AISSMS
COLLEGE OF ENGINEERING
ज्ञानम् सकलजनहिताय
Accredited by NAAC with "A+" Grade

ALL INDIA SHRI SHIVAJI MEMORIAL SOCIETY'S COLLEGE OF ENGINEERING, PUNE - 01

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

THIRD YEAR

SEMISTER - VI

SUBJECT: NETWORK SECURITY

LABORATORY MANUAL

SUBJECT CODE: 304198 (E)

Name of Student:

Class:

Roll No:

Batch:

ACADEMIC YEAR - 2021-22

PREFACE

This instructor's manual is to provide faculty with a comprehensive teaching package that allows them to enhance student learning with software tool and introduce students to the concept of Network Security system in an exciting way.

This manual is designed to be a powerful but simple-to-use teaching tool. There are a broad range of features that are straight forward, current, relevant and easy to teach from.

The aim is to keep it brief and cover all the skill set required for Network Security study and essentials without burdening students or faculty with unnecessary details.

AISSMS
COLLEGE OF ENGINEERING
ज्ञानम् सकलजनहिताय
Accredited by NAAC with "A+" Grade

PROLOGUE

All India Shri Shivaji Memorial Society, established as early as in 1917, is a premier educational institution in Pune. Ours is a result-oriented Society dedicated to the noble cause of Military, General, Technical & Management Education in India by Shri Chhatrapati Shahu Maharaj of Kolhapur also known as "Rajarshi Shahu". He was the first Maharaja of the princely state of Kolhapur and a great social reformer.

He was an invaluable gem in the history of Maharashtra he worked tirelessly for the cause of the lower caste subjects in his state. Primary education to all regardless of caste and creed was one of his most significant priorities. In our Institute we follow the noble practices established by our great leader and as we know Leader decide the destiny we are blessed to have a leadership and a rich cultural heritage which is totally focused on the noble cause of Quality Education. It functions beyond race, caste, creed, religion & political spirit. The Society Management is very pragmatic & progressive.

On Nov 10, 1917, Rajarshi Chhatrapati Shahu Maharaj announced in Delhi, his proposal to establish a Memorial of Chhatrapati Shivaji Maharaj in the City of Pune. In the proposal, he declared his intention of creating a central hall with Chhatrapati Shivaji Maharaj's life size statue and a hostel to accommodate a hundred Maratha students. The institute was to act as a rallying centre for Maratha activities all over India, giving a concrete shape and impetus to the diverse endeavours for the advancement of the Maratha community.

On Dec 27, 1917, Rajarshi Chhatrapati Shahu Maharaj, on a pressing demand from Khaserao Jadhav,

presided over the 11th session of the Maratha Educational Conference at Khamgoan. Rajarshi Chhatrapati Shahu Maharaj had convened a meeting of the committee of the All India Shivaji Memorial Society on 25 May 1922. Chhatrapati Rajaram Maharaj was elected President of the All India Shivaji Memorial Society and took on.

Core Values :

- Leadership and cultural heritage
- Honesty and Integrity
- Freedom of thought and Expression
- Excellence
- Accountability and Transparency
- Encouragement
- Social Responsibility

Honesty & Integrity :

The foreword of our society is "Satyala Maran Nahi" which means "Truth is eternal"

AISSMS
COLLEGE OF ENGINEERING
ज्ञानम् सकलजनहिताय
Accredited by NAAC with "A+" Grade

COLLEGE VISION AND MISSION

Vision :

Service to society through quality education.

Mission :

- Generation of national wealth through education and research
- Imparting quality technical education at the cost affordable to all strata of the society
- Enhancing the quality of life through sustainable development
- Carrying out high quality intellectual work
- Achieving the distinction of highest preferred engineering college in the eyes of the stake holders

Goal :

- To inculcate learning habits
- To create an environment to make the students creative and innovative
- To promote project based learning
- To strengthen industry – institute interaction
- To ensure continuous improvement in quality
- To develop entrepreneurship skills
- To nurture the spirit of team work
- To catalyze all – round development of students
- To develop technologies for sustainable development

DEPARTMENT VISION AND MISSION

Vision :

Society Growth and Welfare Through Competent Electronics and Communication Engineering Graduates.

Mission :

- To impart quality education in the field of E & TC engineering to solve societal and industrial problems with focus on trans-disciplinary approach
- To provide stimulating learning environment with modern tools & technologies.
- To produce dynamic graduates with ethics and moral values.
- To facilitate E & TC graduates with sight of innovation

Program outcomes (PO):

PO1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Educational Objectives (PEO):

PEO1: To build strong fundamental knowledge among graduates required to pursue their higher education and continue professional development.

PEO2: To enable graduates to identify, analyze and solve Electronics Engineering problems by applying basic principles and modern techniques.

PEO3: To enable graduates to innovate, design and develop hardware & software components and groom their ability to succeed in multidisciplinary and diverse field.

PEO4: To inculcate in graduates professional attitude, effective communicational skills, team work skills for becoming a responsible, cultured human being.

PROGRAMME SPECIFIC OUTCOME (PSO):

An E &TC engineering graduate will be able to:

PSO1: Analyze Design and test Analog and Digital circuits and systems for given application

PSO2: Implements technical blocks of hardware – software co-design for Embedded & Robotics automation application.

PSO3: Apply knowledge of E & TC system for social and environmental problems as a individual member or leader of diverse team in multidisciplinary settings

AISSMS
COLLEGE OF ENGINEERING
ज्ञानम् सकलजनहिताय
Accredited by NAAC with "A+" Grade

Course Objectives :

To introduce various network models, security threats and attacks and fundamentals of network security.

- To imbibe good foundation of network security in students for implementation of new network security algorithms.
- To understand different network models and the protocols used in each layer.
- To acquire detailed approach of encryption decryption for the data to transmit.
- To understand the role of network security as a tool for protection of different network entities.
- To be able to accurately apply security algorithms to real world security issues.
- To ensure windows and web browser security through implementation of various encryption standards.

Course Outcomes :

On completion of the course, learner will be able to -

CO1: Analyse attacks on computers and computer security.

CO2: Demonstrate knowledge of cryptography techniques.

CO3: Apply appropriate cryptographic technique by learning Symmetric and Asymmetric key Cryptography.

CO4: Evaluate different Message Authentication Algorithms and Hash Functions.

CO5: Compare various aspects of E-Mail Security.

CO6: Assimilate various aspects of Web Security.

AISSMS
COLLEGE OF ENGINEERING
ज्ञानम् सकलजनहिताय
Accredited by NAAC with "A+" Grade

CERTIFICATE

This is to certify that the **"Network Security Lab Manual"** submitted by **Name of Student** of **Third Year (Roll No)** performed and submitted during 2021 - 2022 academic year, in partial fulfilment of the requirements for the award of the degree of **BACHELOR OF ENGINEERING (THIRD YEAR ENGINEERING)** in **ELECTRONICS & TELECOMMUNICATION**

Signature

Dr. S.B.Dhonde

Savitribai Phule Pune University Third Year of E & Tc Engineering (2019 Course) 304198 (E): Network Security Lab (Elective – II)		
Teaching Scheme:	**Credit**	**Examination Scheme:**
Practical: 02 hrs. / week	**01**	**Practical: 25 Marks**

Companion Course, if any: Network Security

	Group A (Any Three)
1.	Design and implement for the insecurity of default passwords, printed passwords and password transmitted in plain text.
2.	Write a program for Encryption and Decryption.
3.	Write a program to perform encryption and decryption using the following algorithms: Ceaser Cipher, Substitution Cipher http://vlabs.iitb.ac.in/bootcamp/labs/dbms/exp13/
4.	Write a program to implement digital Signature http://cse29-iiith.vlabs.ac.in/
	Group B (Any Two)
6.	Isolating WLAN traffic using separate firewall for VPN connection
7.	Study of different wireless network components and features of any one of the Mobile Security Apps
8.	Implementation of Symmetric and Asymmetric cryptography
9.	Implementation of Steganography
	Group C (Any Three)
10.	Implementation of DES http://cse29-iiith.vlabs.ac.in/
11.	Implementation of AES http://cse29-iiith.vlabs.ac.in/
12.	Implementation of Windows security using firewall and other tools
13.	Steps to ensure Security of any one web browser (Mozilla Firefox/Google Chrome)
14.	Implementation of Hash functions http://cse29-iiith.vlabs.ac.in/

Virtual LAB Links:

Links of the Virtual Lab:

http://vlabs.iitb.ac.in/vlabs-dev/vlab_bootcamp/bootcamp/Byte_Karma/index.html

AISSMS
COLLEGE OF ENGINEERING

ज्ञानम् सकलजनहिताय

Accredited by NAAC with "A+" Grade

INDEX

Sr. No	Title (Eight Experiment to be perform)	Date of Performance	Date of Submission	Signature
1	Design and implement for the insecurity of default passwords, printed passwords and password transmitted in plain text.			
2	Write a program for Encryption and Decryption			
3	Write a program to perform encryption and decryption using the following algorithms: Ceaser Cipher, Substitution Cipher https://cryptii.com/ http://vlabs.iitb.ac.in/bootcamp/labs/dbms/exp13/			
4	Study of different wireless network components and features of any one of the Mobile Security Apps			
5	Implementation of Steganography			
6	Implementation of DES			
7	Implementation of Windows Security using Firewall and other tools.			
8	Steps to ensure Security of any one web browser (Mozilla Firefox/Google Chrome)			
9	Implementation of Symmetric and Asymmetric cryptography			
10	Case Study of AES			
11	Case Study of Hash functions			
12	Write a program to implement digital Signature			
Content beyond Syllabus				
1	One-Time Pad and Perfect Secrecy			
2	Defeating Malware – Rootkit Hunter			

Subject : Network Security (Elective - II)

Experiment No : 01

Name : **Roll No :**

Title : Design and implement for the insecurity of default passwords, printed passwords and password transmitted in plain text

Class : T.E **Date of Performance :** **Date of Submission :**

Aim: Design and implement for the insecurity of default passwords, printed passwords and password transmitted in plain text.

Theory:

Data Security & Need of Data Security:

Data security refers to the process of protecting data from unauthorized access and data corruption throughout its lifecycle. Data security includes data encryption, hashing, tokenization, and key management practices that protect data across all applications and platforms.

ta is a valuable asset that generates, acquires, saves, and exchanges for any company. Protecting it from internal or external corruption and illegal access protects a company from financial loss, reputational harm, consumer trust degradation, and brand erosion. The three components of Data Security that all companies should adhere to are confidentiality, integrity, and availability. The CIA triad is a security paradigm and framework for the protection of data. Here is what each fundamental piece implies in terms of preventing unwanted access and data exfiltration.

- Confidentiality: Ensures that only authorized users, with appropriate credentials, have access to data.
- Integrity: Ensures that all data is accurate, trustworthy, and not prone to unjustified changes.
- Availability: Ensures that data is accessible and available for ongoing business needs in a timely and secure manner.

If you're seeking Cyber Security courses to study, have a look at our Cyber Security course. Accelerate your enlistment and enrol right away!

