Mahatma Gandhi Institute Of Technology

CSE DEPARTMENT

LABORATORY MANUAL

PREPARED BY

D S BHAVANI

Assistant Professor

Department of CSE

INDEX

12	Implement ElGamal Cryptosystem in C/JAVA.	
13	Implement ElGamal Digital Signature in C/JAVA for Authentication.	

1. XOR a string with a Zero

AIM: 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 display the result.

DEESCRIPTION:

- String "Hello World" is taken as Input.
- Each character of the string is XORed with 0 as H^0.
- The above operation internally performed as

H	1	0	0	1	0	0	0
0	0	0	0	0	0	0	0

XOR(^)

	1	0	0	1	0	0	0

- Likewise, every character is XORed with 0 which results in the same character.

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

Viva Questions:

1. Specify the four categories of Security Threats.

   Interruption, Interception, Modification, Fabrication

2. Explain Active and Passive Attack with example.

   Passive attack: Monitoring the message during transmission. Eg: Interception Active attack: It involves the modification of data stream or creation of false data stream. E.g.: Fabrication, Modification, and Interruption

3. Define Integrity and Non-Repudiation.

   Integrity: Service that ensures that only authorized person able to modify the message.

   Non repudiation: This service helps to prove that the person who denies the transaction is true or false.

4. Differentiate Symmetric and Asymmetric Encryption?

   It is a form of cryptosystem in which It is a form of cryptosystem in which encryption and decryption performed using encryption and decryption Performed using the same key. Eg: DES, AES two keys. Eg:RSA,ECC

5. Define Cryptanalysis?

   It is a process of attempting to discover the key or plaintext or both.

2. XOR a string with a 127

AIM: 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.

DESCRIPTION:

AND Operation with 127

- String "Hello World" is taken as Input.
- Each character of the string is performed AND with 127 as H&127.
- The above operation internally performed as

H	1	0	0	1	0	0	0
0	1	1	1	1	1	1	1

AND(&)

	1	0	0	1	0	0	0

- Likewise, every character is performed AND with 127 which results in the same character.

OR Operation with 127

- String "Hello World" is taken as Input.
- Each character of the string is performed OR with 127 as H|127.
- The above operation internally performed as

H	1	0	0	1	0	0	0
0	1	1	1	1	1	1	1

OR(||)

	1	1	1	1	1	1	1

- Likewise, every character is performed OR with 127 which results in DEL character (Unprintable).
- Transformation of Data is done.

XOR Operation with 127

- String “Hello World” is taken as Input.
- Each character of the string is performed XOR with 127 as H^127.
- The above operation internally performed as

H	1	0	0	1	0	0	0
127	1	1	1	1	1	1	1
XOR(^)							
	0	1	1	0	1	1	1

- Likewise, every character is performed XOR with 127 which results in the another character.

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");
for(i=0;i<len;i++)
{
        str3[i] = str2[i]|127;
        printf("%c",str3[i]);
}
        printf("\n");
}
```

Output:

Hello World

• • • • • • ◆O

◆◆

• • • • • • ◆•

◆◆◆

Viva Questions:

1. Define Security mechanism

   It is process that is designed to detect prevent, recover from a security attack. Example: Encryption algorithm, Digital signature, Authentication protocols.

2. Define steganography

   Hiding the message into some cover media. It conceals the existence of a message.

3. Why network need security?

   When systems are connected through the network, attacks are possible during transmission time.

4. Define confidentiality and authentication

   Confidentiality: It means how to maintain the secrecy of message. It ensures that the information in a computer system and transmitted information are accessible only for reading by authorized person. Authentication: It helps to prove that the source entity only has involved the transaction.

5. Define cryptography.

   It is a science of writing Secret code using mathematical techniques. The many schemes used for enciphering constitute the area of study known as cryptography.

3. Encryption & Decryption using Cipher Algorithms

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

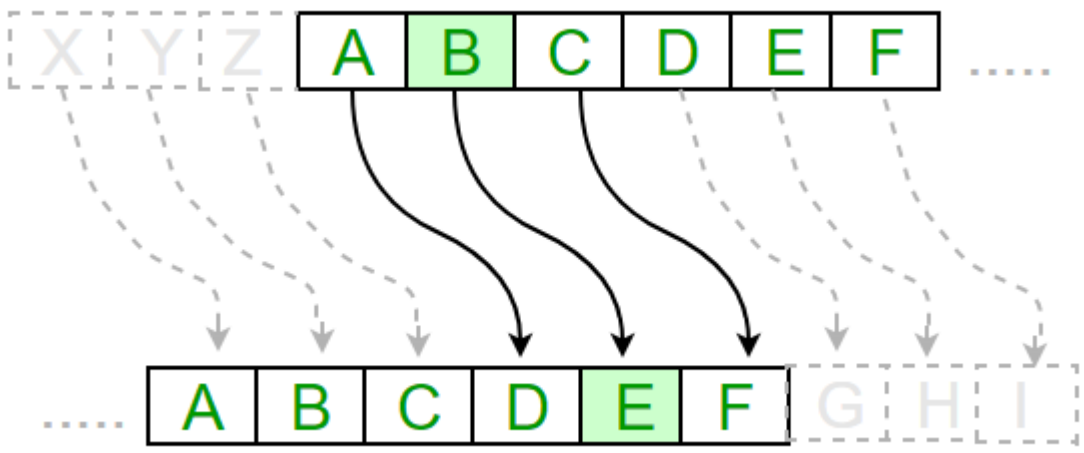
a) Ceasar Cipher
b) Substitution Cipher
c) Hill Cipher
d) Monoalphabetic Cipher
e) Vigenere Cipher
f) Railfence Cipher

CAESAR CIPHER:

DESCRIPTION:

The Caesar Cipher technique is one of the earliest and simplest method of encryption technique. It's simply a type of substitution cipher, i.e., each letter of a given text is replaced by a letter some fixed number of positions down the alphabet.

For example, with a shift of 1, A would be replaced by B, B would become C, and so on. The method is apparently named after Julius Caesar, who apparently used it to communicate with his officials.
Thus, to cipher a given text we need an integer value, known as shift which indicates the number of positions each letter of the text has been moved down.
The encryption can be represented using modular arithmetic by first transforming the letters into numbers, according to the scheme, A = 0, B = 1, Z = 25.

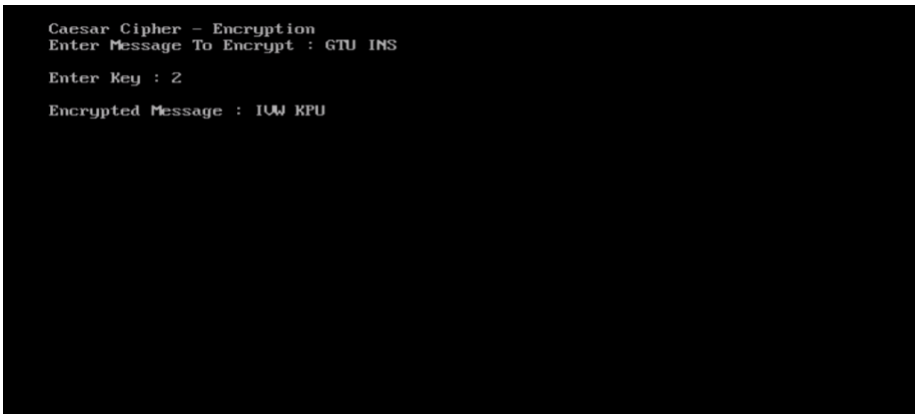


Encryption:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<ctype.h>
void main()
{
int i, key;
char text[100],c;
clrscr();
printf("\nCaesar Cipher - Encryption");
printf("\nEnter Message To Encrypt : ");
gets(text);
printf("\nEnter Key : ");
scanf("%d", &key);
for(i=0;text[i]!='\0';++i)
{
c=text[i];
if(c>='a'&&c<= 'z')
{
```

```
c=c+key;
if(c>'z')
{
c=c-'z'+'a'-1;
}
text[i]=c;
}
else if(c>='A'&&c<='Z')
{
c=c+key;
if(c>'Z')
{
c=c-'Z'+'A'-1;
}
text[i]=c;
}
}
printf("\nEncrypted Message : %s", text);
}
```

Output:

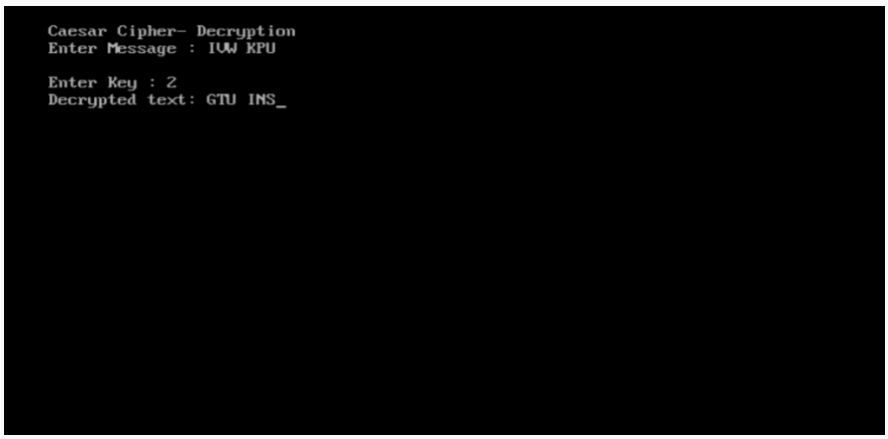


Decryption:

```
#included<stdio.h>
#include<conio.h>
#include<string.h>
#include<ctype.h>
void main()
{
int i, key;
char text[100],c;
printf("\nCaesar Cipher- Decryption");
printf("\nEnter Message To Decrypt : ");
gets(text);
printf("\nEnter Key : ");
scanf("%d", &key);
for(i = 0; text[i] != '\0'; ++i)
{
c = text[i];
if(c >= 'a' && c <= 'z')
{
c = c - key;
if(c < 'a')
{
c = c + 'z' - 'a' + 1;
}
text[i] = c;
}
else if(c >= 'A' && c <= 'Z')
{
c = c - key;
```

```
if(c < 'A')
{
c = c + 'Z' - 'A' + 1;
}
text[i] = c;
}
}
printf("Decrypted text: %s", text);
}
```

Output:



#

Viva Questions:

1. Compare Substitution and Transposition techniques.

   SUBSTITUTION TRANSPOSITION *A substitution techniques is one in * It means, different kind of mapping is which the letters of plaintext are replaced by other letter or by number or symbols. achieved by performing some sort of *Eg: Caeser cipher. permutation on the plaintext letters. *Eg: DES, AES.

2. Define Diffusion & Confusion.

Diffusion: It means each plaintext digits affect the values of many ciphertext digits which is equivalent to each ciphertext digit is affected by many plaintext digits. It can be achieved by performing permutation on the data. It is the relationship between the plaintext and ciphertext. Confusion: It can be achieved by substitution algorithm. It is the relationship between cipher text and key.

3. Define Multiple Encryption.

   It is a technique in which the encryption is used multiple times. Eg: Double DES, Triple DES

4. Specify the design criteria of block cipher.

   f Number of rounds *f* Design of the function F *f* Key scheduling

5. Define Reversible mapping.

   Each plain text is maps with the unique cipher text. This transformation is called reversible mapping.

SUBSTITUTION CIPHER

DESCRIPTION:

The Caesar cipher is the simplest and oldest method of cryptography. The Caesar cipher method is based on a mono-alphabetic cipher and is also called a shift cipher or additive cipher. Julius Caesar used the shift cipher (additive cipher) technique to communicate with his officers. For this reason, the shift cipher technique is called the Caesar cipher. The Caesar cipher is a kind of replacement (substitution) cipher, where all letter of plain text is replaced by another letter.

Let's take an example to understand the Caesar cipher, suppose we are shifting with 1, then A will be replaced by B, B will be replaced by C, C will be replaced by D, D will be replaced by C, and this process continues until the entire plain text is finished.

Caesar ciphers is a weak method of cryptography. It can be easily hacked. It means the message encrypted by this method can be easily decrypted.

Plaintext: It is a simple message written by the user.

Ciphertext: It is an encrypted message after applying some technique.

The formula of encryption is:

$E_n(x) = (x + n) \bmod 26$

The formula of decryption is:

$D_n(x) = (xi - n) \bmod 26$

If any case (Dn) value becomes negative (-ve), in this case, we will add 26 in the negative value.

