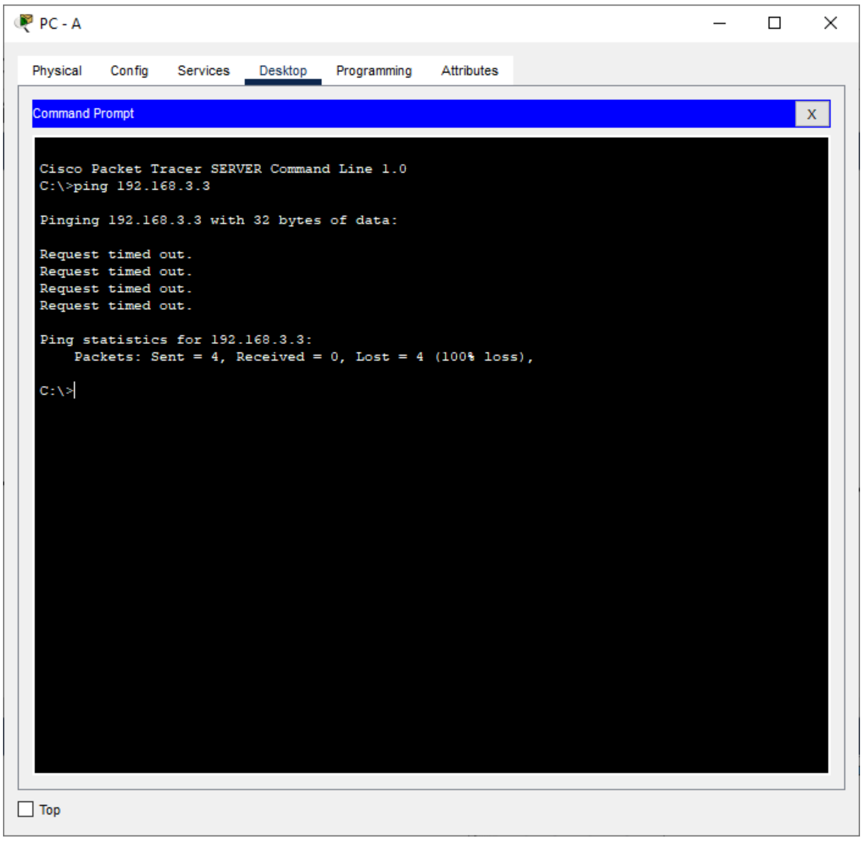
Cisco Packet Tracer

Welcome to Cisco Packet Tracer. Opening doors to new opportunities. Mind Wide Open.

Quick Links:
A small page
Copyrights
Image page
Image

J) Test firewall functionality from OUT-ZONE to IN-ZONE

Open Command Prompt of PC – A and ping 192.168.3.3 that is PC – C. It should fail.



```
Cisco Packet Tracer SERVER Command Line 1.0
C:\>ping 192.168.3.3

Pinging 192.168.3.3 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.3.3:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>
```