Types Of Data Security Controls :

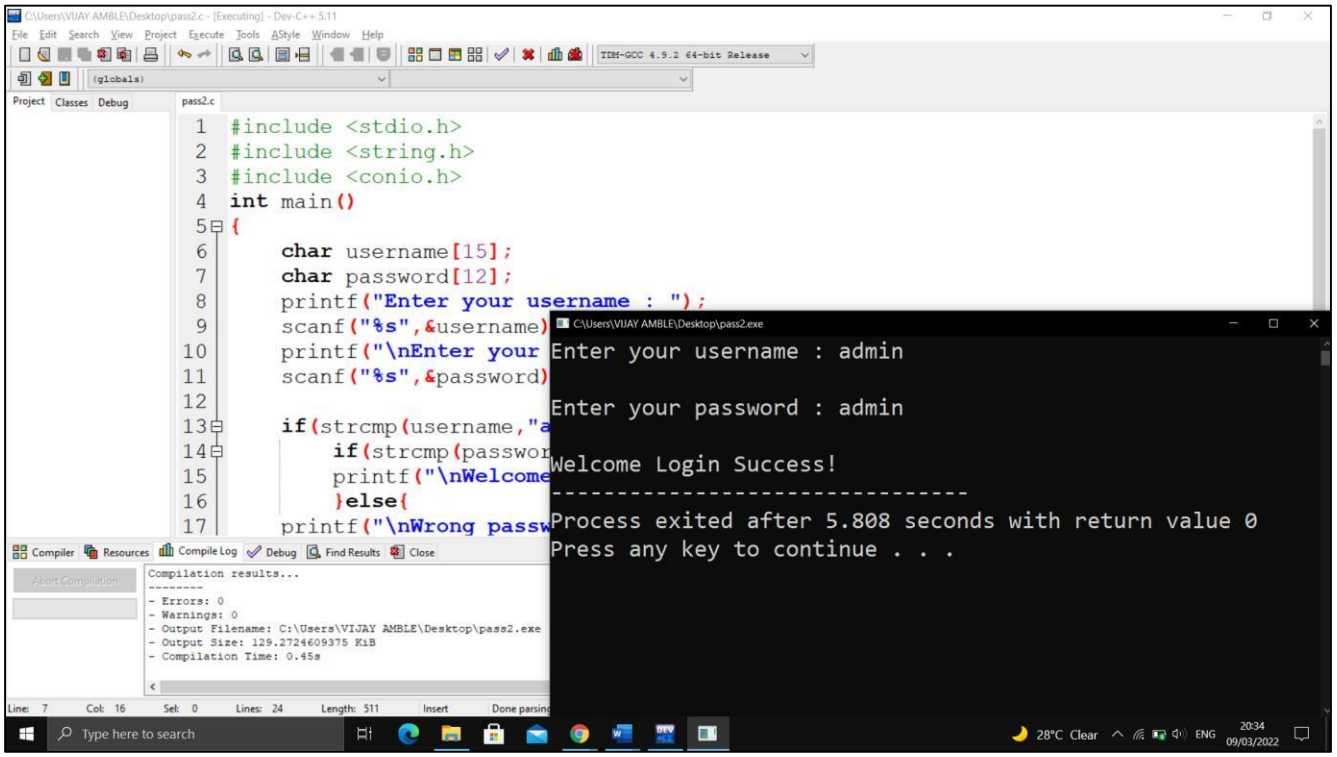
- Access Control
  Limiting both physical and digital access to central systems and data is an example of a strategy for securing data. It involves ensuring that all computers and gadgets are password-protected and that physical places are only accessible to authorized employees.

- Authentication
  Provide authentication measures, such as access restrictions and correct identification of people, before giving access to data. Passwords, PINs, security tokens, swipe cards, and biometrics are common examples.
- Backups and Disaster Recovery
  Good security means you have a strategy in place to safely access data in case of a system failure, disaster, data corruption, or breach. To restore, you will need a backup data copy kept on a distinct format such as a hard drive, local network, or Cloud.
- Data Erasure
  Appropriate discarding of data regularly is necessary. Data erasure is more secure than ordinary data wiping since data erasure uses software to wipe data completely on any storage device. Data erasure ensures that data cannot be recovered and, hence, will not fall into the wrong hands.
- Data Masking
  Data masking software obscures letters and numbers with proxy characters, concealing information. Even if a person obtains access to data illegally, it is successfully masked. Only when an authorized user acquires data, then does it revert back to its original state.
- Data Resilience
  With comprehensive security, you can withstand or recover from failures. Avoid power outages and mitigate natural catastrophes as these factors can breach data protection. Data privacy can be implemented by incorporating resilience into your hardware and software.
- Encryption
  With the help of encryption keys, a computer algorithm converts text characters into an unreadable format. The content can only be unlocked and accessed by authorized people who have the appropriate keys. To some extent, everything from files and databases to email conversations should be secured.

Write a Program for the insecurity of default passwords

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
int main()
{
   char username[15];
   char password[12];
   printf("Enter your username : ");
   scanf("%s",&username);
   printf("\nEnter your password : ");
   scanf("%s",&password);

   if(strcmp(username,"admin")==0){
      if(strcmp(password,"admin")==0){
      printf("\nWelcome Login Success!");
      }else{
   printf("\nWrong password");
}
   }else{
   printf("\nUser doesn't exist");
}
```

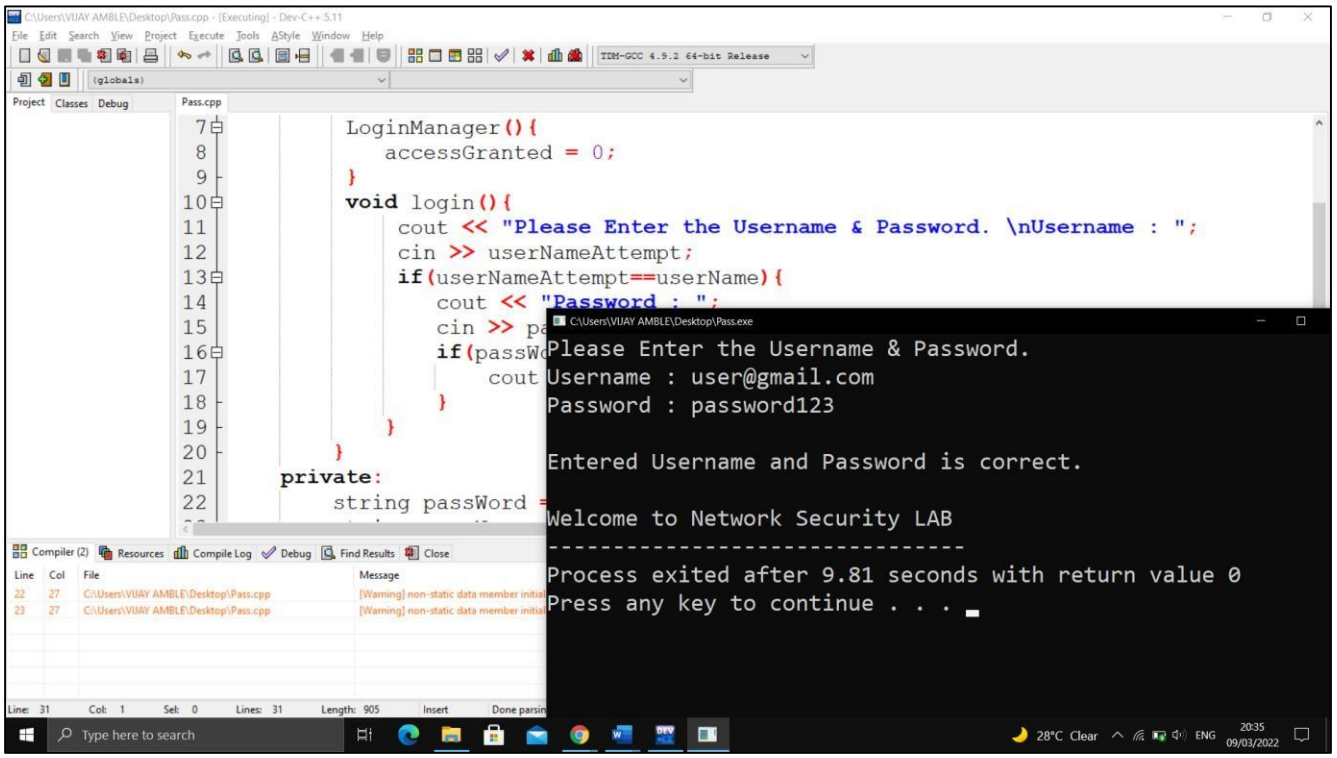
```
    return 0;
}
```



Write a Program for the password transmitted in plain text

```
#include  <iostream>
using namespace std;
class LoginManager{
    public:
        string userNameAttempt;
        string passWordAttempt;
        LoginManager(){
          accessGranted = 0;
        }
        void login(){
            cout << "Please Enter the Username & Password. \nUsername : ";
            cin >> userNameAttempt;
            if(userNameAttempt==userName){
              cout << "Password : ";
              cin >> passWordAttempt;
              if(passWordAttempt==passWord){
                  cout << "\nEntered Username and Password is correct. \n\nWelcome to Network Security
LAB";
              }
            }
        }
    private:
        string passWord = "password123";
        string userName = "user@gmail.com";
        bool accessGranted;
```

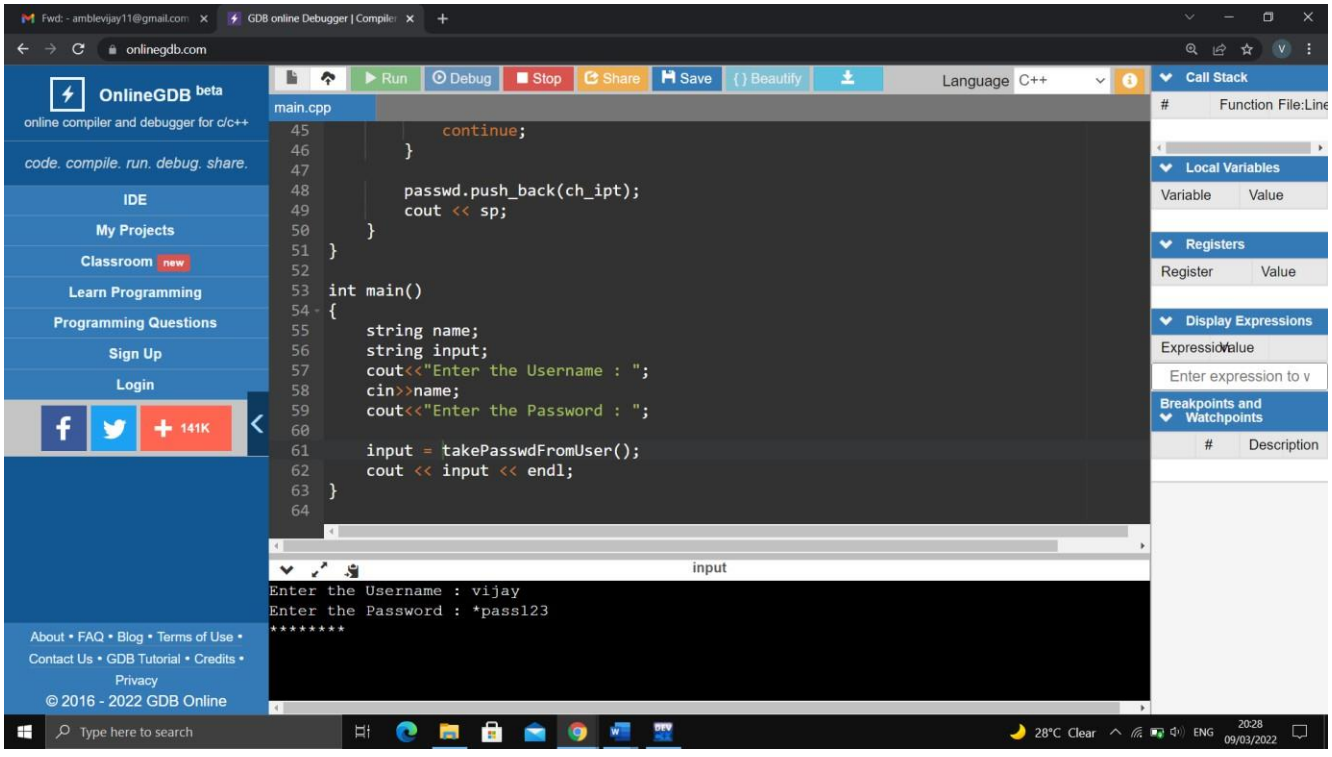
```
};
int main()
{
   LoginManager loginManagerObj;
   loginManagerObj.login();
}
```



Write a Program for the printed passwords