Encryption:

```
#include<stdio.h>

void main()
{
	char message[100], ch;
	int i, key;

	printf("Enter a message to encrypt: ");
	gets(message);
	printf("Enter key: ");
	scanf("%d", &key);

	for(i = 0; message[i] != '\0'; ++i){
		ch = message[i];

		if(ch >= 'a' && ch <= 'z'){
			ch = ch + key;

			if(ch > 'z'){
				ch = ch - 'z' + 'a' - 1;
			}

			message[i] = ch;
		}
		else if(ch >= 'A' && ch <= 'Z'){
			ch = ch + key;

			if(ch > 'Z'){
				ch = ch - 'Z' + 'A' - 1;
			}

			message[i] = ch;
	}
```

Output:

Enter a message to encrypt:axzd

Enter key: 4

Encrypted message: ebdh

Decryption:

```
#include<stdio.h>

void main()
{
char message[100], ch;
int i, key;
printf("Enter a message to decrypt: ");
gets(message);
printf("Enter key: ");
scanf("%d", &key);
for(i = 0; message[i] != '\0'; ++i){
ch = message[i];
if(ch >= 'a' && ch <= 'z'){
ch = ch - key;
if(ch < 'a'){
ch = ch + 'z' - 'a' + 1;
}
message[i] = ch;
}
else if(ch >= 'A' && ch <= 'Z'){
ch = ch - key;
if(ch < 'A'){
ch = ch + 'Z' - 'A' + 1;
}
message[i] = ch;
}
}
printf("Decrypted message: %s", message);
}
```

Output

Enter a message to decrypt: ebdh

Enter key: 4
Decrypted message: axzd

Viva Questions:

1. Specify the basic task for defining a security service.

   A service that enhances the security of the data processing systems and the information transfer of an organization. The services are intended to counter security attack, and they make use of one or more security mechanism to provide the service.

2. Define network security.

   This area covers the use of cryptographic algorithms in network protocols and network applications.

3. Define computer security.

   This term refers to the security of computers against intruders and malicious software.

4. What are hill cipher merits and demerits?

   Completely hides single letter and 2 letter frequency information.

5. List-out the types of attack in ceaser cipher.

   Brute force attack. Just try all the 25 possible keys.

HILL CIPHER:

DESCRIPTION:

Hill cipher is a polygraphic substitution cipher based on linear algebra. Each letter is represented by a number modulo 26. Often the simple scheme A = 0, B = 1, ..., Z = 25 is used, but this is not an essential feature of the cipher. To encrypt a message, each block of n letters (considered as an n-component vector) is multiplied by an invertible n × n matrix, against modulus 26. To decrypt the message, each block is multiplied by the inverse of the matrix used for encryption.
The matrix used for encryption is the cipher key, and it should be chosen randomly from the set of invertible n × n matrices (modulo 26).

Example:

Input : Plaintext: ACT
Key: GYBNQKURP
Output: Ciphertext: POH

Input: Plaintext: GFG
Key: HILLMAGIC
Output: Ciphertext: SWK

Encryption

We must encrypt the message 'ACT' (n=3). The key is 'GYBNQKURP' which can be written as the nxn matrix:

$$\begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix}$$

The message 'ACT' is written as vector:

$$\begin{bmatrix} 0 \\ 2 \\ 19 \end{bmatrix}$$

The enciphered vector is given as:

$$\begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ 19 \end{bmatrix} = \begin{bmatrix} 67 \\ 222 \\ 319 \end{bmatrix} \equiv \begin{bmatrix} 15 \\ 14 \\ 7 \end{bmatrix} (\text{mod } 26)$$

which corresponds to ciphertext of 'POH'

Decryption

To decrypt the message, we turn the ciphertext back into a vector, then simply multiply by the inverse matrix of the key matrix (IFKVIVVMI in letters). The inverse of the matrix used in the previous example is:

$$\begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix}^{-1} \equiv \begin{bmatrix} 8 & 5 & 10 \\ 21 & 8 & 21 \\ 21 & 12 & 8 \end{bmatrix} \pmod{26}$$

For the previous Ciphertext 'POH':

$$\begin{bmatrix} 8 & 5 & 10 \\ 21 & 8 & 21 \\ 21 & 12 & 8 \end{bmatrix} \begin{bmatrix} 15 \\ 14 \\ 7 \end{bmatrix} \equiv \begin{bmatrix} 260 \\ 574 \\ 539 \end{bmatrix} \equiv \begin{bmatrix} 0 \\ 2 \\ 19 \end{bmatrix} \pmod{26}$$

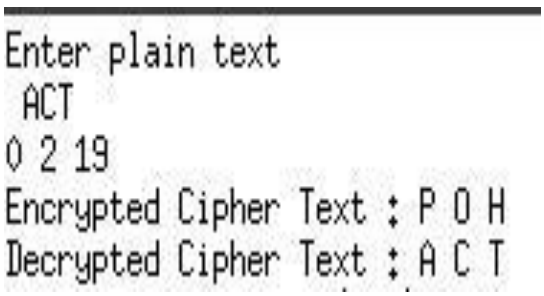
which gives us back 'ACT'.

```
#include<stdio.h>
#include<string.h>
int main(){
```

```
unsigned int a[3][3]={{6,24,1},{13,16,10},{20,17,15}};
unsigned int b[3][3]={{8,5,10},{21,8,21},{21,12,8}};
int i,j;
unsigned int c[20],d[20];
char msg[20];
int determinant=0,t=0;;
printf("Enter plain text\n ");
scanf("%s",msg);
for(i=0;i<strlen(msg);i++)< p=""></strlen(msg);i++)
{
c[i]=msg[i]-65;
printf("%d ",c[i]);
}
for(i=0;i<3;i++)
{
t=0;
for(j=0;j<3;j++)
{
t=t+(a[i][j]*c[j]);
}
d[i]=t%26;
}
printf("\nEncrypted Cipher Text :");
for(i=0;i<3;i++)
printf(" %c",d[i]+65);
for(i=0;i<3;i++)
{
t=0;
for(j=0;j<3;j++)
{
t=t+(b[i][j]*d[j]);
```

```
}
c[i]=t%26;
}
printf("\nDecrypted Cipher Text :");
for(i=0;i<3;i++)
printf(" %c",c[i]+65);
return 0;
}
```

Output:



Viva Questions:

1. What are the essential ingredients of a symmetric cipher?
   There are five main components of a symmetric encryption system: plaintext, encryption algorithm, secret key, ciphertext, and the decryption algorithm.

2. What are the two basic functions used in encryption algorithms?
   Substitution and transposition are the two basic functions used in encryption algorithms
3. How many keys are required for two people to communicate via a cipher?

   For symmetric single key is used to encrypt and decrypt while communicating via cipher while in asymmetric two key are used, one for encryption one for decryption.

4. What is the difference between a block cipher and a stream cipher?

   A block cipher breaks down plaintext messages into fixed-size blocks before converting them into ciphertext using a key. Encrypting information bit-by-bit. A stream cipher, on the other hand, breaks a plaintext message down into single bits, which then are converted individually into ciphertext using key bits.

5. What are the two general approaches to attacking a cipher?

   There are two general approaches to attacking a conventional encryption scheme: Cryptanalysis (cryptanalytic attacks): This attack relies on the nature of the algorithm plus some knowledge of the general characteristics of the plaintext or some sample plaintext–ciphertext pairs.

MONOALPHABETIC CIPHER

DESCRIPTION:

Monoalphabetic cipher is one where each character of a plain text is mapped to a fixed other character of cipher text. The relationship between a character in the plain text and the characters in the cipher text is one-to-one.

Example : if a plain text has a character ‘a’ and any key then if it convert into other character say ‘t’ so wherever there is ‘a’ character in plain text it will be mapped to character ‘t’ ,Therefore it is called as monoalphabetic cipher.

It is a simple type of substitution cipher. Monoalphabetic ciphers are not that stronger as compared to polyalphabetic cipher.

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
S W N A M L X C V J B U Y K P D O Q E R I F H G Z T

PROGRAM:

```
import java.io.*;
import java.util.Scanner;
class Mono {
public static char n[] = { '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 c[]= { '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 encrypt(String s){
String es = "";
for (int i = 0; i < s.length(); i++) {
for (int j = 0; j < 26; j++) {
if (s.charAt(i) == n[j])
{
```

```
es+= c[j];
break;
}
}
}
return es;
}
public static String decrypt(String s)
{
String ds = "";
for (int i = 0; i < s.length(); i++)
{
for (int j = 0; j < 26; j++) {
if (s.charAt(i) == c[j])
{
ds += n[j];
break;
}
}
}
return ds;
}
public static void main(String args[])
{
System.out.println("Enter plain text:");
Scanner sc = new Scanner(System.in);
String str = sc.nextLine();
System.out.println("Plain text: " + str);
String estr = encrypt(str.toLowerCase());
System.out.println("Encrypted message: "+ estr);
System.out.println("Decrypted message: " + decrypt(estr));

    }
}
```

Output:

Enter the Plaintext: hello

Plaintext: hello
Encrypted Message: ITAAF
Decrypted Message: hello

Viva Questions:

1. What are the essential ingredients of a symmetric cipher?
   There are five main components of a symmetric encryption system: plaintext, encryption algorithm, secret key, ciphertext, and the decryption algorithm.

2. What are the two basic functions used in encryption algorithms?
   Substitution and transposition are the two basic functions used in encryption algorithms
3. How many keys are required for two people to communicate via a cipher?
   For symmetric single key is used to encrypt and decrypt while communicating via cipher while in asymmetric two key are used, one for encryption one for decryption.

4. What is the difference between a block cipher and a stream cipher?
   A block cipher breaks down plaintext messages into fixed-size blocks before converting them into ciphertext using a key. Encrypting information bit-by-bit. A stream cipher, on the other hand, breaks a plaintext message down into single bits, which then are converted individually into ciphertext using key bits.

5. What are the two general approaches to attacking a cipher?
   There are two general approaches to attacking a conventional encryption scheme: Cryptanalysis (cryptanalytic attacks): This attack relies on the nature of the algorithm plus some knowledge of the general characteristics of the plaintext or some sample plaintext–ciphertext pairs.

VIGENERE CIPHER

DESCRIPTION:

Vigenere Cipher is a method of encrypting alphabetic text. It uses a simple form of polyalphabetic substitution. A polyalphabetic cipher is any cipher based on substitution, using multiple substitution alphabets. The encryption of the original text is done using the *Vigenère square or Vigenère table*.

- The table consists of the alphabets written out 26 times in different rows, each alphabet shifted cyclically to the left compared to the previous alphabet, corresponding to the 26 possible Caesar Ciphers.
- At different points in the encryption process, the cipher uses a different alphabet from one of the rows.
- The alphabet used at each point depends on a repeating keyword.

Example:

Input : Plaintext : GEEKSFORGEEKS
Keyword : AYUSH

Output : Ciphertext : GCYCZFMLYLEIM

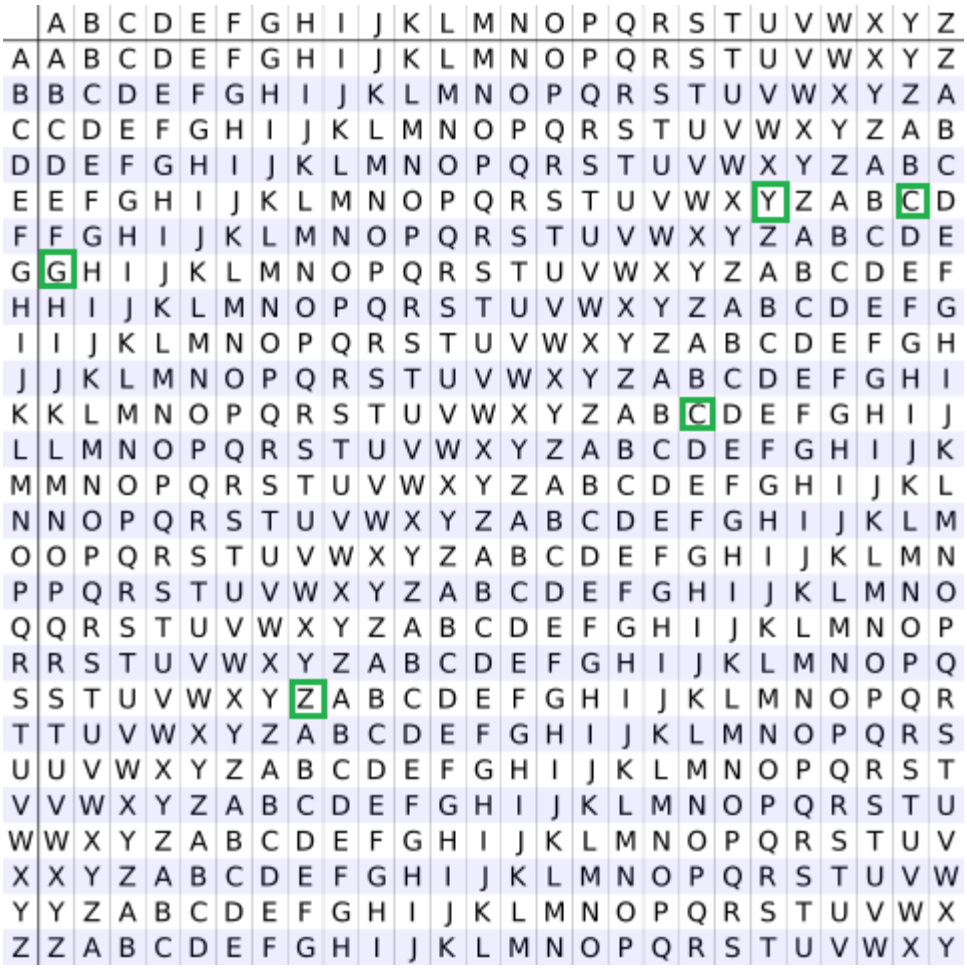
For generating key, the given keyword is repeated in a circular manner until it matches the length of the plain text.

