Central University of Tamilnadu

Department of Computer Science

NETWORK AND SYSTEM SECURITY LAB

OBJECTIVES:

The student should be made to:

➢ **Learn to implement the algorithms**

DES (Data Encryption Standard),

RSA (Rivest–Shamir–Adleman),

MD5 (Message-Digest Algorithm),

SHA-1 (Secure Hash Algorithms)

➢ **Learn to use network security tools like**

GnuPG

KF sensor (*KFSensor* is a commercial host based Intrusion Detection System (IDS))

Net Stumbler *(Network* Stumbler)

OUTCOMES:

At the end of the course, the student should be able to:

➢ Implement the cipher techniques

➢ Develop the various security algorithms

➢ Use different open source tools for network security and analysis

LIST OF HARDWARE REQUIREMENTS & SOFTWARE REQUIREMENTS SOFTWARE REQUIREMENTS

➢ C

➢ C++

➢ Java or equivalent compiler GnuPG

➢ KF Sensor or Equivalent

➢ Snort

➢ Net Stumbler or Equivalent

HARDWARE REQUIREMENTS

➢ Standalone desktops (or) Server supporting 30 terminals or more

Use online compiler and debugger for c/c++ /Java etc

https://www.onlinegdb.com/online_c_compiler

Online C Compiler.

Code, Compile, Run and Debug C program online. Write your code in this editor and press "Run" button to compile and execute it.

LAB EXCERCISE 1

Mathematical Background of CAESAR CIPHER

Introduction:

Data Encryption and Decryption is done using various Techniques.

One of the old and Well Known Technique is called as CAESAR CIPHER.

Cesar Cipher is basic Building block of other techniques.

How it is working:

When you transfer Password from One Computer to another computer that can be easily hacked.

Encryption: Conversion of Plain Text to Cyber Text

Decryption: Conversion of Cyber Text to Plain Text

Consider the following Plain Text

P=Plain Text

C=Cyber Text

K=Key (Used to Convert Plain text to cyber and Cyber to Plain Text)

Solution:

P=HELLOW WORLD

K=3

C=(P+K)Mod 26

Let us find cyber text of Each letter.

To solve this write letter from A to Z.

A	B	C	D	E	F	G	H
0	1	2	3	4	5	6	7
I	J	K	L	M	N	O	P
8	9	10	11	12	13	14	15
Q	R	S	T	U	V	W	X
16	17	18	19	20	21	22	23
Y	Z						
24	25						

C=(P+K)Mod 26

= (7+3) Mod 26

= 10 Mod 26   (10 is smaller than 26. Therefore10 is considered)

C= 10

C=K

For the Plain Text H , the equivalent cyper text is K

Consider the second character, E

C=(P+K)Mod 26

= (4+3) Mod 26

= 7 Mod 26

C   = 7

C= H

Similarly,

For Plain text L= O is equivalent cyber text

For Plain text L= O is equivalent cyber text

For Plain text W= R is equivalent cyber text

C=KHOOR

Plain Text (HELLOW)=Cyber Text (KHOOR)

<u>Decryption</u>

Conversion of Cyber text in to Plain Text

P=(C-K) Mod 26

C=K; C=H; C=O; C=O and C=R

P= (10-3) Mod 26

= 7 Mod 26

= 7

P=7

P=H

P=(C-K) Mod 26

= (7-3) Mod 26

= 4 Mod 26

P= 4

P= E

Similarly,

P=L

P=L

and

P=W.

Example 1

P=XYZ

C=(P+K)Mod 26

= (23+3) Mod 26

= 26 Mod 26

= 0

C=0

C=A

Cyber Text for letter X=A

C= (P+K)Mod 26

= (24+3) Mod 26

C= 1

= B

Cyber Text for letter X=B

C= (P+K)Mod 26

= (25+3) Mod 26

C= (28) Mod 26

C= 2

C=C

Cyber text=ABC

Example: 2

C=ABC

Write equivalent plain text?

P=(C-K) Mod 26

= (0-3) Mod 26

= (-3) Mod 26 (Whenever you get -ve number just add with 26)

P= 23

The Caesar Cipher algorithm is one of the oldest methods of password encryption and decryption system. It is popular by the following naming conventions:

- Caesar shift
- Caesar’s cipher
- Shift cipher
- Caesar’s code

This Caesar cipher encryption algorithm is a kind of substitution cipher wherein every character in the plain-text or the user input is replaced by another character which is defined with a fixed number of positions away from the existing character.