```
#include <conio.h>
#include <iostream>
using namespace std;
enum IN {
	IN_BACK = 8,
	IN_RET = 13
};

std::string takePasswdFromUser(
	char sp = '*')
{
	string passwd = "";
	char ch_ipt;
	while (true) {
		ch_ipt = getchar();
		if (ch_ipt == IN::IN_RET) {
			cout << endl;
			return passwd;
		}
		else if (ch_ipt == IN::IN_BACK
				&& passwd.length() != 0) {
```

```
                passwd.pop_back();
                cout << "\b \b";
                continue;
            }
            else if (ch_ipt == IN::IN_BACK
                    && passwd.length() == 0) {
                continue;
            }
            passwd.push_back(ch_ipt);
            cout << sp;
        }
}
int main()
{
        string name;
        string input;
        cout<<"Enter the Username : ";
        cin>>name;
        cout<<"Enter the Password : ";

        input = takePasswdFromUser();
        cout << input << endl;
}
```



Conclusion :

Subject : Network Security (Elective - II)

Experiment No : 02

Name : **Roll No :**

Title : Write a program for Encryption and Decryption

Class : T.E **Date of Performance :** **Date of Submission :**

Aim : Write a program for Encryption and Decryption.

Theory :

Network Data Encryption :

Encryption is the process of transforming data into an unintelligible form to prevent the unauthorized use of the data. To read an encrypted file, you must have access to a secret decryption key or password. Unencrypted data is called plain text; encrypted data is called cipher text. A cipher is an encryption-decryption algorithm.

Network Data Decryption :

Decryption is the process of transforming data that has been rendered unreadable through encryption back to its unencrypted form. In decryption, the system extracts and converts the garbled data and transforms it to texts and images that are easily understandable not only by the reader but also by the system. Decryption may be accomplished manually or automatically. It may also be performed with a set of keys or passwords.

Example

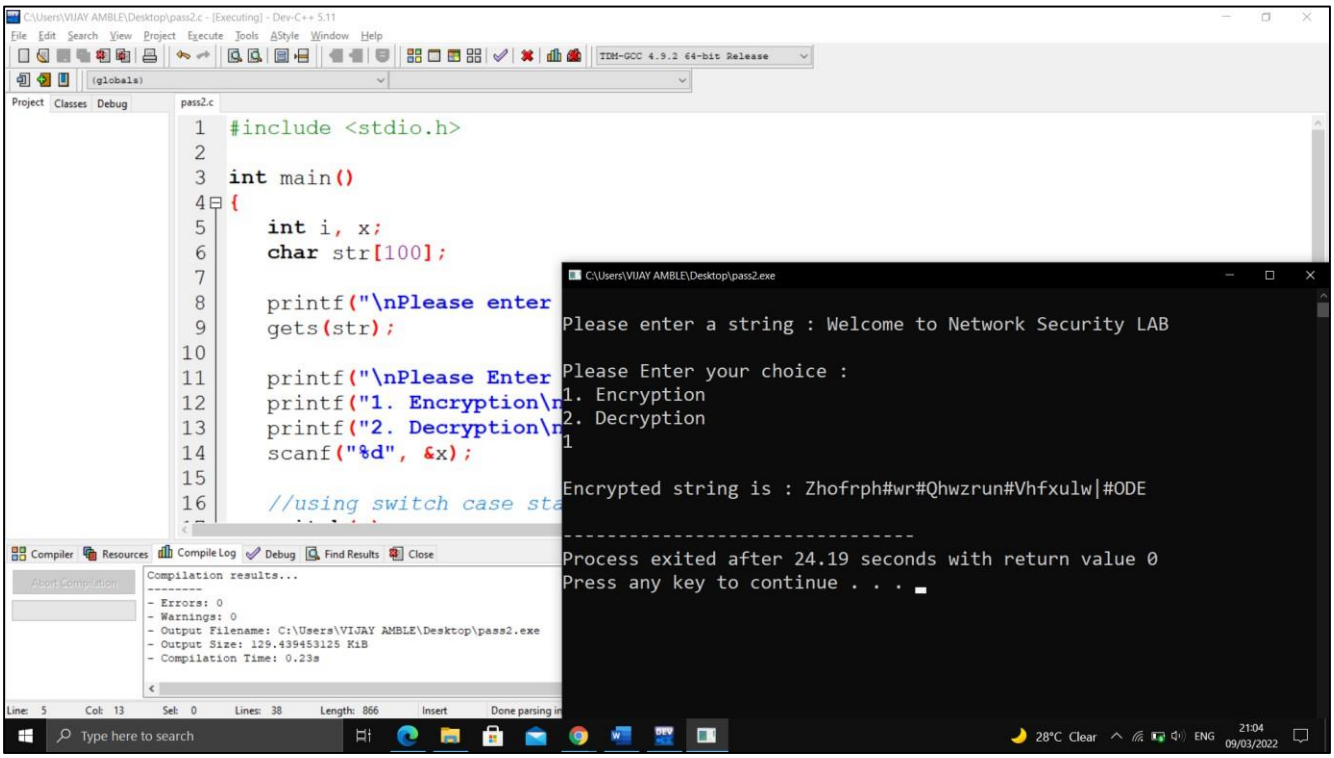
An employer wants to send some sensitive documents to an employee for processing but can't risk hackers getting hold of the documents and reading them. The employer can encrypt the documents and then send them to the employee who can then decrypt the documents using an employer-provided encryption key. They can then start working on the documents as they would normally.

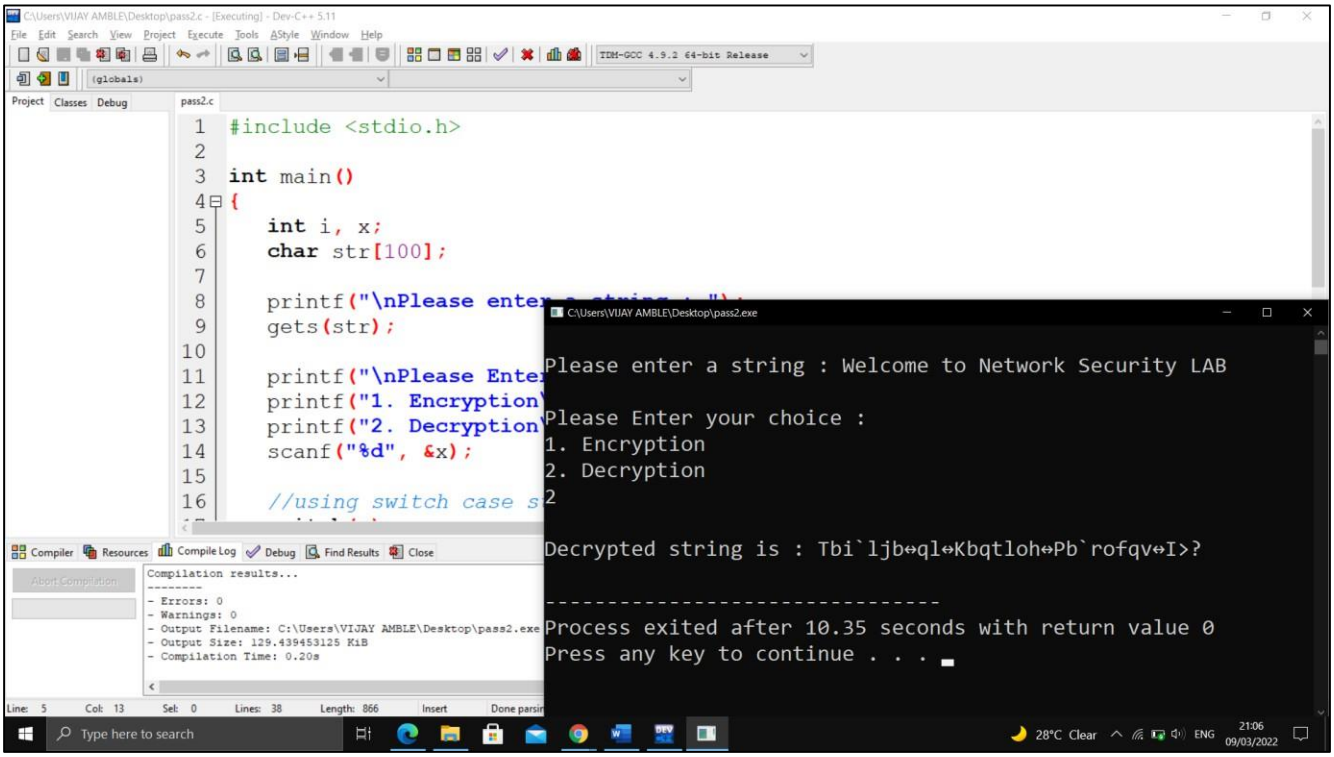
More specifically, let's say a document contains a top-secret word like "ALUMINUM." This word could be encrypted with an algorithm that replaces each letter with the letter that follows it in the alphabet.

So, the "A" in ALUMINUM would become "B," the "L" would become "M" and so on to give a cipher "BMVNJOVN." Anyone who wants to know what the original message was must know the algorithm which was used to encrypt the original information. This is a very simple example so one can easily tell what the original message was. In the real world, far more complicated algorithms are used to encrypt information. In any case, the sending party would let the receiving party know the encryption key. In this case, the key is to replace each letter with the letter that follows it in the alphabet. Using this key, the receiving party would decrypt the message, read the contents and move on.

Write a Program for Encryption & Decryption