The keyword "AYUSH" generates the key "AYUSHAYUSHAYU"

The plain text is then encrypted using the process explained below.

Encryption

The first letter of the plaintext, G is paired with A, the first letter of the key. So use row G and column A of the Vigenère square, namely G. Similarly, for the second letter of the plaintext, the second letter of the key is used, the letter at row E, and column Y is C. The rest of the plaintext is enciphered in a similar fashion.



	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
A	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
B	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	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	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

Decryption

Decryption is performed by going to the row in the table corresponding to the key, finding the position of the ciphertext letter in this row, and then using the column's label as the plaintext. For example, in row A (from AYUSH), the ciphertext G appears in column G, which is the first plaintext letter. Next, we go to row Y (from AYUSH), locate the ciphertext C which is found in column E, thus E is the second plaintext letter.
A more **easy implementation** could be to visualize Vigenère algebraically by converting [A-Z] into numbers [0–25].

PROGRAM:

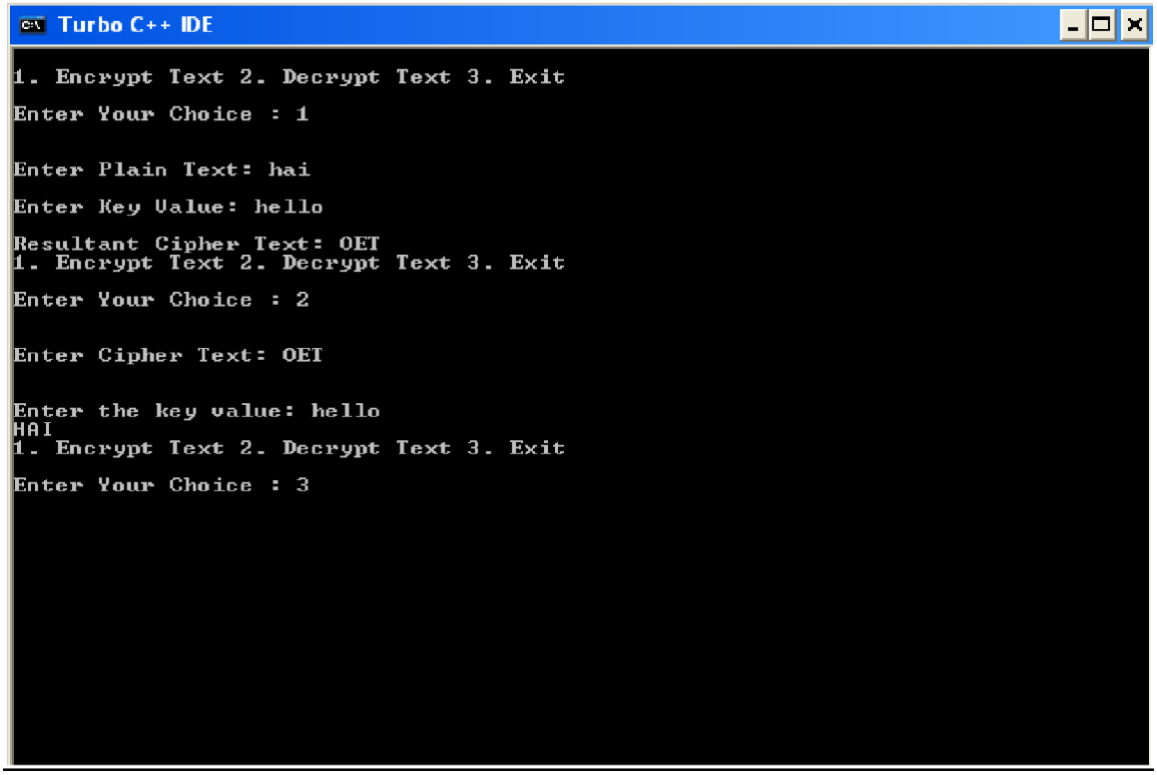
```
#include <stdio.h>
#include<conio.h>
#include <ctype.h>
#include <string.h>
```

```
void encipher(); void
decipher(); void
main()
{
        int choice;
        clrscr(); while(1)
        {
                printf("\n1. Encrypt Text"); printf("\t2.
                Decrypt Text"); printf("\t3. Exit");
                printf("\n\nEnter Your Choice : ");
                scanf("%d",&choice);
                if(choice == 3)
                exit(0);
                else if(choice == 1)encipher();
                else if(choice == 2)decipher();
                else
                printf("Please Enter Valid Option.");
        }
}
void encipher()
{
        unsigned int i,j;
        char input[50],key[10];
        printf("\n\nEnter Plain Text: ");
        scanf("%s",input); printf("\nEnter
        Key Value: ");scanf("%s",key);
        printf("\nResultant Cipher Text: ");
        for(i=0,j=0;i<strlen(input);i++,j++)
        {
                if(j>=strlen(key))
                        {       j=0;
                        }
        printf("%c",65+(((toupper(input[i])-65)+(toupper(key[j])- 65))%26));
        }}
void decipher()
{
```

```
        unsigned int i,j;
        char  input[50],key[10];int
        value;
        printf("\n\nEnter Cipher Text: ");
        scanf("%s",input);
        printf("\n\nEnter the key value: ");
        scanf("%s",key);
    for(i=0,j=0;i<strlen(input);i++,j++)
    {
        if(j>=strlen(key))
        {       j=0;            }
        value  =  (toupper(input[i])-64)-(toupper(key[j])-64);if( value < 0)
        {  value = value * -1;
        }
        printf("%c",65 + (value % 26));
    }}
```

OUTPUT:

Viva Questions:



1. Specify the basic task for defining a security service.

   A service that enhances the security of the data processing systems and the

information transfer of an organization. The services are intended to counter security attack, and they make use of one or more security mechanism to provide the service.

2. Define network security.

   This area covers the use of cryptographic algorithms in network protocols and network applications.

3. Define computer security.

   This term refers to the security of computers against intruders and malicious software.

4. What are hill cipher merits and demerits?

   Completely hides single letter and 2 letter frequency information.

5. List-out the types of attack in ceaser cipher.

   Brute force attack. Just try all the 25 possible keys.

RAILFENCE CIPHER

DESCRIPTION:

Given a plain-text message and a numeric key, cipher/de-cipher the given text using Rail Fence algorithm.

The rail fence cipher (also called a zigzag cipher) is a form of transposition cipher. It derives its name from the way in which it is encoded.

Examples:

Encryption

Input : "GeeksforGeeks "

Key = 3

Output : GsGsekfrek eoe

Decryption

Input : GsGsekfrek eoe

Key = 3

Output : "GeeksforGeeks "

Encryption

Input : "defend the east wall"

Key = 3

Output : dnhaweedtees alf tl

Decryption

Input : dnhaweedtees alf tl

Key = 3

Output : defend the east wall

Encryption

Input : "attack at once"

Key = 2

Output : atc toctaka ne

Decryption

Input : "atc toctaka ne"

Key = 2

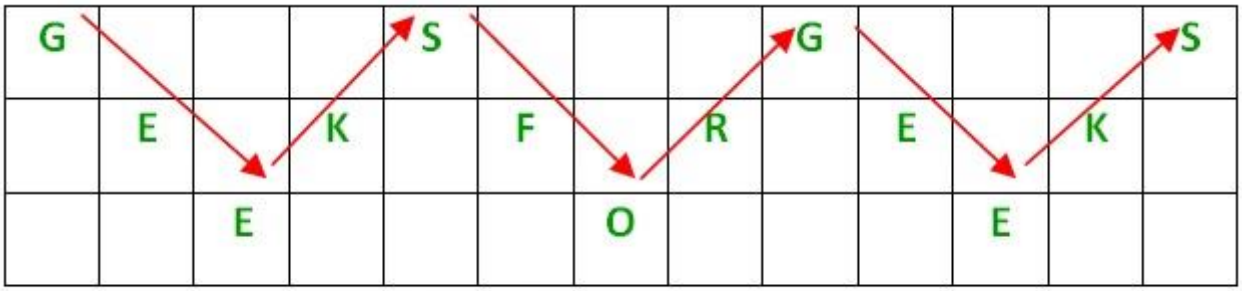
Output : attack at once

Encryption

In a transposition cipher, the order of the alphabets is re-arranged to obtain the cipher-text.

- In the rail fence cipher, the plain-text is written downwards and diagonally on successive rails of an imaginary fence.
- When we reach the bottom rail, we traverse upwards moving diagonally, after reaching the top rail, the direction is changed again. Thus the alphabets of the message are written in a zig-zag manner.
- After each alphabet has been written, the individual rows are combined to obtain the cipher-text.

For example, if the message is “GeeksforGeeks” and the number of rails = 3 then cipher is prepared as:



Decryption

As we've seen earlier, the number of columns in rail fence cipher remains equal to the length of plain-text message. And the key corresponds to the number of rails.

- Hence, rail matrix can be constructed accordingly. Once we've got the matrix we can figure-out the spots where texts should be placed (using the same way of moving diagonally up and down alternatively).
- Then, we fill the cipher-text row wise. After filling it, we traverse the matrix in zig-zag manner to obtain the original text.

Implementation:

Let cipher-text = "GsGsekfrek eoe" , and Key = 3

- Number of columns in matrix = len(cipher-text) = 13
- Number of rows = key = 3

Hence original matrix will be of 3*13 , now marking places with text as '*' we get

```
* _ _ _ * _ _ _ * _ _ _ *
_ * _ * _ * _ * _ * _ *
_ _ * _ _ _ * _ _ _ * _
```

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>void main()
{
    int i,j,k,l;
    char  a[20],c[20],d[20];clrscr();
    printf("\n\t\t RAIL FENCE TECHNIQUE");
    printf("\n\nEnter the input string : ");gets(a);
```

```
        l=strlen(a);

/*Ciphering*/ for(i=0,j=0;i<l;i++)
{
        if(i%2==0) c[j++]=a[i];
}
for(i=0;i<l;i++)
{
        if(i%2==1) c[j++]=a[i];
}
c[j]='\0';
printf("\nCipher text after applying rail fence :");printf("\n%s",c);

/*Deciphering*/if(l%2==0)
        k=l/2;
else
        k=(l/2)+1;
for(i=0,j=0;i<k;i++)
{
      d[j]=c[i];j=j+2;
}
for(i=k,j=1;i<l;i++)
{
      d[j]=c[i];j=j+2;
}
d[l]='\0';
printf("\nText after decryption : ");printf("%s",d);
getch();
}
```

Viva Questions:

1. Specify the basic task for defining a security service.

   A service that enhances the security of the data processing systems and the information transfer of an organization. The services are intended to counter security attack, and they make use of one or more security mechanism to provide the service.

2. Define network security.

   This area covers the use of cryptographic algorithms in network protocols and network applications.

3. Define computer security.

   This term refers to the security of computers against intruders and malicious software.

4. What are hill cipher merits and demerits?

   Completely hides single letter and 2 letter frequency information.

5. List-out the types of attack in ceaser cipher.

   Brute force attack. Just try all the 25 possible key

PLAYFAIR CIPHER

DESCRIPTION:

Playfair cipher is an encryption algorithm to encrypt or encode a message. It is the same as a traditional cipher. The only difference is that it encrypts a **digraph** (a pair of two letters) instead of a single letter.

It initially creates a key-table of 5*5 matrix. The matrix contains alphabets that act as the key for encryption of the plaintext. Note that any alphabet should not be repeated. Another point to note that there are 26 alphabets and we have only 25 blocks to put a letter inside it. Therefore, one letter is excess so, a letter will be omitted (usually J) from the matrix. Nevertheless, the plaintext contains J, then **J** is replaced by **I**. It means treat I and J as the same letter, accordingly.

Since Playfair cipher encrypts the message **digraph by digraph**. Therefore, the Playfair cipher is an example of a **digraph substitution cipher**.

1. First, split the plaintext into **digraphs** (pair of two letters). If the plaintext has the odd number of letters, append the letter **Z** at the end of the plaintext. It makes the plaintext of **even** For example, the plaintext **MANGO** has five letters. So, it is not possible to make a digraph. Since, we will append a letter **Z** at the end of the plaintext, i.e. **MANGOZ.**

2. After that, break the plaintext into **digraphs** (pair of two letters). If any letter appears twice (side by side), put **X** at the place of the second occurrence. Suppose, the plaintext is **COMMUNICATE** then its digraph becomes **CO MX MU NI CA TE**. Similarly, the digraph for the plaintext **JAZZ** will be **JA ZX ZX**, and for plaintext **GREET**, the digraph will be **GR EX ET.**

3. To determine the cipher (encryption) text, first, build a 5*5 key-matrix or key-table and filled it with the letters of alphabets, as directed below:

- Fill the first row (left to right) with the letters of the given keyword (**ATHENS**). If the keyword has duplicate letters (if any) avoid them. It means a letter will be considered only once. After that, fill the remaining letters in alphabetical order. Let's create a 5*5 key-matrix for the keyword **ATHENS**.

A	T	H	E	N
S	B	C	D	F
G	I/J	K	L	M
O	P	Q	R	U
V	W	X	Y	Z

Note that in the above matrix any letter is not repeated. The letters in the first row (in green color) represent the keyword and the remaining letters sets in alphabetical order.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<ctype.h>
#define MX 5
void playfair(char ch1,char ch2, char key[MX][MX])
{
        int i,j,w,x,y,z;FILE
        *out;
if((out=fopen("cipher.txt","a+"))==NULL)
{
        printf("File  Currupted.");
}
for(i=0;i<MX;i++)
{
```