In the times of Julius Caesar was used only the shift of 3 characters, but nowadays the term Caesar cipher refers to all variants (shifts) of this cryptosystem.

Caesar Cipher Encryption and Decryption Example

Input: ABCDEFGHIJ

Encrypted String: KLMNOPQRST

As you can find out from the encrypted string, we have moved every character’s position by 10 towards the right. You can implement your own complex calculations as well.

EX. NO: 1

IMPLEMENTATION OF CAESAR CIPHER

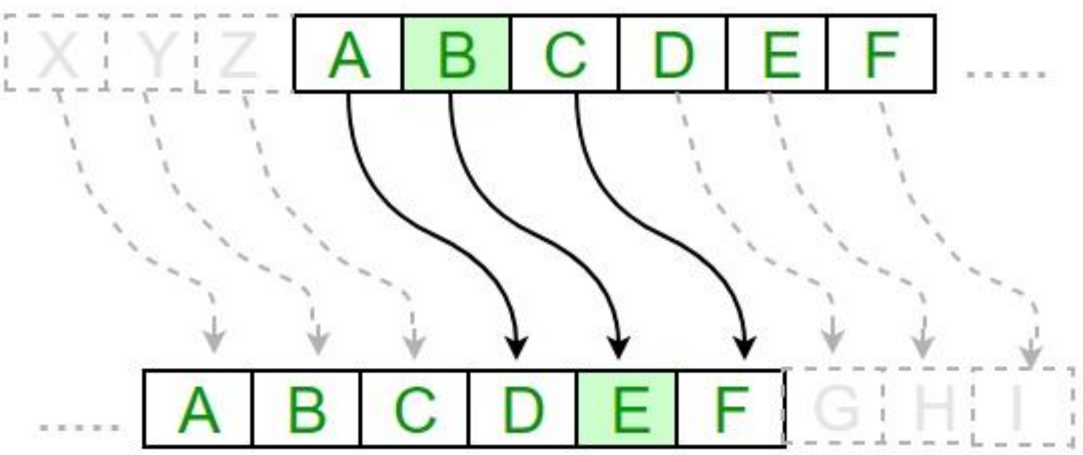
AIM:

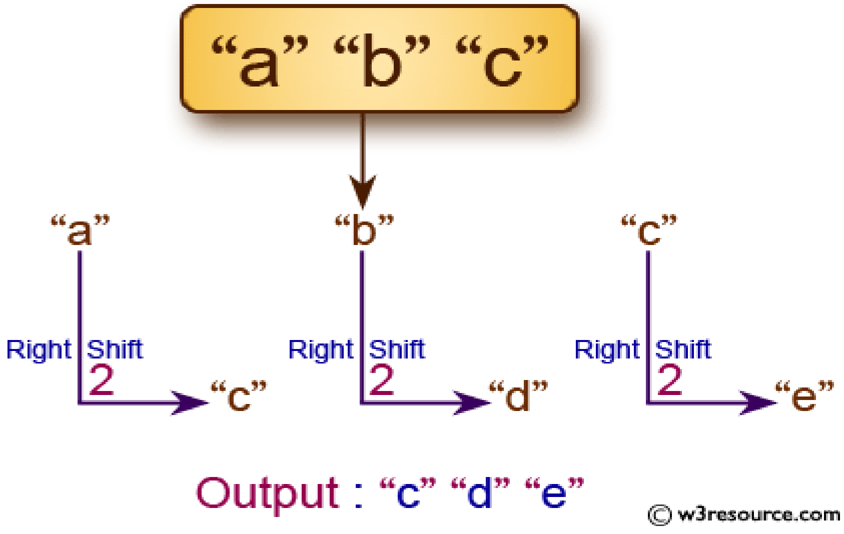
To implement the simple substitution technique named Caesar cipher using C language.

DESCRIPTION:

To encrypt a message with a Caesar cipher, each letter in the message is changed using a simple rule: shift by three. Each letter is replaced by the letter three letters ahead in the alphabet. A becomes D, B becomes E, and so on. For the last letters, we can think of the alphabet as a circle and "wrap around". W becomes Z, X becomes A, Y becomes B, and Z becomes C. To change a message back, each letter is replaced by the one three before it.

EXAMPLE:





ALGORITHM:

Caesar cipher (shift cipher) is a simple substitution cipher based on a replacement of every single character of the open text with a character, which is fixed number of positions further down the alphabet.