```
#include <stdio.h>
int main()
{
  int i, x;
  char str[100];

  printf("\nPlease enter a string : ");
  gets(str);
  printf("\nPlease Enter your choice : \n");
  printf("1. Encryption\n");
  printf("2. Decryption\n");
  scanf("%d", &x);
  //using switch case statements
  switch(x)
  {
  case 1:
    for(i = 0; (i < 100 && str[i] != '\0'); i++)
      str[i] = str[i] + 3; //the key for encryption is 3 that is added to ASCII value
    printf("\nEncrypted string is : %s\n", str);
    break;
  case 2:
    for(i = 0; (i < 100 && str[i] != '\0'); i++)
      str[i] = str[i] - 3; //the key for encryption is 3 that is subtracted to ASCII value
    printf("\nDecrypted string is : %s\n", str);
    break;
  default:
    printf("\nError : Please Enter correct choice \n");
  }
  return 0;
}
```


#include <stdio.h>
int main()
{
int i, x;
char str[100];
printf("\nPlease enter
gets(str);
printf("\nPlease Enter
printf("1. Encryption\n
printf("2. Decryption\n
scanf("%d", &x);
//using switch case sta
C:\Users\VIJAY AMBLE\Desktop\pass2.exe
Please enter a string : Welcome to Network Security LAB
Please Enter your choice :
1. Encryption
2. Decryption
1
Encrypted string is : Zhofrph#wr#Qhwzrun#Vhfxulw|#ODE
Process exited after 24.19 seconds with return value 0
Press any key to continue . . .


#include <stdio.h>
int main()
{
int i, x;
char str[100];
gets(str);
scanf("%d", &x);
//using switch case s
C:\Users\VIJAY AMBLE\Desktop\pass2.exe
Please enter a string : Welcome to Network Security LAB
Please Enter your choice :
1. Encryption
2. Decryption
2
Decrypted string is : Tbi`ljb↔ql↔Kbqtloh↔Pb`rofqv↔I>?
Process exited after 10.35 seconds with return value 0
Press any key to continue . . .

Conclusion :

AISSMS
COLLEGE OF ENGINEERING
ज्ञानम् सकलजनहिताय
Accredited by NAAC with "A+" Grade

Subject : Network Security (Elective - II)

Experiment No : 03

Name : **Roll No :**

Title : Write a program to perform encryption and decryption using the following algorithms Caeser Cipher, Substitution Cipher

Class : T.E **Date of Performance :** **Date of Submission :**

Aim : Write a program to perform encryption and decryption using the following algorithms Ceaser Cipher, Substitution Cipher.

Theory :

Caesar Cipher is the substitution cipher techniques. It involves replacing each letter of the alphabet with the letter standing in any position. Caesar Cipher is used to encrypt data information to secure data information which is flowing into online. 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 position 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 E (Cipher Text) = (Plain Text P + Key K) mod 26

Decryption D (Plain Text) = (Cipher Text E - Key K) mod 26

Procedure :

To pass an encrypted message from one person to another, it is first necessary that both parties have the 'key' for the cipher, so that the sender may encrypt it and the receiver may decrypt it. For the caesar cipher, the key is the number of characters to shift the cipher alphabet.

Here is a quick example of the encryption and decryption steps involved with the caesar cipher. The text we will encrypt is 'defend the east wall of the castle', with a shift (key) of 1.

plaintext: defend the east wall of the castle

ciphertext: efgfoe uif fbtu xbmm pg uif dbtumf

It is easy to see how each character in the plaintext is shifted up the alphabet. Decryption is just as easy, by using an offset of -1.

plain: abcdefghijklmnopqrstuvwxyz

cipher: bcdefghijklmnopqrstuvwxyza

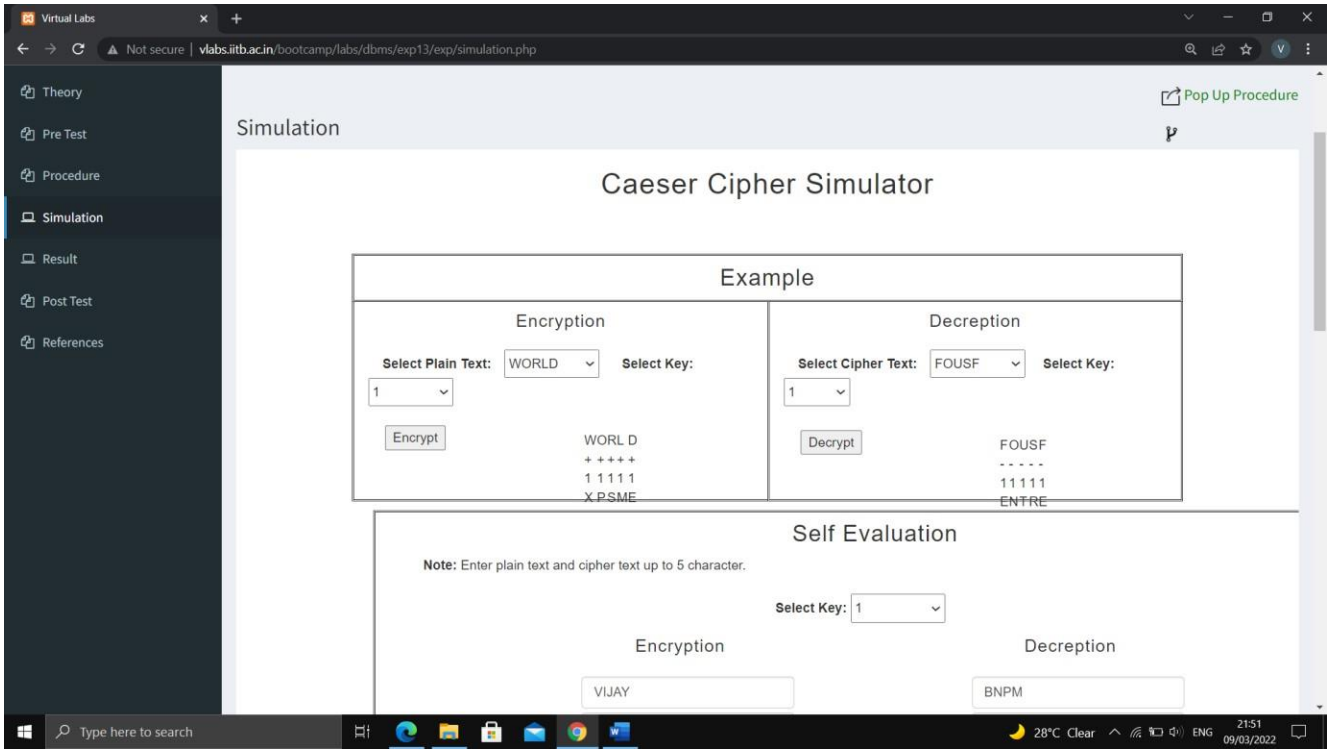
If a different key is used, the cipher alphabet will be shifted a different amount.

First we translate all of our characters to numbers, 'a'=0, 'b'=1, 'c'=2, ... , 'z'=25. We can now represent the caesar cipher encryption function, e(x), where x is the character we are encrypting, as:

Encryption E (Cipher Text) = (Plain Text P + Key K) mod 26

Where k is the key (the shift) applied to each letter. After applying this function the result is a number which must then be translated back into a letter. The decryption function is :

Decryption D (Plain Text) = (Cipher Text E - Key K) mod 26



Conclusion :

AISSMS
COLLEGE OF ENGINEERING
ज्ञानम् सकलजनहिताय
Accredited by NAAC with "A+" Grade

Subject : Network Security (Elective - II)

Experiment No : 04

Name : **Roll No :**

Title : Study of different wireless network components and features of any one of the Mobile Security Apps

Class : T.E **Date of Performance :** **Date of Submission :**

OBJECTIVE:

- Ability to understand the devices and components in a wireless network.
- Ability to Demonstrate a mobile security app and how it works for mobile security.
- Ability to interpret the network security issues in different types of network devices.

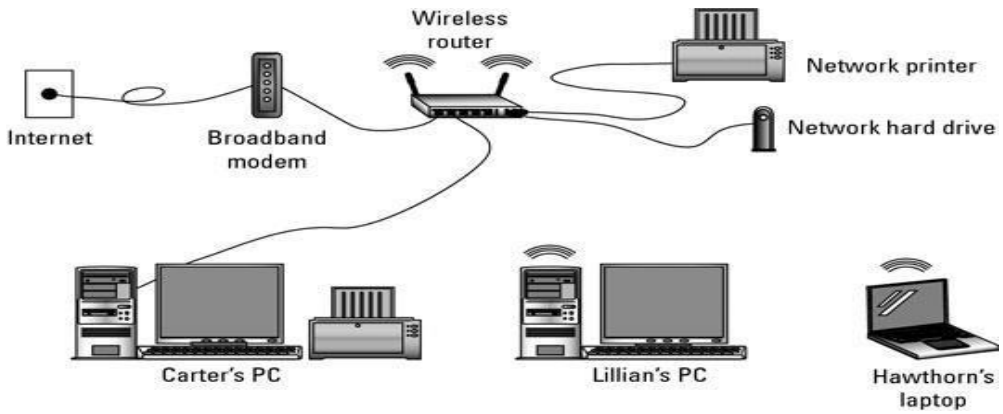
AIM: To perform encryption and decryption using the Ceaser Cipher algorithms

SOFTWARE USED: Virtual Labs - IIT Bombay
http://vlabs.iitb.ac.in/bootcamp/labs/dbms/exp13/

THEORY:

1.1 Introduction

As long as you have all the hardware, you can quickly set up any wireless network. Here is everything you need to know about the hardware you need to have in place before you use Windows to configure the wireless network. There are two types of wireless networks: infrastructure and ad hoc. The infrastructure network is most likely the type of wireless setup you have in your home or office. It's laid out similarly to a wired network, but without wires.



The basic wireless, peer-to-peer network consists of these components:

Wireless Network Adapters

Wireless network adapters (also known as wireless NICs or wireless network cards) are required for each device on a wireless network. All newer laptop computers incorporate wireless adapters as a built-in feature of the system. No wireless hardware other than adapters is required to build a small local network. However, to increase the performance of network connections, accommodate more computers, and increase the network's range, additional types of hardware can be deployed.

Wireless router

Wireless routers function comparably to traditional routers for wired Ethernet networks. One generally deploys wireless routers when building an all-wireless network from the ground up. Similar to routers, access points allow wireless networks to join an existing wired network. One typically deploys access points when growing a network that already has routers installed. In home networking, a single access point (or router) possesses sufficient range to span most residential buildings. Businesses in office buildings often must deploy multiple access points and/or routers.

Wireless Antennas

Access points and routers often utilize a Wi-Fi wireless antenna that significantly increase the communication range of the wireless radio signal. These antennas are optional and removable on most equipment. It's also possible to mount aftermarket add-on antennas on wireless clients to increase the range of wireless adapters.

Wireless Repeaters

A wireless repeater connects to a router or access point. Often called signal boosters or range expanders, repeaters serve as a two-way relay station for wireless radio signals, helping clients otherwise unable to receive a network's wireless signal to join.

Wire-based connections:

Almost every wireless router has one or more standard, wire-based Ethernet port. One port is used to connect the router to a broadband modem. Other Ethernet ports might be also available, allowing you to connect standard wire-based networking to the wireless hub.

Wireless NIC:

Your computer needs a wireless Network Interface Card, or NIC, to talk with the wireless router. A laptop comes standard with a wireless NIC, but for a desktop PC you have to get a wireless NIC as an option. It’s installed internally as an expansion card, or you can use one of the various plug-in USB wireless NICs. other type of network called the ad hoc type of wireless network is basically a group of wireless computers connected with each other. An ad-hoc network has no central hub or router. Instead, all its computers can directly access the other computers’ files and shared resources. They may or may not have Internet access, but that’s not the point of the ad hoc network.

 One of the beauties of a wireless network is that you can mix in wired components as needed. If you need more Ethernet ports, for example, simply add a switch to the wireless router.

 Despite the wireless nature of wireless networking, you still need an Ethernet cable (a wire) to connect a wireless router to a broadband modem.

 Another advantage of a wireless network is that it’s portable. It’s easier to pull up stakes with a wireless network than to pack up all the bits and pieces of a wired network. If you live in an apartment, or just move around a lot, wirelesses setup a good option.

The term access point is often abbreviated AP. Don't be puzzled when you see the words wireless AP — it simply refers to the access point, not to the Associated Press.

- A wireless network is often called a WLAN, for wireless local-area network.
- A wireless network is also referred to by the term Wi-Fi. It stands for wireless fidelity.
- Ad hoc networks are often used by computer gamers to gather in a single location to play games with each other.

1.1 Mobile App Security

Mobile app security is the extent of protection that mobile device applications (apps) have from malware and the activities of crackers and other criminals. The term can also refer to various technologies and production practices that minimize the risk of exploits to mobile devices through their apps. A mobile device has numerous components built, and used by multiple
players, each of whom plays a crucial role the security of a device. Each player should incorporate security measures into mobile devices as they are designed and built and into mobile apps as they are conceived and written, but these tasks are not always adequately carried out. Common vulnerabilities for mobile devices include architectural flaws, device loss or theft, platform weakness, isolation and permission problems.

When evaluating mobile devices and apps for security, developers should ask themselves the following questions.

- How do users obtain a particular app?
- Should a firm create its own app store?
- How is an app vetted before it is offered for sale?
- How is an app protected against malware?
- How can users tell the difference between a legitimate app and a fake?
- How easily can automatic update features get hijacked?
- What measures exist to control the risk of device jail breaking?
- What kind of permissions should a particular app ask for?
- Can any other apps keep track of when, where, and how a certain app is used?

Let us now discuss the features of a popular mobile security app called CM Security

1.1.1 CM Security

CM security (Clean Master) is an all-singing, Mobile that brings you a whole host of anti-virus and security features for free Features of CM security Feature-wise it offer everything - anti-virus, browsing protection, battery saving, privacy protection of apps.Clean Master is one of the best cleaner apps available in the market. The app comes with a variety of features, and if users want more, they can easily upgrade to the pro version. Apart from providing access in various languages, the tool improves the speed of your mobile and cleans unwanted files quickly.
The app is designed to perform numerous tasks like scanning your mobile, categorizing files into different groups, and wiping clean any files that are causing your system to slow.
Moreover, the app also scans for viruses while **getting rid of cache files and cookies**. The clean and intuitive interface is user-friendly, which ensures that even individuals that aren't tech-savvy can easily use it to get rid of junk!

The Clean Master app comes with a 'Clean Now' button, which **removes all unwanted files at once**. Included in this are documents in the recycle bin, system files, temp files, logs, OS files, web caches, registry files as well as software junk files.

Merit :
Scans and removes unwanted junk files Prominent features are free to use Comes with a Privacy Clean option The interface is user-friendly
Demerit :
Professional version has a lot more features

OUTPUT: (attach screenshot mobile app along with output scan test)

CONCLUSION:

Subject : Network Security (Elective - II)

Experiment No : 05

Name : **Roll No :**

Title : Implementation of Steganography

Class : T.E **Date of Performance :** **Date of Submission :**

Aim : Implementation of Steganography.

Theory :

What is Steganography?

Steganography is the technique of hiding secret data within an ordinary, non-secret, file or message in order to avoid detection; the secret data is then extracted at its destination. The use of steganography can be combined with encryption as an extra step for hiding or protecting data. The word steganography is derived from the Greek words steganos (meaning hidden or covered) and the Greek root graph (meaning to write).

Steganography can be used to conceal almost any type of digital content, including text, image, video or audio content; the data to be hidden can be hidden inside almost any other type of digital content. The content to be concealed through steganography -- called hidden text -- is often encrypted before being incorporated into the innocuous-seeming cover text file or data stream. If not encrypted, the hidden text is commonly processed in some way in order to increase the difficulty of detecting the secret content.

Examples of Steganography :

Steganography is practiced by those wishing to convey a secret message or code. While there are many legitimate uses for steganography, malware developers have also been found to use steganography to obscure the transmission of malicious code.

Forms of steganography have been used for centuries and include almost any technique for hiding a secret message in an otherwise harmless container. For example, using invisible ink to hide secret messages in otherwise inoffensive messages; hiding documents recorded on microdot -- which can be as small as 1 millimeter in diameter -- on or inside legitimate-seeming correspondence; and even by using multiplayer gaming environments to share information.

Advantages of steganography over cryptography :

Steganography is distinct from cryptography, but using both together can help improve the security of the protected information and prevent detection of the secret communication. If steganographically-hidden data is also encrypted, the data may still be safe from detection -- though the channel will no longer be safe from detection. There are advantages to using steganography combined with encryption over encryption-only communication.

The primary advantage of using steganography to hide data over encryption is that it helps obscure the fact that there is sensitive data hidden in the file or other content carrying the hidden text. Whereas an encrypted file, message or network packet payload is clearly marked and identifiable as such, using steganographic techniques helps to obscure the presence of the secure channel.

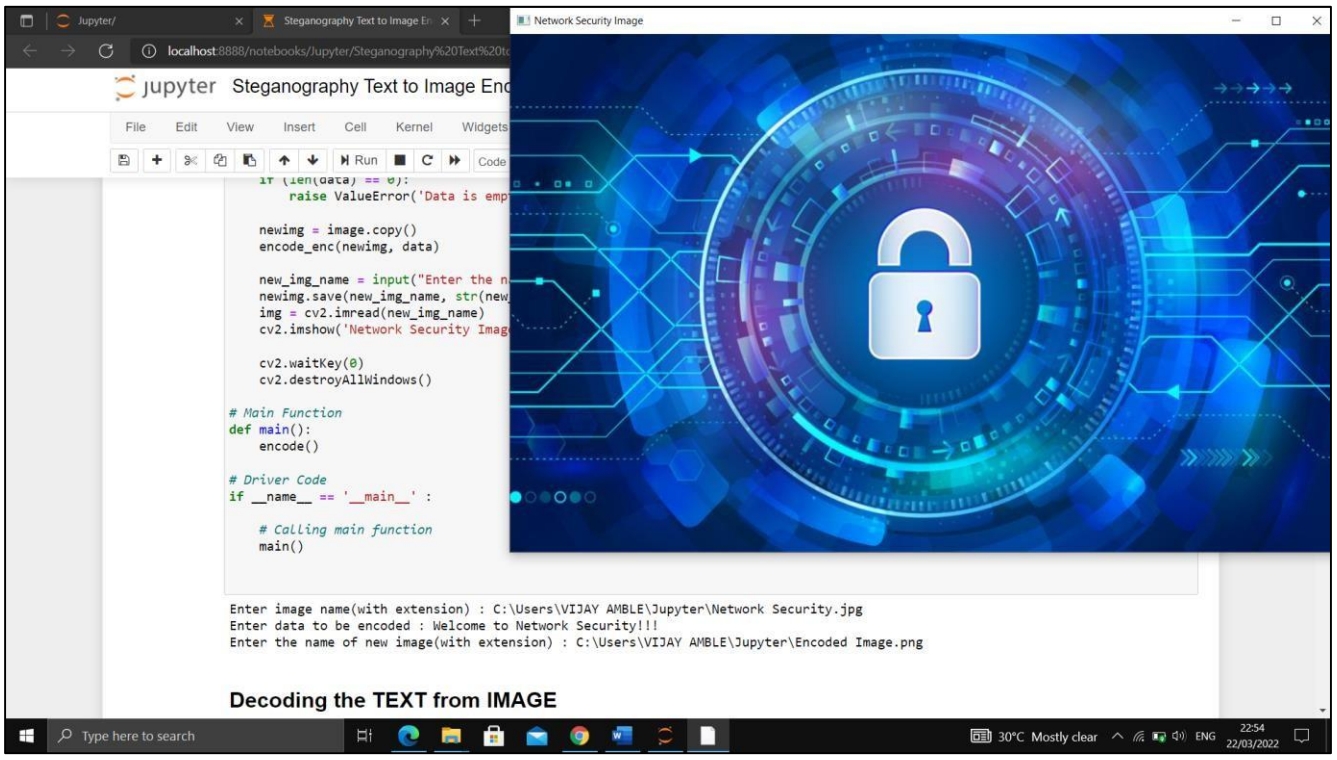
Write a program to for implementation of Steganography
(Text to Image Encryption & Decrpytion)

Encoding the TEXT in IMAGE