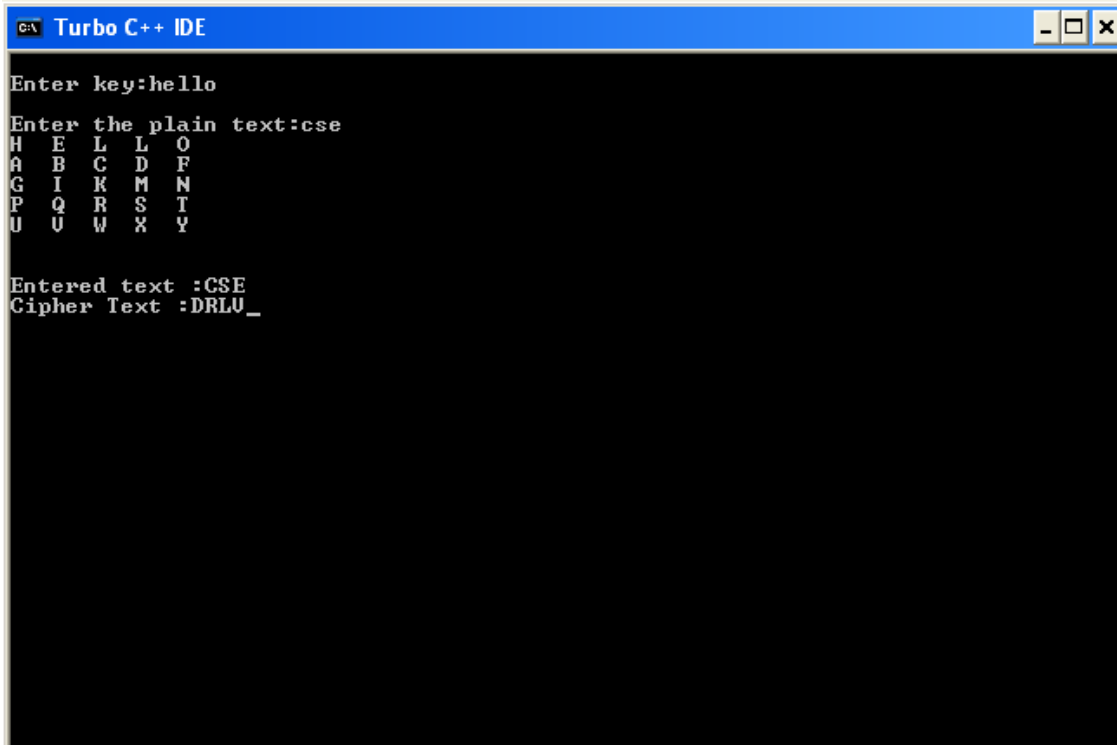
```
        for(j=0;j<MX;j++)
        {
        if(ch1==key[i][j])
        {
        w=i;
        x=j;
        }
        else  if(ch2==key[i][j])
        {
        y=i;
        z=j;
}}}

//printf("%d%d  %d%d",w,x,y,z);if(w==y)
{
        x=(x+1)%5;z=(z+1)%5;
        printf("%c%c",key[w][x],key[y][z]);
        fprintf(out,  "%c%c",key[w][x],key[y][z]);
}
else  if(x==z)
{
```

```
        w=(w+1)%5;y=(y+1)%5;
        printf("%c%c",key[w][x],key[y][z]);
        fprintf(out,  "%c%c",key[w][x],key[y][z]);
}
else
{
        printf("%c%c",key[w][z],key[y][x]);
        fprintf(out,  "%c%c",key[w][z],key[y][x]);
}

fclose(out);
}
void main()
{
        int i,j,k=0,l,m=0,n;
        char  key[MX][MX],keyminus[25],keystr[10],str[25]={0}; char
        alpa[26]={'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'}
        ;
        clrscr(); printf("\nEnter
        key:");gets(keystr);
        printf("\nEnter the plain text:");
        gets(str);
        n=strlen(keystr);
        //convert the characters to uppertextfor
        (i=0; i<n; i++)
        {
                if(keystr[i]=='j')keystr[i]='i';
                else  if(keystr[i]=='J')keystr[i]='I';keystr[i] =
                toupper(keystr[i]);
        }
        //convert all the characters of plaintext to uppertextfor (i=0;
        i<strlen(str); i++)
        {
```

```
                } j=0;
if(str[i]=='j')str[i]='i';
else  if(str[i]=='J')str[i]='I';str[i] = toupper(str[i]);
                for(i=0;i<26;i++)
                {
                        for(k=0;k<n;k++)
                        {
                        if(keystr[k]==alpa[i])break;
                        else  if(alpa[i]=='J')break;
                        }
                        if(k==n)
                        {
                        keyminus[j]=alpa[i];j++;
                        }
                }

                //construct key keymatrixk=0;
                for(i=0;i<MX;i++)
                {
                        for(j=0;j<MX;j++)
                        {
                        if(k<n)
                        {
                        key[i][j]=keystr[k];k++;}
                        else
                        {
                        key[i][j]=keyminus[m];m++;
                        }
                        printf("%c    ",key[i][j]);
                        }
```

```
                printf("\n");
            }
            printf("\n\nEntered text :%s\nCipher Text :",str);
            for(i=0;i<strlen(str);i++)
            {
            if(str[i]=='J')str[i]='I';if(str[i+1]=='\0')
            playfair(str[i],'X',key);
else
  {
  playfair(str[i],str[i+1],key);i++;
  }}
```

OUTPUT:



Viva Questions:

1. Specify the basic task for defining a security service.

   A service that enhances the security of the data processing systems and the information transfer of an organization. The services are intended to counter security attack, and they make use of one or more security mechanism to provide the service.

2. Define network security.

   This area covers the use of cryptographic algorithms in network protocols and network applications.

3. Define computer security.

   This term refers to the security of computers against intruders and malicious software.

4. What are hill cipher merits and demerits?

   Completely hides single letter and 2 letter frequency information.

5. List-out the types of attack in ceaser cipher.

   Brute force attack. Just try all the 25 possible keys.

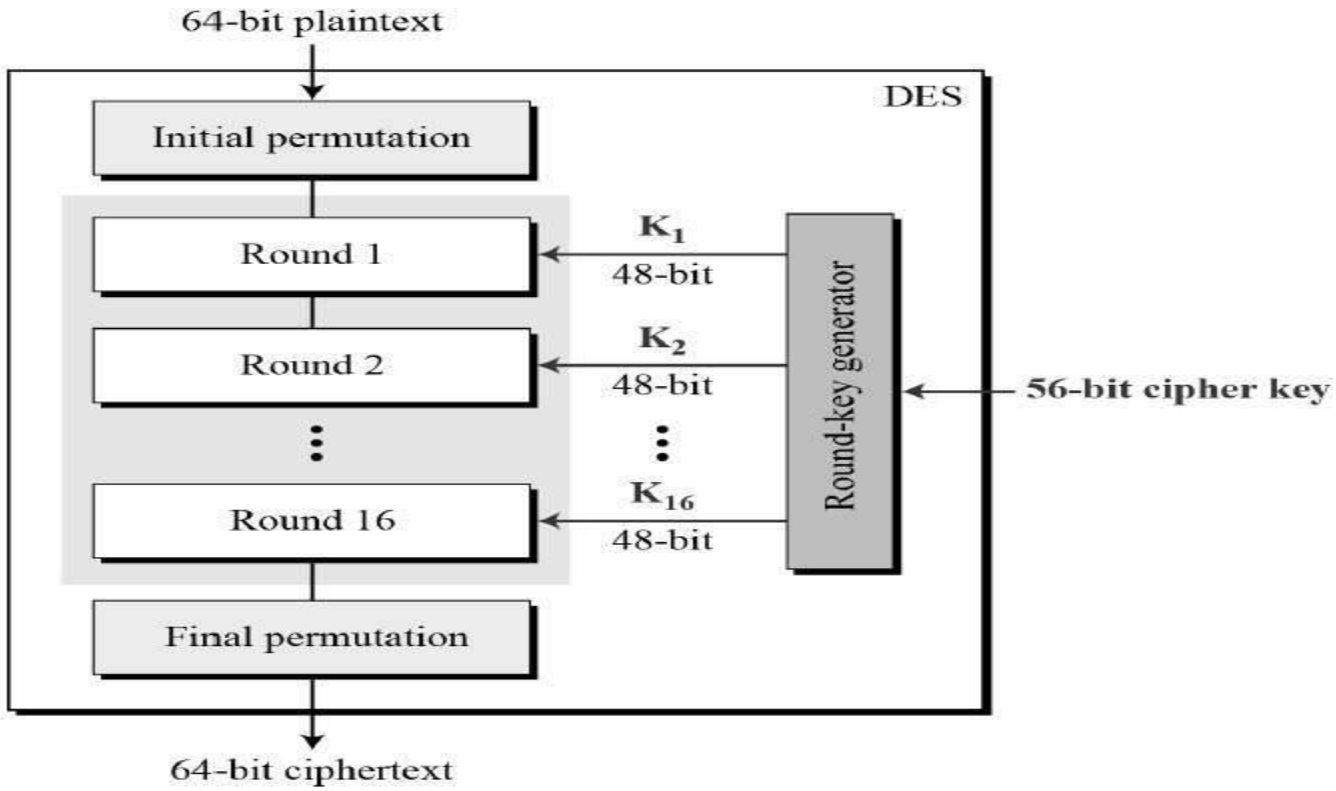
4. Java program for DES algorithm logic

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

DESCRIPTION:

The Data Encryption Standard (DES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST).

DES is an implementation of a Feistel Cipher. It uses 16 round Feistel structure. The block size is 64-bit. Though, key length is 64-bit, DES has an effective key length of 56 bits, since 8 of the 64 bits of the key are not used by the encryption algorithm (function as check bits only). General Structure of DES is depicted in the following illustration −