The encryption can be described with the following formula:

$$C_i = (T_i + k) \pmod{m}$$

C_i - *i*-th character of the closed text
T_i - *i*-th character of the open text
k - shift
m - length of the alphabet

The Process of decryption uses reverted procedure:

$$T_i = (C_i - k) \pmod{m}$$

STEP-1: Read the plain text from the user.

STEP-2: Read the key value from the user.

STEP-3: If the key is positive then encrypt the text by adding the key with each character in the plain text.

STEP-4: Else subtract the key from the plain text.

STEP-5: Display the cipher text obtained above.

Program

Caesar Cipher in C Language [Encryption]

```
void encrypt(char arr[])
{
    int i;
    for(i = 0; i < strlen(arr); i++)
    {
        arr[i] = arr[i] - 10;
    }
}
```

Caesar Cipher in C Language [Decryption]

```
void decrypt(char arr[])
{
    int i;
    for(i = 0; i < strlen(arr); i++)
    {
        arr[i] = arr[i] + 10;
    }
}
```

```
}
```

Caesar Cipher Encryption and Decryption Program in C

(Sample only)

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
 void decrypt(char arr[ ])
{
    int i;
    for(i = 0; i < strlen(arr); i++)
    {
        arr[i] = arr[i] + 10;
    }
}
 void encrypt(char arr[])
{
    int i;
    for(i = 0; i < strlen(arr); i++)
    {
        arr[i] = arr[i] - 10;
    }
}
```

```
 int main()
{
      char password[40];
      int ch;
      printf("Enter a Password:\t");
      scanf("%s", password);
      printf("\nPassword:\t%s\n",password);
      encrypt(password);
      printf("\nEncrypted Password:\t%s\n", password);
      decrypt(password);
      printf("\nDecrypted Password:\t%s\n", password);
      return 0;
}
```

Expected Output

```
Enter the Plain Text :hellow
Enter the Key Value: 3
Plain Text: hellow
Encrypted Text:
After Decryption:
```

Caesar Cipher Encryption and Decryption Program in Java (Sample only)

```
import java.util.Scanner;

public class ceasercipher
{
public static final String ALPHABET = "abcdefghijklmnopqrstuvwxyz";

public static String encrypt(String plainText, intshiftKey)
    {
plainText = plainText.toLowerCase();
        String cipherText = "";
for (inti = 0; i<plainText.length(); i++)
        {
intcharPosition = ALPHABET.indexOf(plainText.charAt(i));
/*manual prepared by www.gr-solution.blogspot.com*/
intkeyVal = (shiftKey + charPosition) % 26;
charreplaceVal = ALPHABET.charAt(keyVal);
cipherText += replaceVal;
        }
returncipherText;
    }

public static String decrypt(String cipherText, intshiftKey)
    {
cipherText = cipherText.toLowerCase();
        String plainText = "";
for (inti = 0; i<cipherText.length(); i++)
        {
intcharPosition = ALPHABET.indexOf(cipherText.charAt(i));
intkeyVal = (charPosition - shiftKey) % 26;
if (keyVal< 0)
            {
keyVal = ALPHABET.length() + keyVal;
```

```
      }
charreplaceVal = ALPHABET.charAt(keyVal);
plainText += replaceVal;
    }
returnplainText;
  }

public static void main(String[] args)
  {
    Scanner sc = new Scanner(System.in);
System.out.println("Enter the String for Encryption: ");
    String message = new String();
/*manual prepared by www.gr-solution.blogspot.com*/
message = sc.next();
System.out.println("Encryption message=  "+encrypt(message, 3));
System.out.println("Decryption message=  "+decrypt(encrypt(message, 
3), 3));
sc.close();
  }
}
```

Expected Output

```
Enter the Plain Text :hellow

Enter the Key Value: 3

Plain Text: hellow

Encrypted Text:

After Decryption:
```

Caesar Cipher Encryption and Decryption Program in Python (Sample only)

```
def caesar_encrypt(realText, step):
    outText = []
    cryptText = []
    uppercase = ['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']
    lowercase = ['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']

    for each Letter in real Text:
        if each Letter in uppercase:
            index = uppercase.index(eachLetter)
            crypting = (index + step) % 26
            cryptText.append(crypting)
            newLetter = uppercase[crypting]
            outText.append(newLetter)
        elif eachLetter in lowercase:
            index = lowercase.index(eachLetter)
            crypting = (index + step) % 26
            cryptText.append(crypting)
            newLetter = lowercase[crypting]
            outText.append(newLetter)
    return outText

code = caesar_encrypt('abc', 2)
print()
print(code)
print()
```

Sample Output:

```
['c', 'd', 'e']
```