```
# Python program implementing Image Steganography

# PIL module is used to extract
# pixels of image and modify it
from PIL import Image
import cv2
# Convert encoding data into 8-bit binary
# form using ASCII value of characters

from IPython import display  #to load the image

def genData(data):

        # list of binary codes
        # of given data
        newd = []

        for i in data:
            newd.append(format(ord(i), '08b'))
        return newd

# Pixels are modified according to the
# 8-bit binary data and finally returned
def modPix(pix, data):

    datalist = genData(data)
    lendata = len(datalist)
    imdata = iter(pix)

    for i in range(lendata):

        # Extracting 3 pixels at a time
        pix = [value for value in imdata.__next__()[:3] +
                              imdata._next_()[:3] +
                              imdata._next_()[:3]]

        # Pixel value should be made
        # odd for 1 and even for 0
        for j in range(0, 8):
            if (datalist[i][j] == '0' and pix[j]% 2 != 0):
                pix[j] -= 1
```

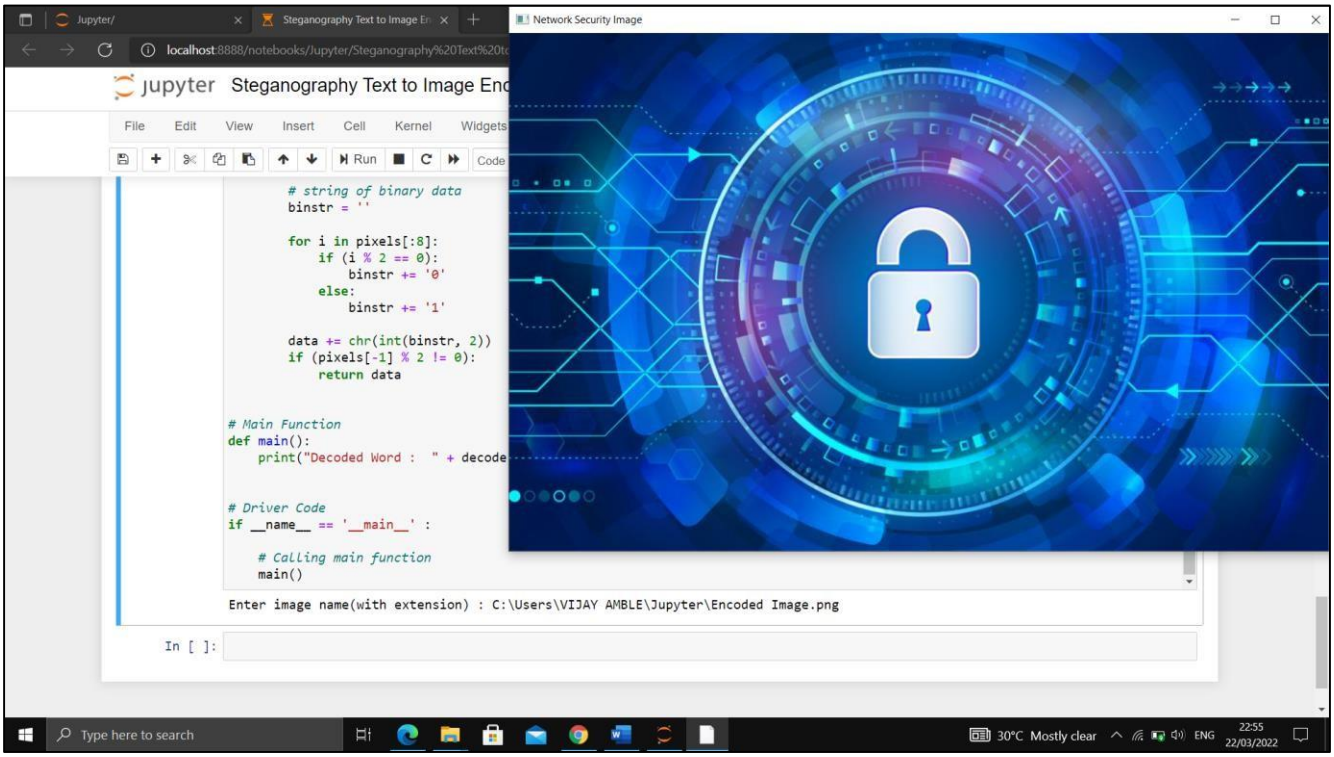
```
            elif (datalist[i][j] == '1' and pix[j] % 2 == 0):
                if(pix[j] != 0):
                    pix[j] -= 1
                else:
                    pix[j] += 1
                # pix[j] -= 1

        # Eighth pixel of every set tells
        # whether to stop ot read further.
        # 0 means keep reading; 1 means thec
        # message is over.
        if (i == lendata - 1):
            if (pix[-1] % 2 == 0):
                if(pix[-1] != 0):
                    pix[-1] -= 1
                else:
                    pix[-1] += 1

        else:
            if (pix[-1] % 2 != 0):
                pix[-1] -= 1

        pix = tuple(pix)
        yield pix[0:3]
        yield pix[3:6]
        yield pix[6:9]

def encode_enc(newimg, data):
    w = newimg.size[0]
    (x, y) = (0, 0)

    for pixel in modPix(newimg.getdata(), data):

        # Putting modified pixels in the new image
        newimg.putpixel((x, y), pixel)
        if (x == w - 1):
            x = 0
            y += 1
        else:
            x += 1

# Encode data into image
def encode():
    img = input("Enter image name(with extension) : ")
    image = Image.open(img, 'r')

    data = input("Enter data to be encoded : ")
    if (len(data) == 0):
```

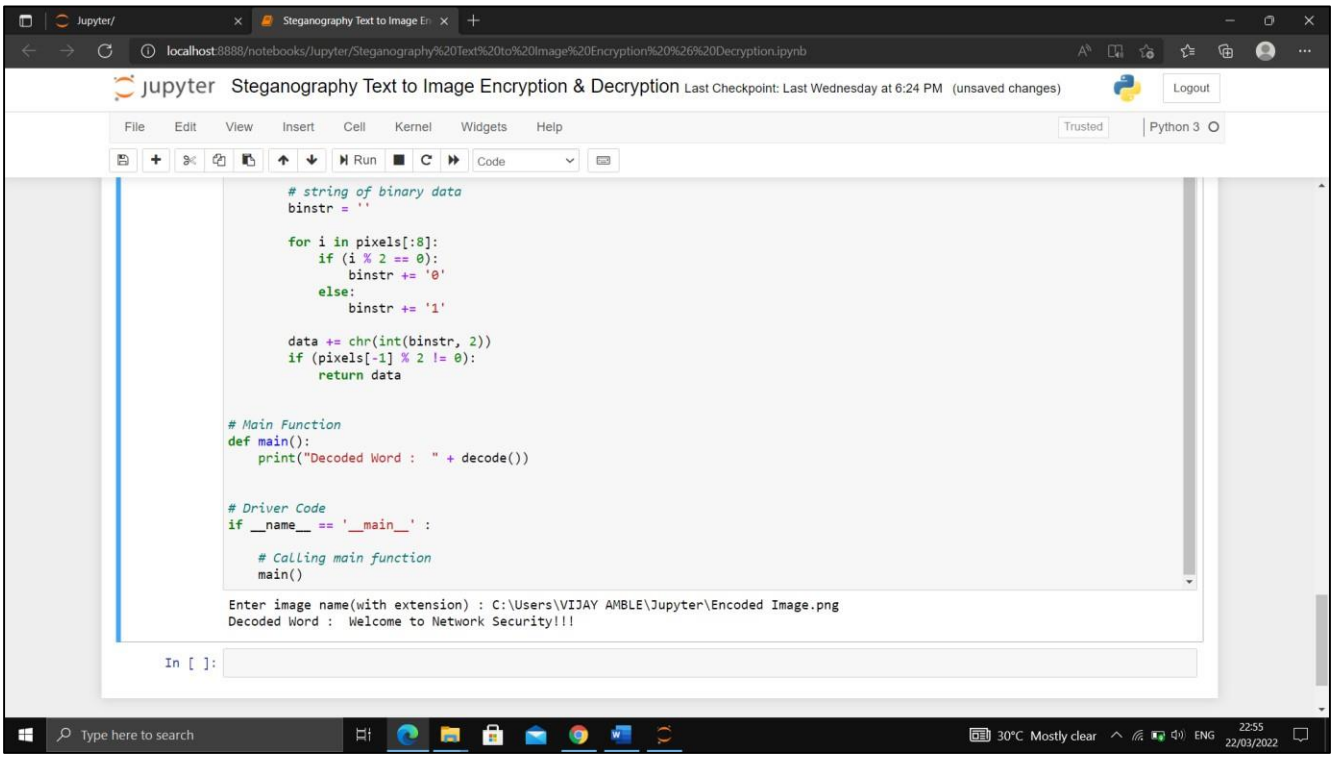
```
        raise ValueError('Data is empty')

    newimg = image.copy()
    encode_enc(newimg, data)

    new_img_name = input("Enter the name of new image(with extension) : ")
    newimg.save(new_img_name, str(new_img_name.split(".")[1].upper()))
    img = cv2.imread(new_img_name)
    cv2.imshow('Network Security Image',img)

    cv2.waitKey(0)
    cv2.destroyAllWindows()

# Main Function
def main():
    encode()

# Driver Code
if __name___== '__main__' :

    # Calling main function
    main()
```



Decoding the TEXT from IMAGE

```
# Decode the data in the image
def decode():
    img = input("Enter image name(with extension) : ")
    image = Image.open(img, 'r')

    data = ''
    imgdata = iter(image.getdata())
    img1 = cv2.imread(img)
    cv2.imshow('Network Security Image', img1)

    cv2.waitKey(0)
    cv2.destroyAllWindows()

    while (True):
        pixels = [value for value in imgdata.__next__()[:3] +
                                imgdata.__next__()[:3] +
                                imgdata.__next__()[:3]]

        # string of binary data
        binstr = ''

        for i in pixels[:8]:
            if (i % 2 == 0):
                binstr += '0'
            else:
                binstr += '1'

        data += chr(int(binstr, 2))
        if (pixels[-1] % 2 != 0):
            return data