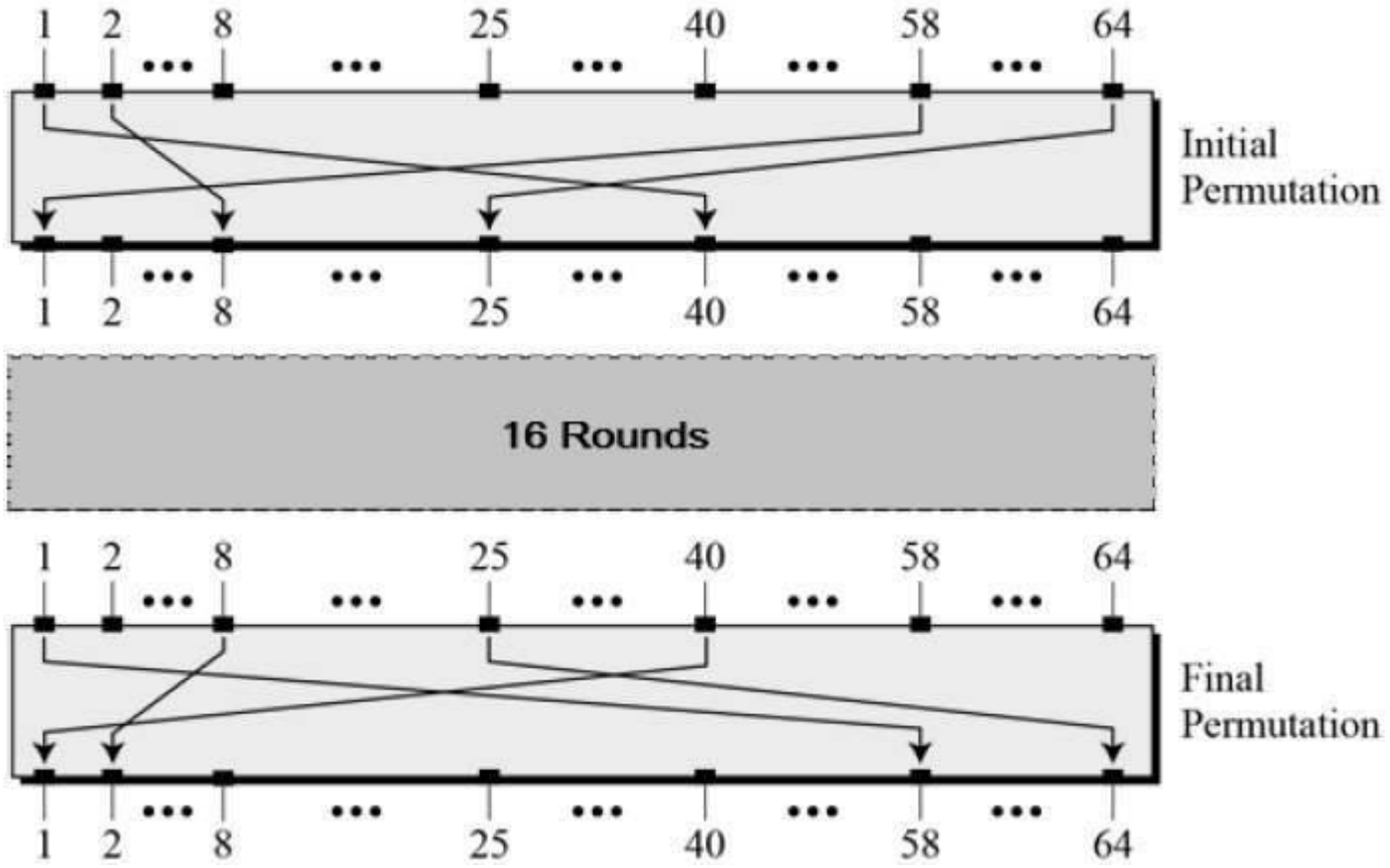


Since DES is based on the Feistel Cipher, all that is required to specify DES is −

- Round function
- Key schedule
- Any additional processing − Initial and final permutation

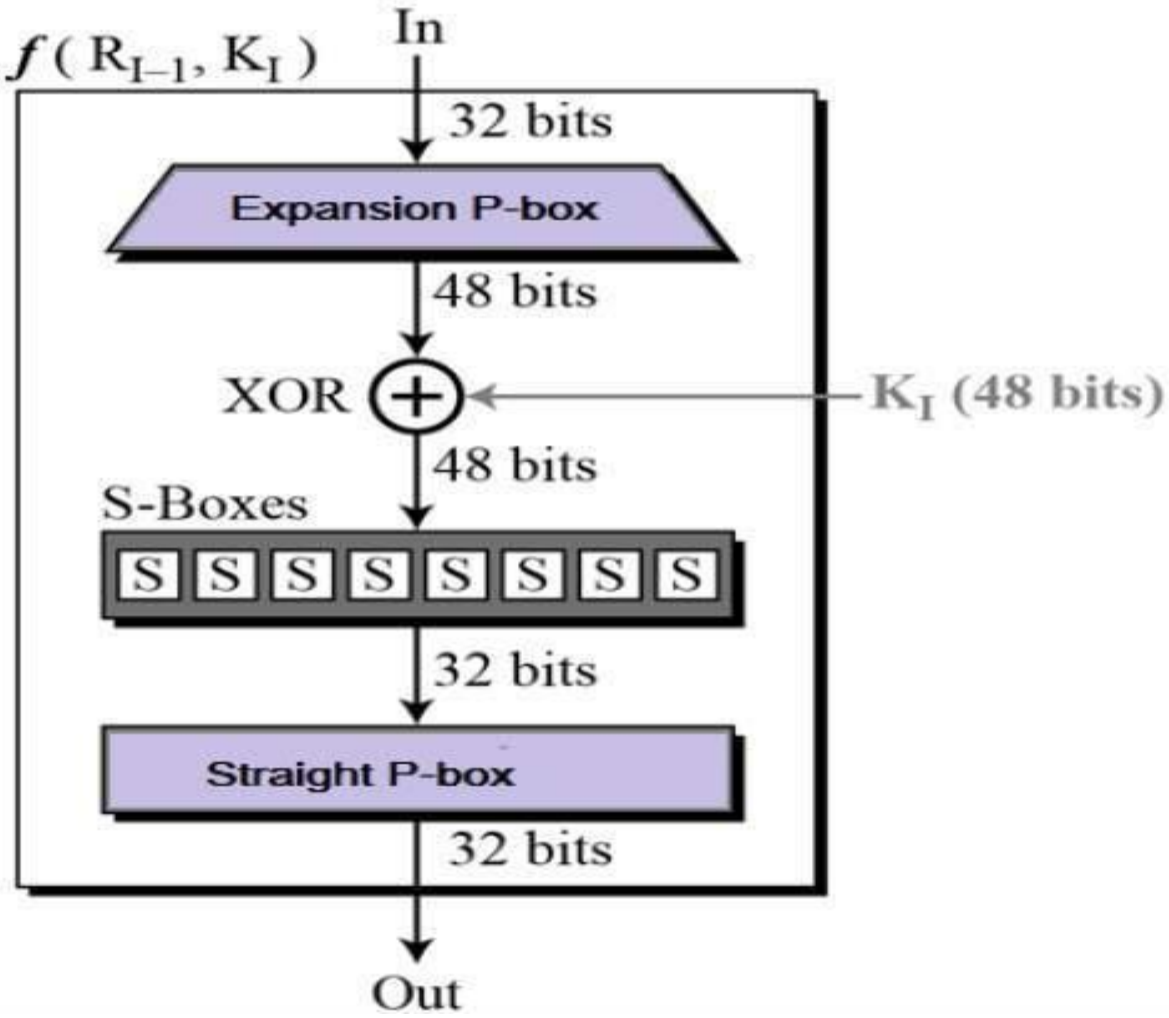
Initial and Final Permutation

The initial and final permutations are straight Permutation boxes (P-boxes) that are inverses of each other. They have no cryptography significance in DES. The initial and final permutations are shown as follows −

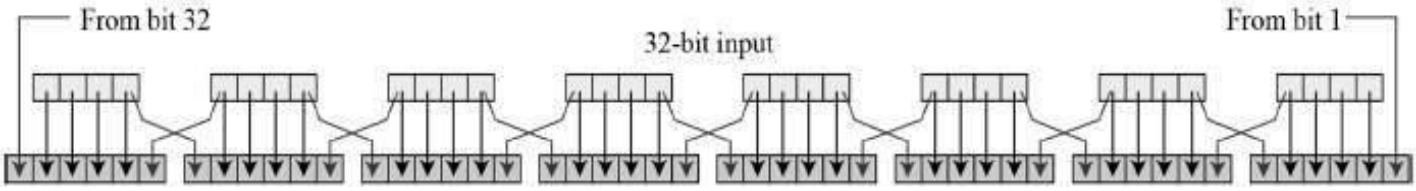


Round Function

The heart of this cipher is the DES function, *f*. The DES function applies a 48-bit key to the rightmost 32 bits to produce a 32-bit output.



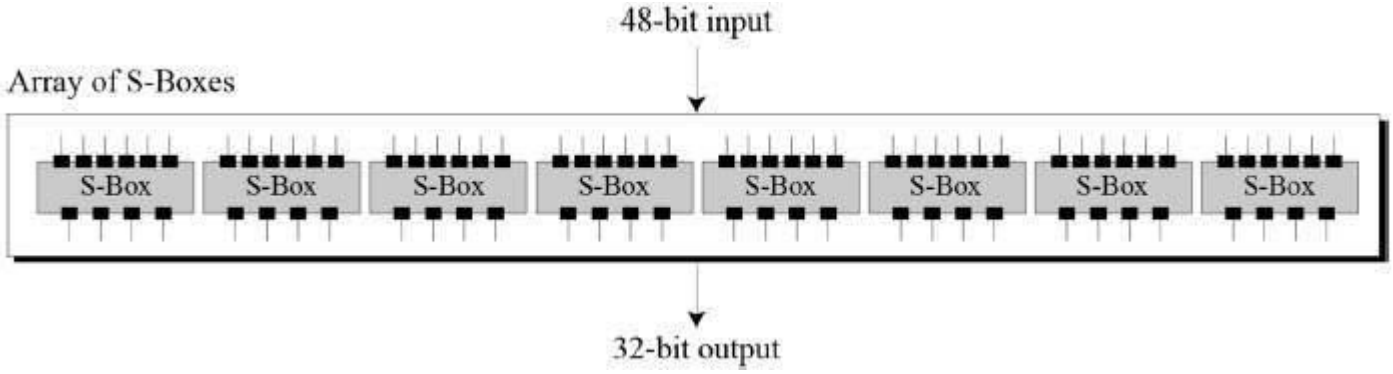
- **Expansion Permutation Box** − Since right input is 32-bit and round key is a 48-bit, we first need to expand right input to 48 bits. Permutation logic is graphically depicted in the following illustration −



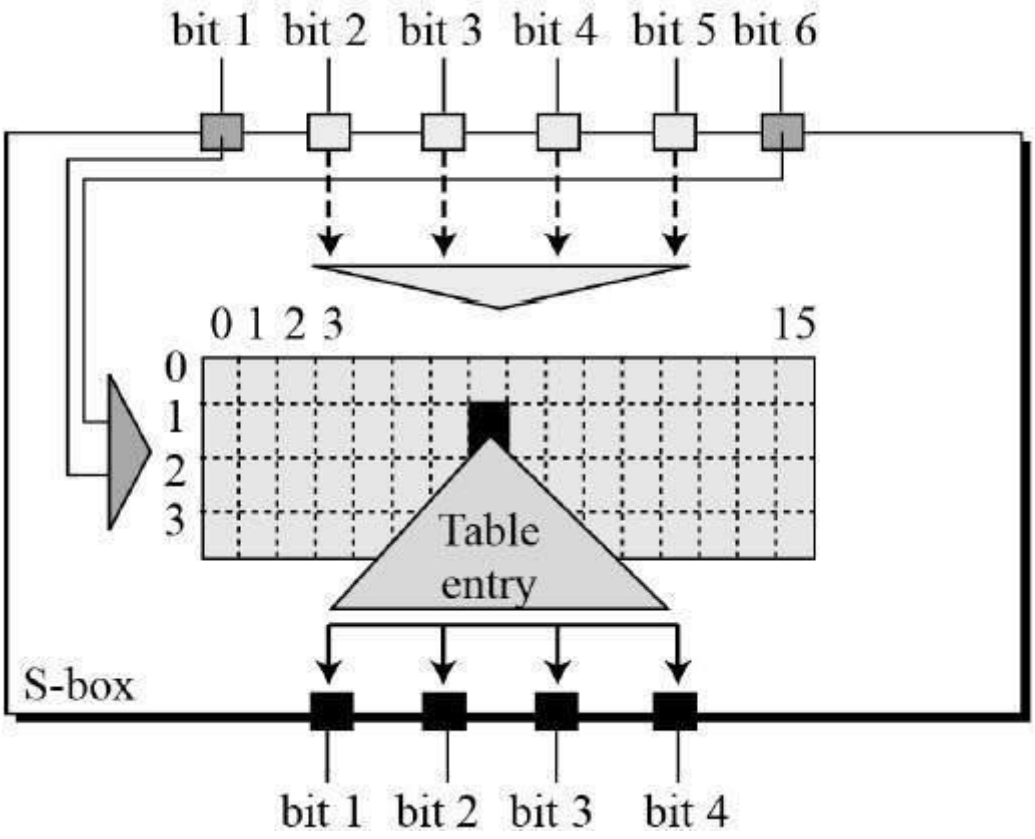
- The graphically depicted permutation logic is generally described as table in DES specification illustrated as shown −

32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	31	31	32	01

- **XOR (Whitener).** − After the expansion permutation, DES does XOR operation on the expanded right section and the round key. The round key is used only in this operation.
- **Substitution Boxes.** − The S-boxes carry out the real mixing (confusion). DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output. Refer the following illustration −