# Main Function
def main():
    print("Decoded Word :  " + decode())


# Driver Code
if __name__== '__main__' :

    # Calling main function
    main()
```


Network Security Image
jupyter Steganography Text to Image Enc
File Edit View Insert Cell Kernel Widgets
Run Code
# string of binary data
binstr = ''
for i in pixels[:8]:
if (i % 2 == 0):
binstr += '0'
else:
binstr += '1'
data += chr(int(binstr, 2))
if (pixels[-1] % 2 != 0):
return data
# Main Function
def main():
print("Decoded Word : " + decode
Driver Code
if __name__ == '__main__' :
# Calling main function
main()
Enter image name(with extension) : C:\Users\VIJAY AMBLE\Jupyter\Encoded Image.png
In [ ]:
Type here to search
30°C Mostly clear
ENG
22:55
22/03/2022


jupyter Steganography Text to Image Encryption & Decryption Last Checkpoint: Last Wednesday at 6:24 PM (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run Code
# string of binary data
binstr = ''
for i in pixels[:8]:
if (i % 2 == 0):
binstr += '0'
else:
binstr += '1'
data += chr(int(binstr, 2))
if (pixels[-1] % 2 != 0):
return data
# Main Function
def main():
print("Decoded Word : " + decode())
# Driver Code
if __name__ == '__main__' :
# Calling main function
main()
Enter image name(with extension) : C:\Users\VIJAY AMBLE\Jupyter\Encoded Image.png
Decoded Word : Welcome to Network Security!!!
In [ ]:
Type here to search
30°C Mostly clear
ENG
22:55
22/03/2022

Conclusion :

AISSMS
COLLEGE OF ENGINEERING
ज्ञानम् सकलजनहिताय
Accredited by NAAC with "A+" Grade

Subject : Network Security (Elective - II)

Experiment No : 06

Name : **Roll No :**

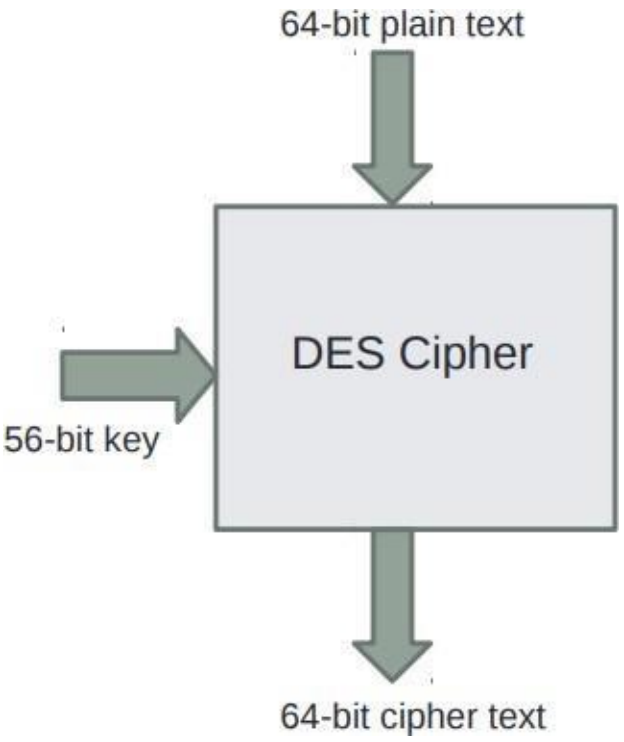
Title : Implementation of DES

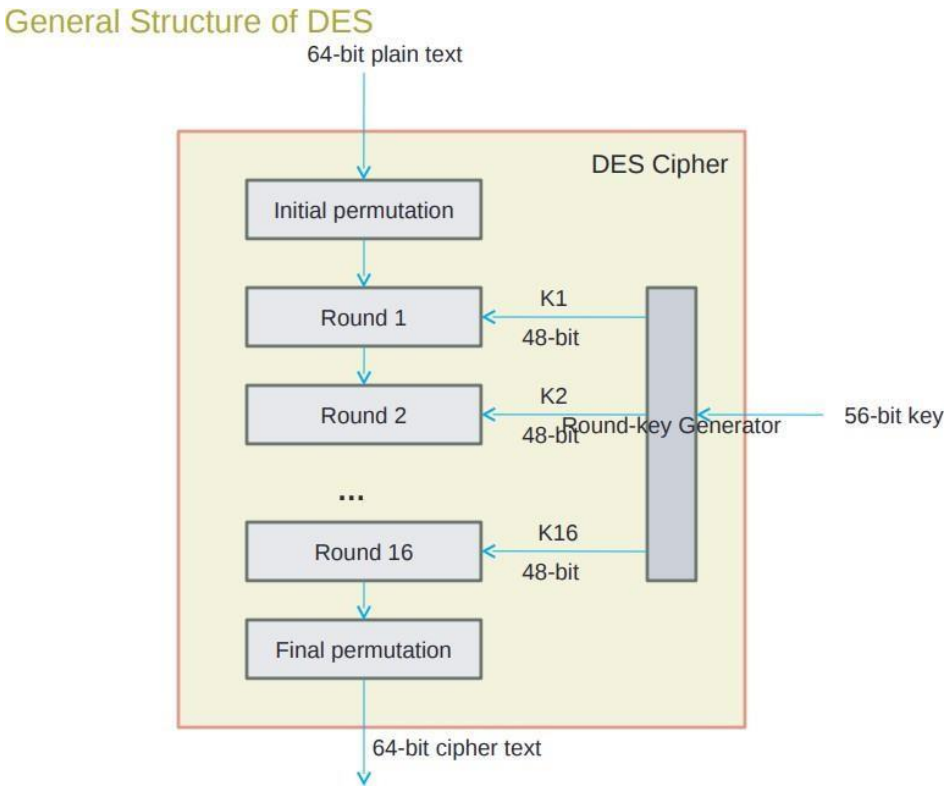
Class : T.E **Date of Performance :** **Date of Submission :**

Aim : Implementation of DES

Theory :

DES is a Block cipher, which takes 64-bit plain text and creates a 64-bit cipher text





Initial and Final permutations

Initial Permutation Table

The 58th bit of the input 64-bit plain text becomes the 1st bit, the 50th bit becomes the 2nd bit and so on according to the initial permutation table

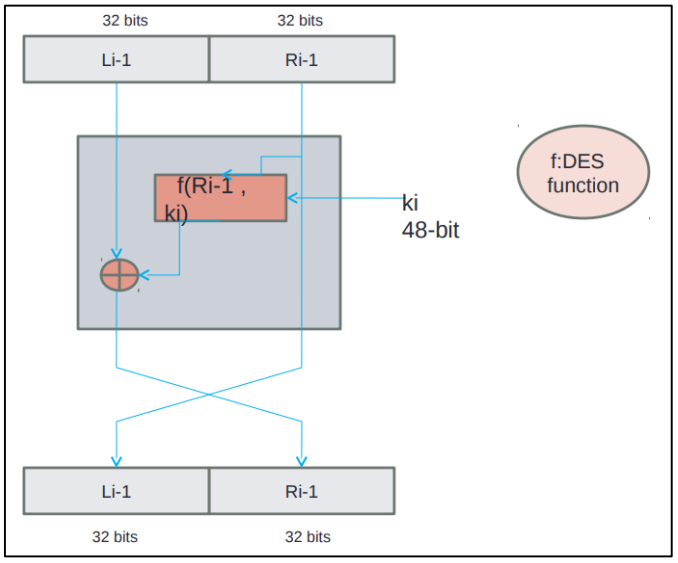
58	50	42	34	26	18	10	02
60	52	44	36	28	20	12	04
62	54	46	38	30	22	14	06
64	56	48	40	32	24	16	08
57	49	41	33	25	17	09	01
59	51	43	35	27	19	11	03
61	53	45	37	29	21	13	05
63	55	47	39	31	23	15	07

Final Permutation Table

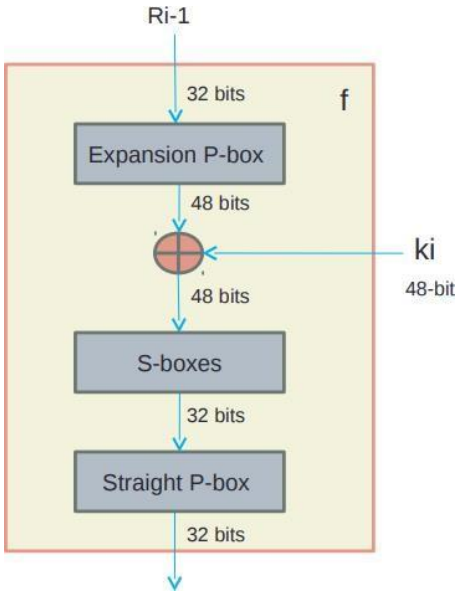
The 40th bit of the 64-bit output of the Round 16 becomes the1st bit, the 8th bit becomes the 2nd bit and so on according to The final permutation table

40	08	48	16	56	24	64	32
39	07	47	15	55	23	63	31
38	06	46	14	54	22	62	30
37	05	45	13	53	21	61	29
36	04	44	12	52	20	60	28
35	03	43	11	51	19	59	27
34	02	42	10	50	18	58	26
33	01	41	09	49	17	57	25

One Round in DES (Feistel structure)

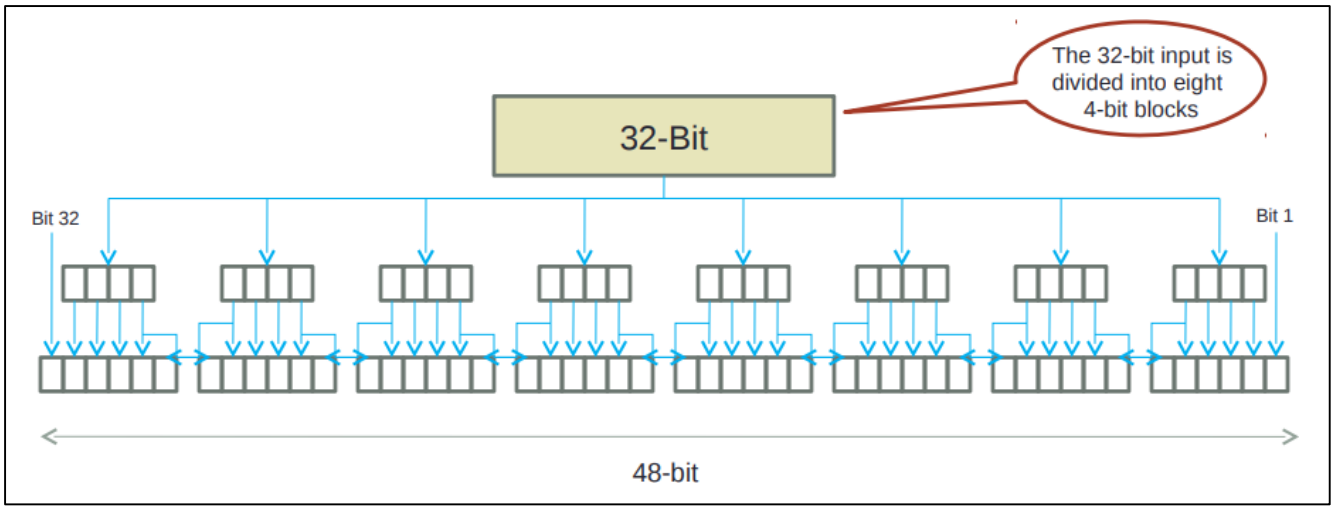


DES Function



DES Function : Expansion permutation

The input 32-bits are expanded to 48 bits in the Expansion P-Box module in the following way



The resulting 48-bit output is permuted using the Expansion P-Box

DES Function : Expansion Permutation and Straight permutation

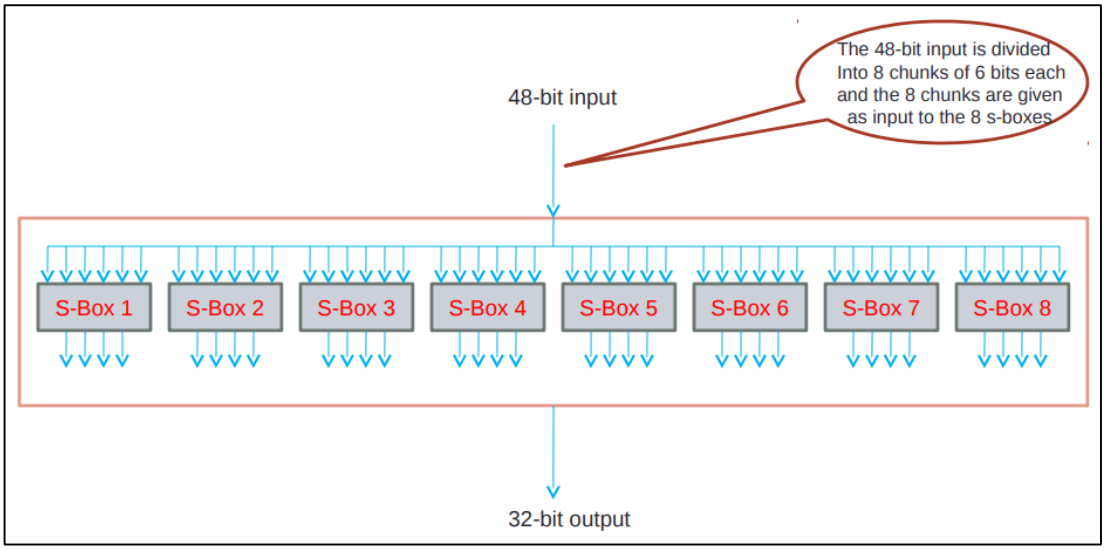
Expansion P-box

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	30	31	32	01

Straight P-box

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

DES Function : Substitution Boxes