- The S-box rule is illustrated below −

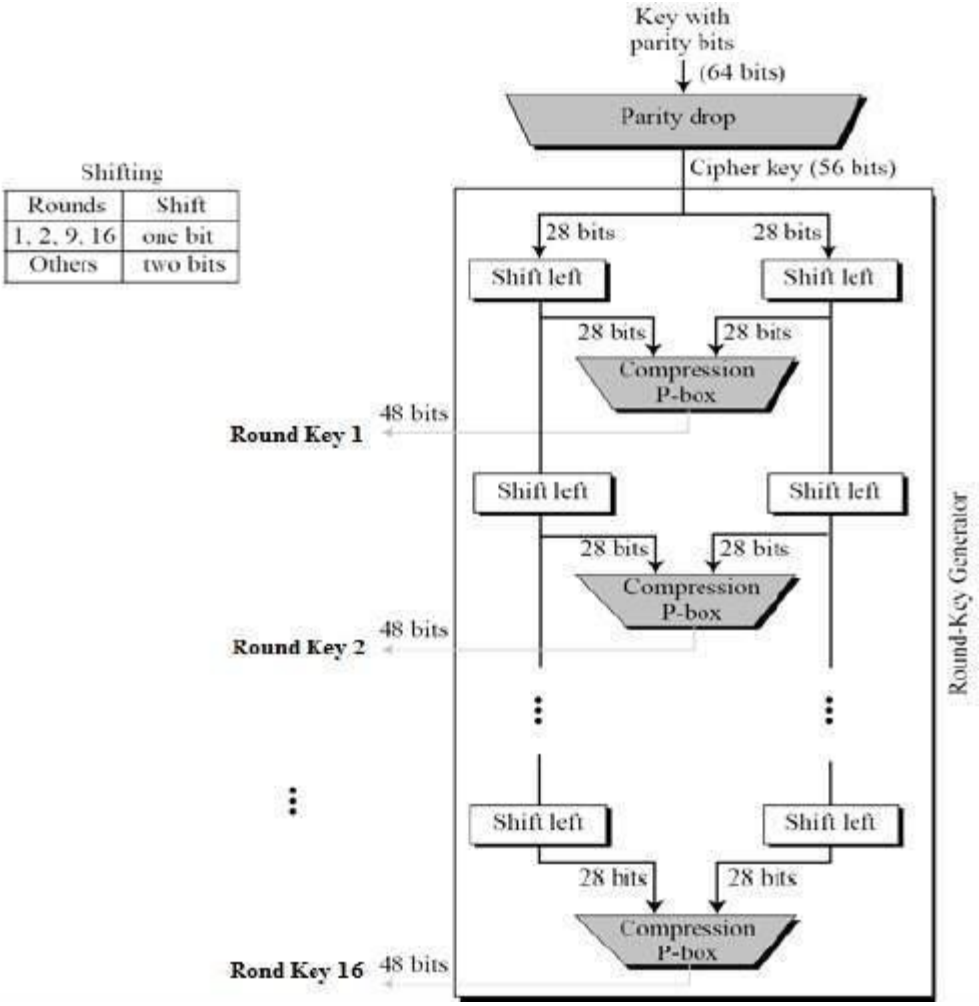


- There are a total of eight S-box tables. The output of all eight s-boxes is then combined in to 32 bit section.
- **Straight Permutation** − The 32 bit output of S-boxes is then subjected to the straight permutation with rule shown in the following illustration:

16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
02	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

Key Generation

The round-key generator creates sixteen 48-bit keys out of a 56-bit cipher key. The process of key generation is depicted in the following illustration −



Shifting

Rounds	Shift
1, 2, 9, 16	one bit
Others	two bits

The logic for Parity drop, shifting, and Compression P-box is given in the DES description.

DES Analysis

The DES satisfies both the desired properties of block cipher. These two properties make cipher very strong.

- **Avalanche effect** − A small change in plaintext results in the very great change in the ciphertext.
- **Completeness** − Each bit of ciphertext depends on many bits of plaintext.

During the last few years, cryptanalysis have found some weaknesses in DES when key selected are weak keys. These keys shall be avoided.

DES has proved to be a very well designed block cipher. There have been no significant cryptanalytic attacks on DES other than exhaustive key search.

PROGRAM:

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

    }
```

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

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

OUTPUT:

```
Enter the string: Welcome
String To Encrypt: Welcome
Encrypted Value : BPQMwc0wKvg=
Decrypted Value : Welcome
```

Viva Questions:

1. Compare stream cipher with block cipher with example.
   Stream cipher: Processes the input stream continuously and producing one element at a time.
   Example: caeser cipher. Block cipher: Processes the input one block of elements at a time producing an output block for each input block. Example: DES.

2. Differentiate unconditionally secured and computationally secured .
   An Encryption algorithm is unconditionally secured means, the condition is if the cipher text generated by the encryption scheme doesn't contain enough information to determine corresponding plaintext.
   Encryption is computationally secured means, 1.The cost of breaking the cipher

exceed the value of enough information.
Time required to break the cipher exceed the useful lifetime of information.

3. Define Diffusion & Confusion.
   Diffusion: It means each plaintext digits affect the values of many ciphertext digits which is equivalent to each ciphertext digit is affected by many plaintext digits. It can be achieved by performing permutation on the data. It is the relationship between the plaintext and ciphertext.
   Confusion: It can be achieved by substitution algorithm. It is the relationship between cipher text and key.

4. What are the design parameters of Feistel cipher network?
   *Block size *Key size *Number of Rounds *Sub key generation algorithm *Round function *Fast software Encryption/Decryption *Ease of analysis

5. Define Product cipher.
   It means two or more basic cipher are combined and it produce the resultant cipher is called the product cipher.

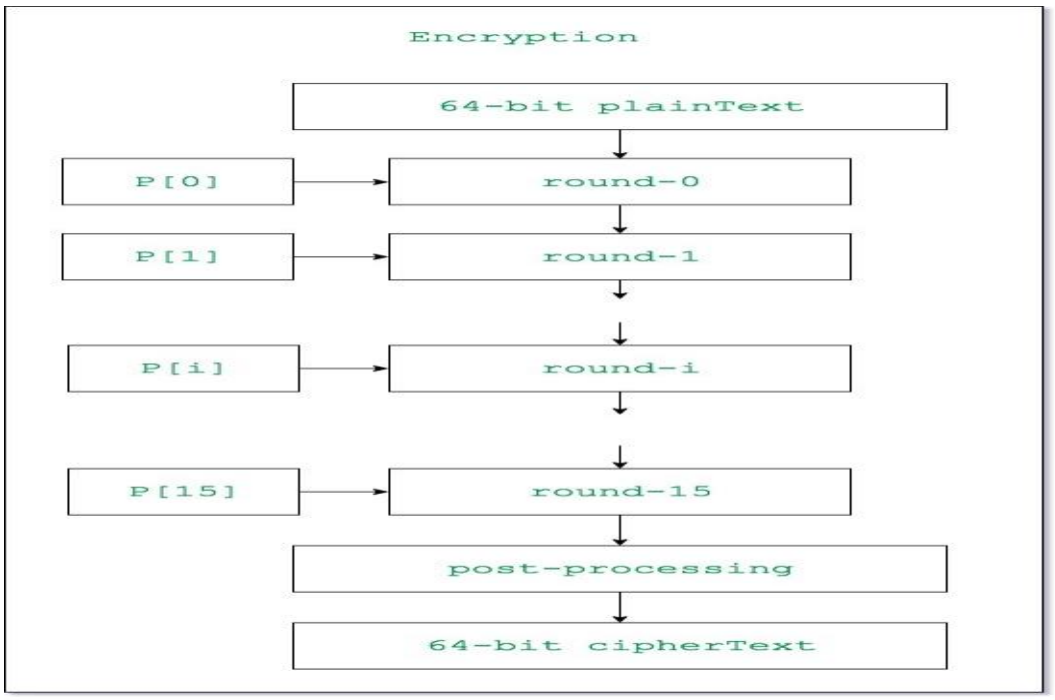
6. Program to implement BlowFish algorithm logic

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

DESCRIPTION:

Blowfish is an encryption technique designed by **Bruce Schneier** in 1993 as an alternative to DES Encryption Technique. It is significantly faster than DES and provides a good encryption rate with no effective cryptanalysis technique found to date. It is one of the first, secure block cyphers not subject to any patents and hence freely available for anyone to use.

1. **BlockSize**: 64-bits
2. **KeySize**: 32-bits to 448-bits variable size
3. **Number of subkeys**: 18 [P-array]
4. **Number of rounds**: 16
5. **Number of substitution boxes**: 4 [each having 512 entries of 32-bits each]



PROGRAM:

```
import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.swing.JOptionPane;
import javax.crypto.*;
import javax.crypto.spec.*;
import java.io.*;

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

  KeyGenerator kgen = KeyGenerator.getInstance("Blowfish");
  Cipher cipher = Cipher.getInstance("Blowfish");
  SecretKey skey = kgen.generateKey();
  byte[] raw = skey.getEncoded();
  SecretKeySpec skeySpec = new SecretKeySpec(raw, "Blowfish");
  cipher.init(Cipher.ENCRYPT_MODE,skey);
  String inputText = JOptionPane.showInputDialog("Input your message: ");
  byte[] encrypted = cipher.doFinal(inputText.getBytes());
  cipher.init(Cipher.DECRYPT_MODE,skey);
  byte[] decrypted = cipher.doFinal(encrypted);
  JOptionPane.showMessageDialog(JOptionPane.getRootFrame(), "\nEncrypted text: " +
new String(encrypted) + "\n" + "\nDecrypted text: " + new String(decrypted));
  System.exit(0);
 }
}
```

OUTPUT:

Enter the string: Welcome

String To Encrypt: Welcome
Encrypted Value : BPQMwc0wKvg=
Decrypted Value : Welcome

Viva Questions:

1. Why do we use Blowfish algorithm?
Blowfish is an encryption technique designed by Bruce Schneier in 1993 as an alternative to DES Encryption Technique. It is significantly faster than DES and provides a good encryption rate with no effective cryptanalysis technique found to date.

2. Is Blowfish a hashing algorithm?
Blowfish is not a hashing algorithm. Its an encryption algorithm. What that means is that you can encrypt something using blowfish, and then later on you can decrypt it back to plain text.

3. What are the demerits of the Blowfish algorithm?
Disadvantages of Blowfish: The small block size of Blowfish (64 bits) is more vulnerable to birthday attacks than the 128 bits used by AES.

4. Why Blowfish is more secure?
Blowfish is an encryption algorithm, or cipher, specifically a block cipher. Blowfish has a 64-bit block size and it supports key lengths of 32-448 bits. No successful cryptanalysis of Blowfish is known, making it secure.

5. What advantage does the Blowfish hashing algorithm have over many others?
Blowfish is license-free and is available free for all uses. It is also a symmetric block cipher that can be used as a drop-in replacement for DES or IDEA. It takes a variable-

length key, from 32 bits to 448 bits, making it ideal for both domestic and exportable use

7. Program to implement Rijndael algorithm logic

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

DESCRIPTION:

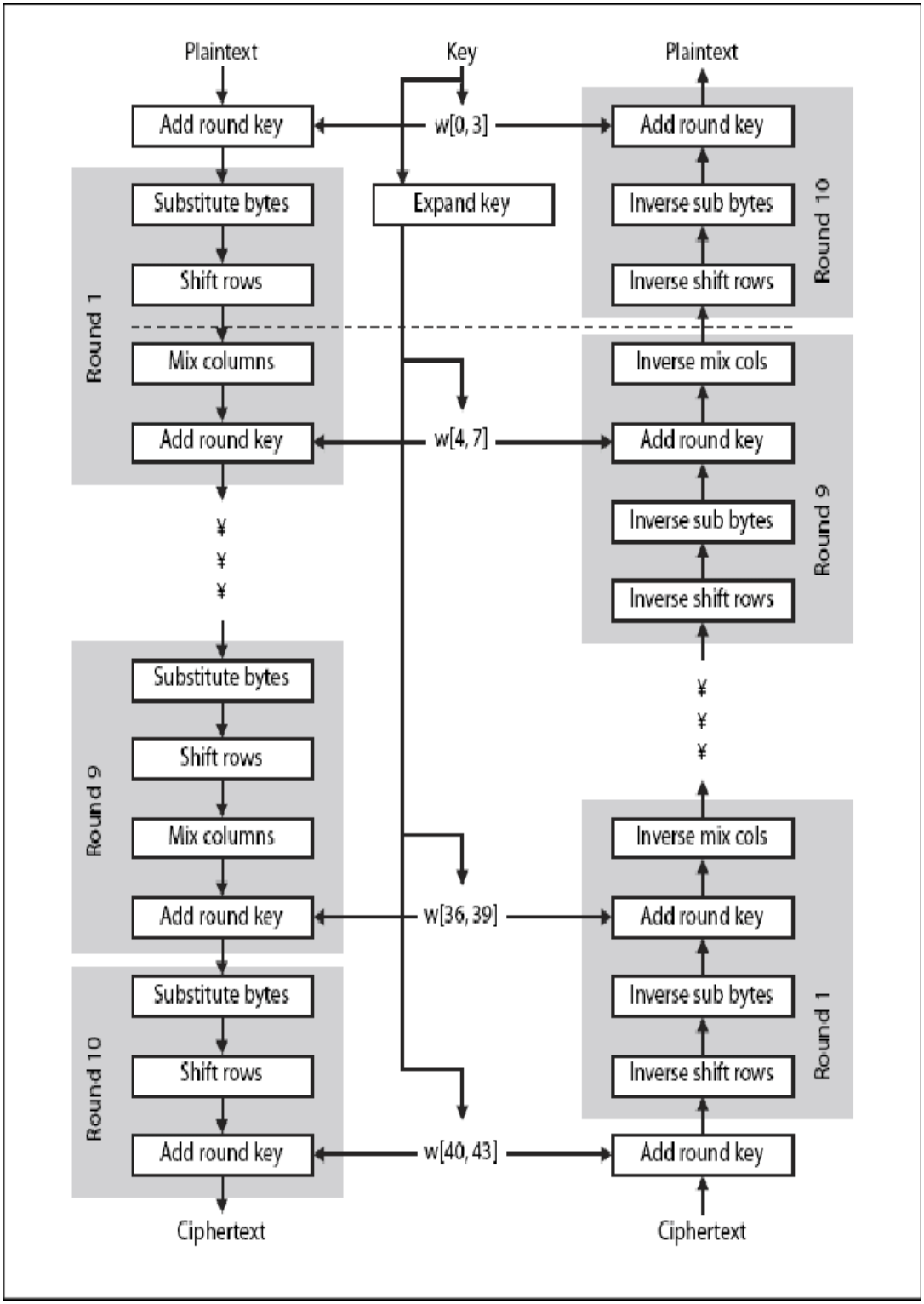
AES is widely used today as it is a much stronger than DES and triple DES despite being harder to implement.

- AES is a block cipher.
- The key size can be 128/192/256 bits.
- Encrypts data in blocks of 128 bits each.

That means it takes 128 bits as input and outputs 128 bits of encrypted cipher text as output. AES relies on substitution-permutation network principle which means it is performed using a series of linked operations which involves replacing and shuffling of the input data.

AES performs operations on bytes of data rather than in bits. Since the block size is 128 bits, the cipher processes 128 bits (or 16 bytes) of the input data at a time.
The number of rounds depends on the key length as follows :

- 128 bit key – 10 rounds
- 192 bit key – 12 rounds
- 256 bit key – 14 rounds


Plaintext
Key
Plaintext
Add round key
w[0, 3]
Add round key
Substitute bytes
Expand key
Inverse sub bytes
Round 10
Shift rows
Inverse shift rows
Round 1
Mix columns
Inverse mix cols
Add round key
w[4, 7]
Add round key
Inverse sub bytes
Round 9
Inverse shift rows
Substitute bytes
Shift rows
Round 9
Mix columns
Inverse mix cols
Add round key
w[36, 39]
Add round key
Substitute bytes
Inverse sub bytes
Round 1
Shift rows
Inverse shift rows
Round 10
Add round key
w[40, 43]
Add round key
Ciphertext
Ciphertext

PROGRAM:

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

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

OUTPUT:

Input your message: Hello KGRCET

Encrypted text: 3ooo&&(*&*4r4

Decrypted text: Hello KGRCET

Viva Questions:

1.What is the difference between Rijndael and AES?

AES is a reduced version of Rijndael where it is only defined for block sizes of 128 bit whereas Rijndael is defined for block sizes of 128, 192 and 256 bits. If a different block size between encryption and decryption is used, then it is not possible to recover the original plaintext.

2. How S-box is created in AES?

The S-BOX is designed by using Advanced Encryption Standard (AES). The AES is a symmetric key standard for encryption and decryption of blocks of data. In encryption, the AES accepts a plaintext input, which is limited to 128 bits, and a key that can be specified to be 128 bits to generate the Cipher text.

3. What is the difference between SubBytes and SubWord?

Sub Bytes uses an S-box to perform a byte-by-byte substitution and SubWord performs a byte substitution on each byte of its input word using the Sbox.

4. What is the difference between the AES decryption algorithm and the equivalent inverse cipher?

In AES decryption, inverse shift rows inverse sub bytes, add round key, inverse mix columns are the operations. But in equivalent inverse cipher, we interchange inverse shift rows and inverse sub bytes.

5. When Should AES be used?

The AES algorithm successively applies a series of mathematical transformations to each 128-bit block of data. Because the computational requirements of this approach are low, AES can be used with consumer computing devices such as laptops and smartphones, as well as for quickly encrypting large amounts of data.

8. RSA Algorithm

AIM: Write a Java program to implement RSA Algorithm.

PROGRAM:

```
#include<stdio.h>
#include<stdlib.h>


int modfun(int a,int n,int b){
if(b==1){
return a%n;
}
else{
return ((a%n)*modfun(a,n,b-1))%n;
}
}

int gcd(int a,int b){
if(a==0){
return b;
}
else if(b==0){
return a;
}
else if(a==b){
return a;
}
else if(a>b) return gcd(a-b,b);
else return gcd(a,b-a);
}
```

```
int totient(int n){
int count=0;
for(int i=1;i<=n;i++){
if(gcd(n,i)==1){
count++;
}
}
return count;
}


void main(){
int a,b,n,e,d,phi,m;
int em,dm;
printf("enter two prime numbers\n");
scanf("%d %d",&a,&b);
n=a*b;
phi=totient(n);
printf("%d",phi);
/*phi=(a-1)*(b-1);*/
printf("\nenter e value: ");
scanf("%d",&e);
if(0<e<n && gcd(e,phi)==1){
for(int i=2;i<n;i++){
if((e*i)%phi==1){
printf("d value is: %d",i);
d=i;
break;
}
```

```
}
printf("\nKU {%d %d}\n",e,n);
printf("KR {%d %d}",d,n);

printf("\nEnter message\n");
scanf("%d",&m);
em=modfun(m,n,e);
dm=modfun(em,n,d);

printf("encrypted message %d\n",em);
printf("decrypted message %d\n",dm%n);

}
else{
printf("invalid e value");
}
}
```

OUTPUT:

```
Enter 2 Prime numbers: 17 11
Phi 160
Enter e value: 7
d value is: 23

KU  {7,187}

KR  {23,187}


Enter message  88

Encrypted message 11

Decrypted message 88
```

Viva Questions:

1. List out the Four possible approaches to attacking the RSA algorithm.

Brute force: This involves trying all possible private keys.

• Mathematical attacks: There are several approaches, all equivalent in effort to factoring the product of two primes.
• Timing attacks: These depend on the running time of the decryption algorithm.
• Chosen ciphertext attacks: This type of attack exploits properties of the RSA algorithm.

2. How is RSA used?

RSA is still seen in a range of web browsers, email, VPNs, chat and other communication channels. RSA is also often used to make secure connections between VPN clients and VPN servers. Under protocols like OpenVPN, TLS handshakes can use the RSA algorithm to exchange keys and establish a secure channel

3. What are the advantages of RSA algorithm?

It is very easy to implement RSA algorithm. RSA algorithm is safe and secure for
transmitting confidential data. Cracking RSA algorithm is very difficult as it involves complex mathematics. Sharing public key to users is easy.

4. Why is RSA better than AES?

Because there is no known method of calculating the prime factors of such large numbers, only the creator of the public key can also generate the private key required for
decryption. RSA is more computationally intensive than AES, and much slower. Its normally
used to encrypt only small amounts of data.

5. What are the weaknesses of RSA?

- Weak growth.
- Poverty, growing inequalities, high unemployment (especially among young people),

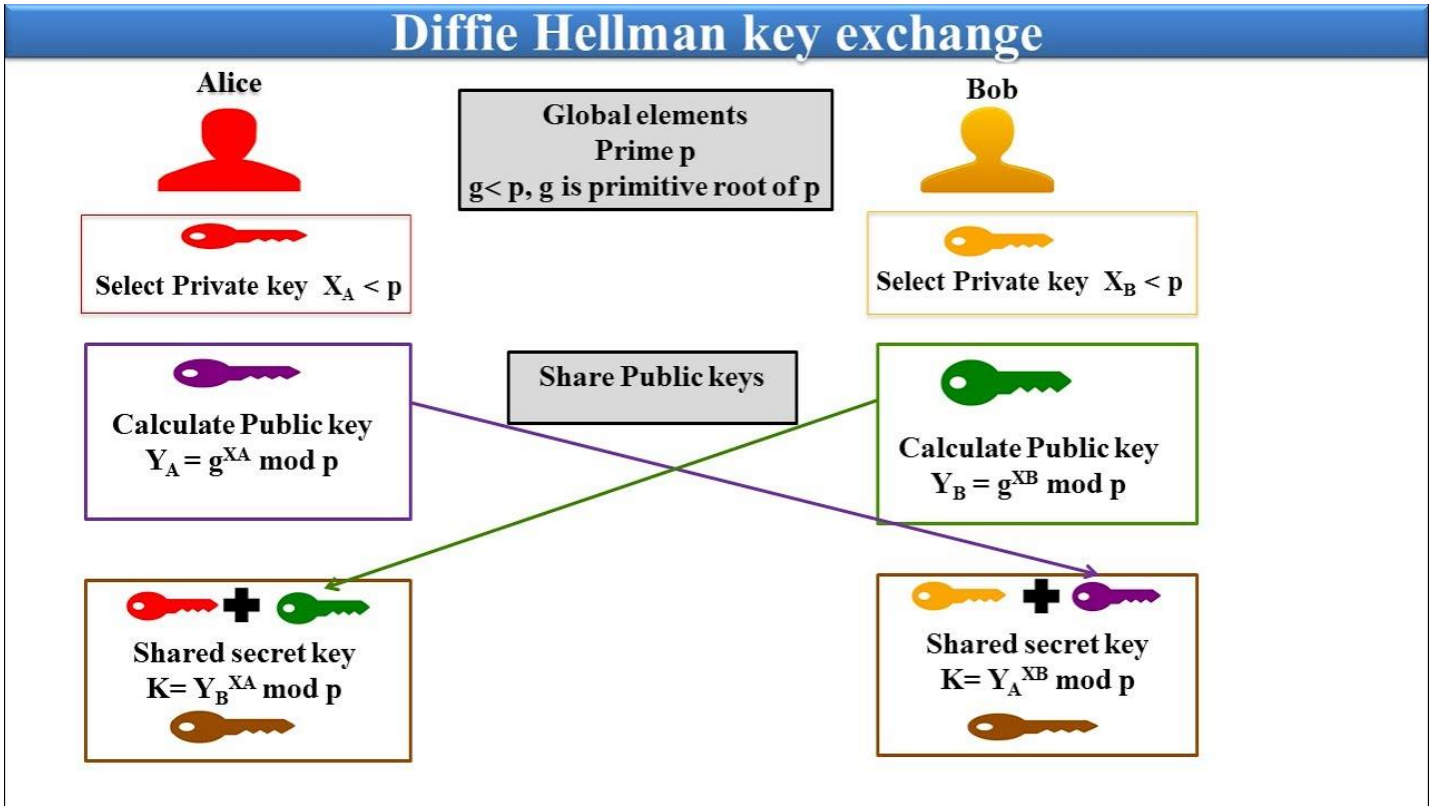
sources of social risk (crime, strikes and demonstrations).

- Skill shortages, labour market rigidity.
- Low efficiency of public spending, corruption.

- **Diffie-Hellman**

AIM: Implement the Diffie-Hellman Key Exchange Mechanism.

DESCRIPTION:



Example:

Alice and Bob both use public numbers P = 23, G = 5

2. Alice selected private key a = 4, and Bob selected b = 3 as the private key

3. Both Alice and bob now calculate the value of x and y as follows:

- Alice: $x = (5^4 \bmod 23) = 4$

- Bob: $y = (5^3 \bmod 23) = 10$

4. Now, both Alice and Bob exchange public numbers with each other.

5. Alice and Bob now calculate the symmetric keys

- Alice: $k_a = y^a \bmod p = 10^4 \bmod 23 = 18$
- Bob: $k_b = x^b \bmod p = 4^3 \bmod 23 = 18$

6. 18 is the shared secret key.

PROGRAM:

```
import java.io.*;
import java.util.Scanner;
public class DHKE{
public static void main(String[] args){
int q,alpha_picked,xa,xb,ya,yb,ka,kb,index=0;
int alpha [] =new int[100];
System.out.println("Enter the prime : ");
Scanner sc=new Scanner(System.in);
q=sc.nextInt();
for(int i=2;i<q;i++){
int alpharnot[]=new int[q];
for (int j=1;j<=q;j++){
alpharnot[j-1]=(int)((java.lang.Math.pow(i,j))%q);
int c=0;
for(int k=0;k<q;k++){
```

```
for(int p=k+1;p<q;p++){
if(alpharnot[k]==alpharnot[p]){c++;}
}
}
if(c==0){
alpha[index]=i;index++;
}
}
}
for(int i=0;i<index;i++){
System.out.println("Primitive root is  : "+ alpha[i]);
}
System.out.println("Select one of the root : ");
alpha_picked=sc.nextInt();
System.out.println("Select  Xa: ");
xa=sc.nextInt();
System.out.println("Select  Xb: ");
xb=sc.nextInt();
ya=(int)((java.lang.Math.pow(alpha_picked,xa))%q);
yb=(int)((java.lang.Math.pow(alpha_picked,xb))%q);
ka=(int)((java.lang.Math.pow(yb,xa))%q);
kb=(int)((java.lang.Math.pow(ya,xb))%q);
System.out.println("Ka: "+ka+" Kb :"+kb);
if(ka==kb){
System.out.println("Keys are same");
}
}
}
```

OUTPUT:

Enter the Prime: 5

Primitive Root is: 2

Primitive Root is: 3

Select one of the Root: 2

Select Xa: 2

Select Xb: 3

Ka: 4 Kb: 4

Keys are same

Viva Questions:

1. What algorithm does Diffie Hellman use?

Diffie Hellman uses a private-public key pair to establish a shared secret, typically a symmetric key. DH is not a symmetric algorithm – it is an asymmetric algorithm used to establish a shared secret for a symmetric key algorithm.

2. Is Diffie-Hellman key exchange vulnerable to man-in-the-middle attack?

The Diffie-Hellman key exchange is vulnerable to a man-in-the-middle attack. In this attack, an opponent Carol intercepts Alice public value and sends her own public value to Bob. When Bob transmits his public value, Carol substitutes it with her own and sends it to Alice.

3. Where is Diffie-Hellman used?

The Diffie-Hellman algorithm will be used to establish a secure communication channel. This channel is used by the systems to exchange a private key. This private

key is then used to do symmetric encryption between the two system.

4.Which is better Diffie-Hellman or RSA?

RSA can be mixed with ECC to improve security and performance. DH can be integrated

with digital and public key certificates to prevent attacks.

5. What is the most significant feature of the Diffie Hellman exchange?

The Diffie-Hellman key-exchange algorithm is a secure algorithm that offers high

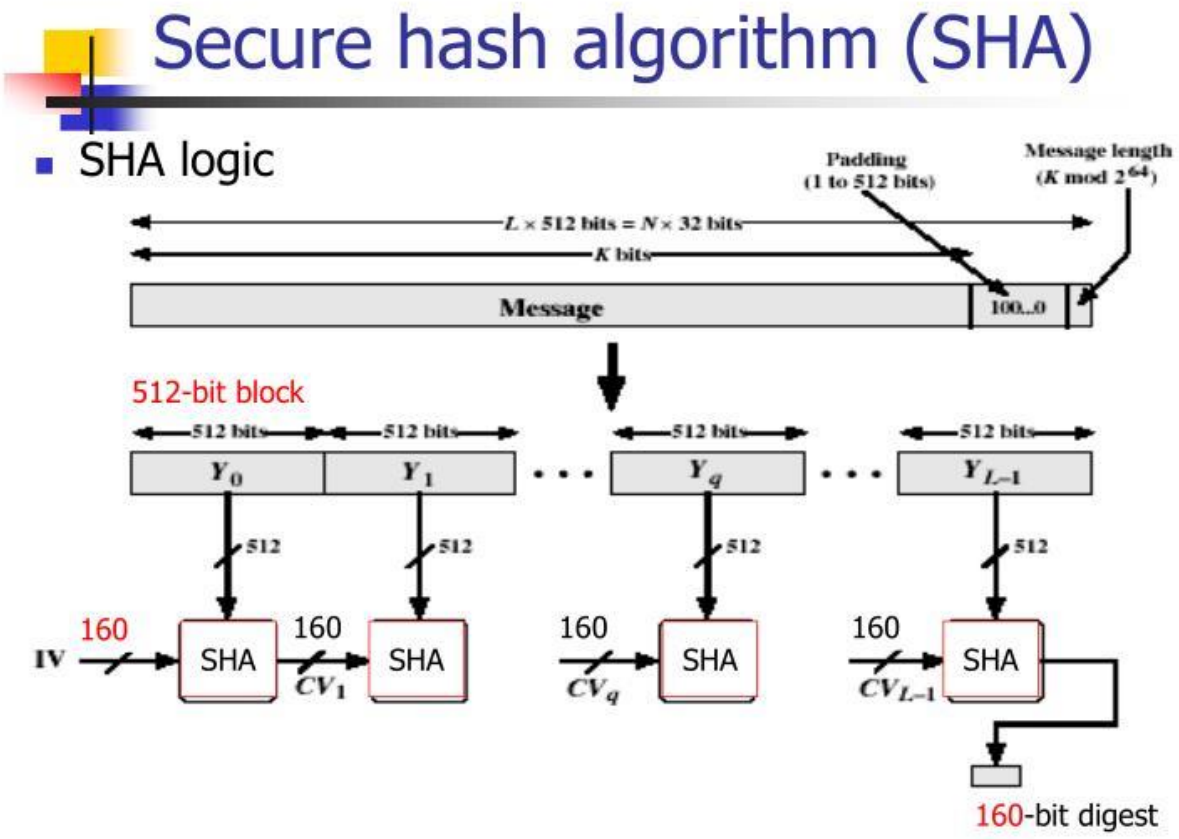
performance, allowing two computers to publicly exchange a shared value without using data encryption. This exchanged information is protected with a hash function.

- **SHA-1**

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

DESCRIPTION:

Security Hash Algorithm (SHA) was developed in 1993 by the National Institute of Standards and Technology (NIST) and National Security Agency (NSA). It was designed as the algorithm to be used for secure hashing in the US Digital Signature Standard. Hashing function is one of the most commonly used encryption methods. A hash is a special mathematical function that performs one-way encryption. SHA-l is a revised version of SHA designed by NIST and was published as a Federal Information Processing Standard (FIPS). Like MD5, SHA-l processes input data in 512-bit blocks. SHA-l generates a 160-bit message digest. Whereas MD5 generated message digest of 128 bits.



PROGRAM:

```
import java.security.*;
import java.math.BigInteger;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

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'};
  StringBuffer buf = 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 = SHA1
Provider = SUN version 1.6
ToString = SHA1 Message Digest from SUN, <initialized> SHA1("") =
DA39A3EE5E6B4B0D3255BFEF95601890AFD80709 SHA1("abc") =
A9993E364706816ABA3E25717850C26C9CD0D89D
SHA1("abcdefghijklmnopqrstuvwxyz")=32D10C7B8CF96570CA04CE37F2A19D8424D3A89

Viva Questions:

1.What characteristics are needed in a secure hash function?

There should be a fixed length output generated by the hash function.

There should be easiness for finding the hash function for a certain or given message.

There should not be same hash value for the two different messages.

2. What is the role of a compression function in a hash function?

A compression function takes a fixed length input and returns a shorter, fixed-length output. The blocks are then processed sequentially, taking as input the result of the hash

so
far and the current message block, with the final output being the hash value for the message.

3. What basic arithmetical and logical functions are used in SHA?
Basic arithmetical and logical functions in SHA are circular shifts , AND , OR , NOT , and XOR.

4. Why SHA algorithm is used?
Secure Hash Algorithms, also known as SHA, are a family of cryptographic functions designed to keep data secured. A common application of SHA is to encrypting passwords, as the server side only needs to keep track of a specific user hash value, rather than the actual password.

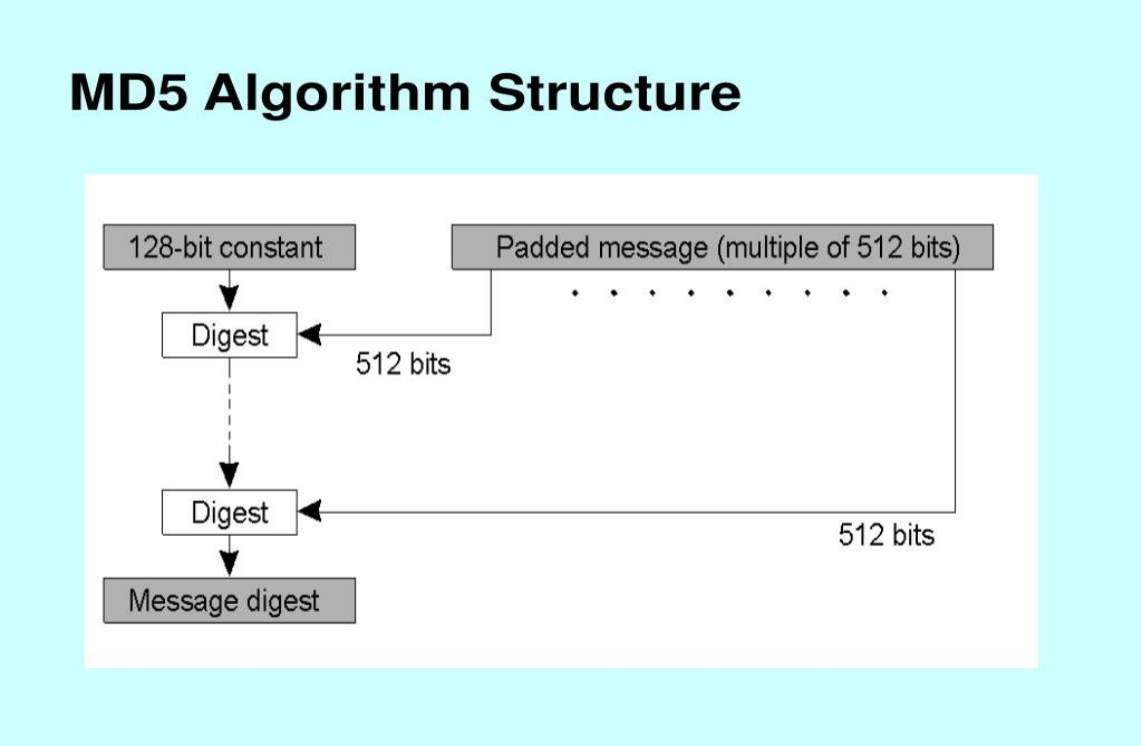
5. Is SHA symmetric or asymmetric?
Since it uses Public and Private key, SHA-256 is Asymmetric. it is a HASHING algorithm used to ensure INTEGRITY, AUTHENTICITY, NON-REPUDIATION.

- **Message Digest Algorithm5 (MD5)**

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

DESCRIPTION:

MD5 message-digest algorithm is the 5th version of the Message-Digest Algorithm developed by Ron Rivest to produce a 128-bit message digest. MD5 is quite fast than other versions of the message digest, which takes the plain text of 512-bit blocks, which is further divided into 16 blocks, each of 32 bit and produces the 128-bit message digest, which is a set of four blocks, each of 32 bits. MD5 produces the message digest through five steps, i.e. padding, append length, dividing the input into 512-bit blocks, initialising chaining variables a process blocks and 4 rounds, and using different constant it in each iteration.



PROGRAM:

```
import java.security.*;
public class MD5 {
```

```
public static void main(String[] a) {
        // TODO code application logic here
try {
MessageDigest md = MessageDigest.getInstance("MD5");
System.out.println("Message digest object info: ");
System.out.println(" Algorithm = " +md.getAlgorithm());
System.out.println(" Provider = " +md.getProvider());
System.out.println(" ToString = " +md.toString());

            String input = "";
md.update(input.getBytes());
byte[] output = md.digest();
System.out.println();
System.out.println("MD5(\""+input+"\") = " +bytesToHex(output));

input = "abc";
md.update(input.getBytes());
output = md.digest();
System.out.println();
System.out.println("MD5(\""+input+"\") = " +bytesToHex(output));

input = "abcdefghijklmnopqrstuvwxyz";
md.update(input.getBytes());
output = md.digest();
System.out.println();
System.out.println("MD5(\"" +input+"\") = "
+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

Viva questions

1. How MD5 works step by step?

Step1: Append Padding Bits. Padding means adding extra bits to the original message.

Step 2: Append Length. After padding, 64 bits are inserted at the end, which is used to

record the original input length.

Step 3: Initialize MD buffer.

Step 4: Processing message in 16-word block.

2. What type of algorithm best describes MD5?

MD5 is a type of algorithm that is known as a cryptographic hash algorithm. MD5

produces a hash value in a hexadecimal format. This competes with other designs where hash functions take in a certain piece of data, and change it to provide a key or value that can be used in place of the original value.

3. Is MD5 a secure hashing algorithm?

Unfortunately, MD5 has been cryptographically broken and considered insecure. For this reason, it should not be used for anything. Instead, developers should switch to the Secure Hash Algorithm or a Symmetric Cryptographic Algorithm.

4. Is MD5 more secure than SHA1?

Both MD5 stands for Message Digest and SHA1 stands for Secure Hash Algorithm square measure the hashing algorithms wherever The speed of MD5 is fast in comparison of SHA1speed. However, SHA1 provides more security than MD5.

5. What is the most secure hashing algorithm?

Common attacks like brute force attacks can take years or even decades to crack the hash digest, so SHA-2 is considered the most secure hash algorithm.