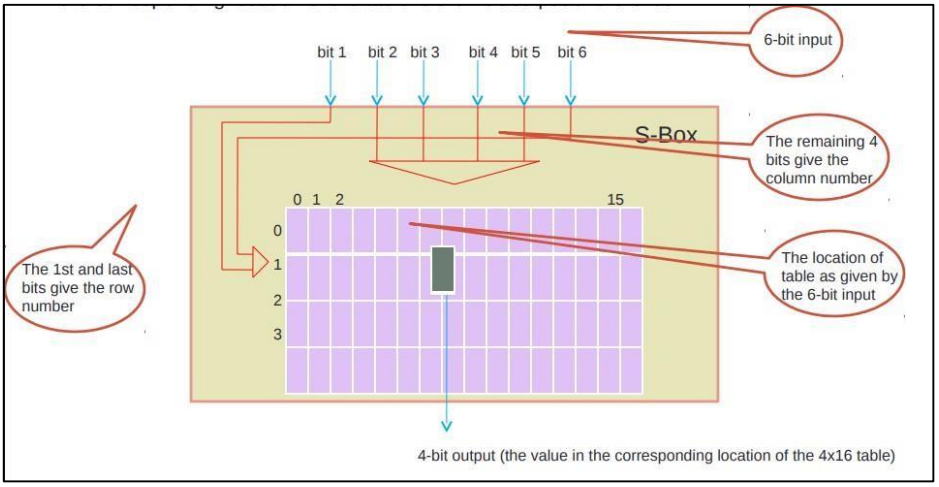


The output of each S-box is 4-bit. When these are combined the result is a 32-bit output

DES Function : Substitution Boxes

Each S-box uses a corresponding 4 row by 16 column table

Given a 6-bit input, the 1st and the 6th bits are used to address one of the rows and the remaining 4 bits are used to address one of the 16 columns. Finally, the value found in the corresponding location of the table is the 4bit output of the S-box



DES Function : Substitution Boxes

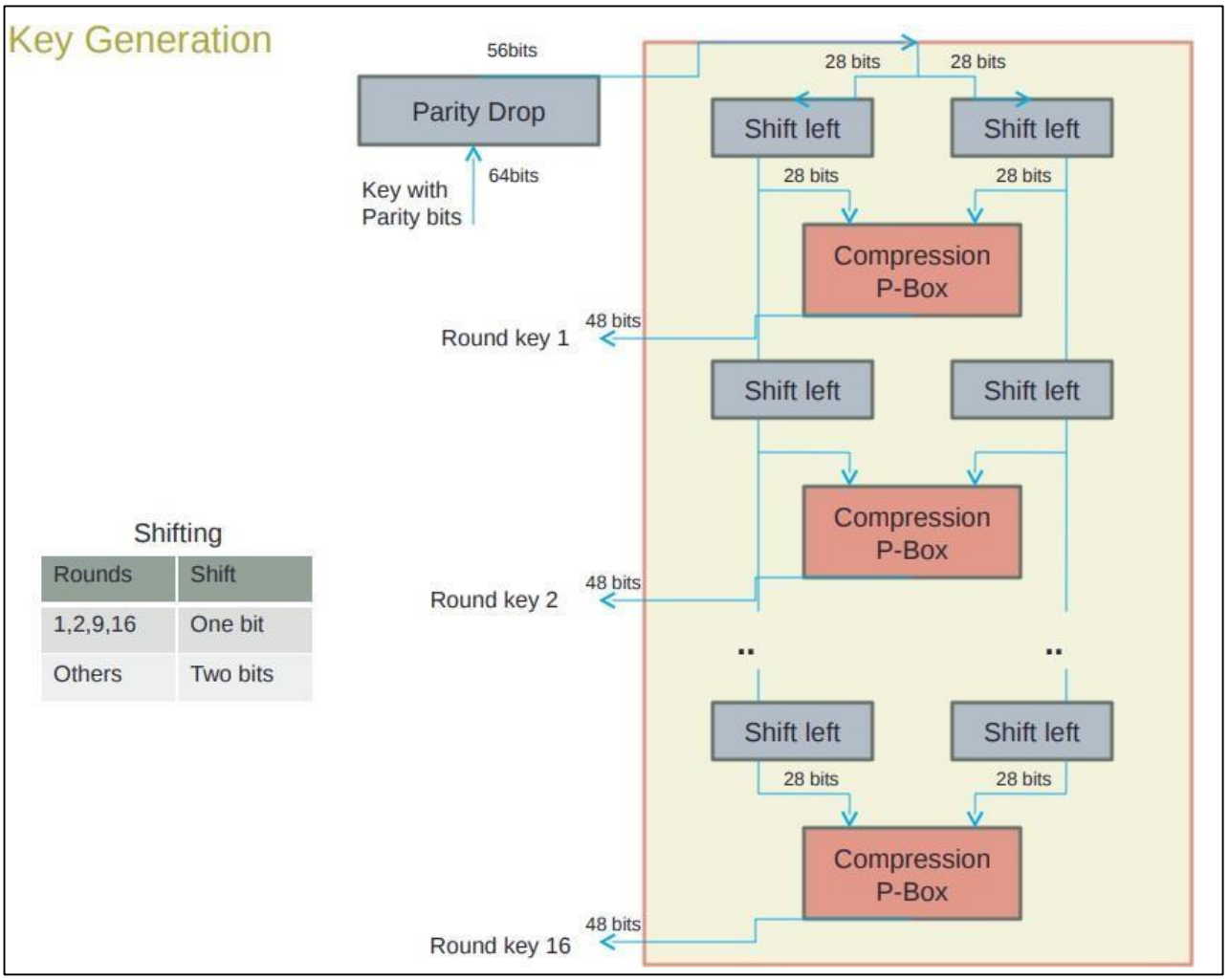
An Example

Consider the 6-bit input to s-box 1 is 100011 The 1st and last bits put together is 11 which is ‘3’ in decimal. So we select the 3rd row The middle bits are 00001 which is ‘1’ in decimal. So we select the 1st column The corresponding table for S-box 1 is shown below

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	10	03	06	12	11	09	05	03	08
3rd	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

The value in the 3rd in binary)

Key Generation



Parity Drop and Compression Permutation

The parity drop module drops the parity bits (bits 8,16,24,..,64) from the 64-bit key and permutes the rest of the 56 bits according to the parity drop table The Compression permutation module changes the 56 bits to 48 bits using the key compression table, which are used as the key for a round

Parity drop table

57	49	41	33	25	17	09	01
58	50	42	34	26	18	10	02
59	51	43	35	27	19	11	03
60	52	44	36	63	55	47	39
31	23	15	07	62	54	46	38
30	22	14	06	61	53	45	37
29	21	13	05	28	20	12	04

Key compression table

14	17	11	24	01	05	03	28
15	06	21	10	23	19	12	04
26	08	16	07	27	20	13	02
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

Procedure :

From DES to 3-DES

Step 1 : Generate Plaintext **m, keyA** and **keyB** by clicking on rexpective buttons **PART I** of the simulation page.

Step 2 : Enter generated Plaintext **m** from **PART I** to **PART II** in "Your text to be encrypted/decrypted:" block.

Step 3 : Enter generated **keyA** from **PART I** to **PART II** "Key to be used:" block and click on DES encrpt button to output ciphertext **c1.**This is First Encryption.

Step 4 : Enter generated ciphertext **c1** from **PART II** "Output:" Block to **PART II** in "Your text to be encrypted/decrypted:" block.

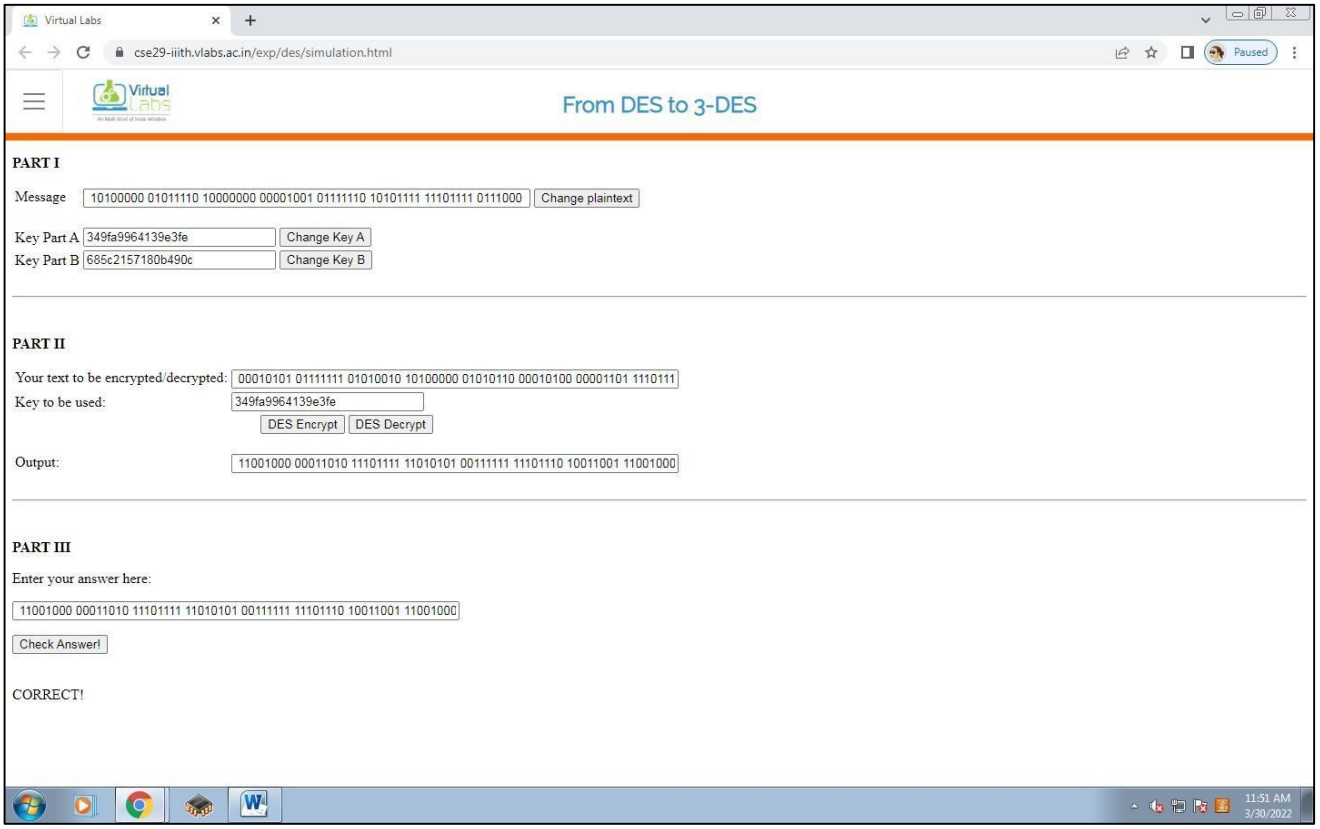
Step 5 : Enter generated **keyB** from **PART I** to **PART II** in "Key to be used:" block and click on DES decrypt button to output ciphertect **c2.**This is Second Encryption.

Step 6 : Enter generated ciphertext c2** from **PART II** "Output:" block to **PART II** in "Your text to be encrypted/decrypted:" block.

Step 7 : Enter generated **keyA** from **PART I** to **PART II** "Key to be used:" block and click on DES encrpt button to output ciphertext **c3**.This is Third Encryption. As Encryption is done thrice.This Scheme is called triple DES

Step 8 : Enter generated ciphertext **c3** from **PART II** "Output:" Block to **PART III** "Enter your answer here:" block inorder to verify your Triple DES.

Simulation :



Conclusion :

Subject : Network Security (Elective - II)

Experiment No : 07

Name : **Roll No :**

Title : Implementation of Windows Security using Firewall and other tools.

Class : T.E **Date of Performance :** **Date of Submission :**

OBJECTIVE:

- To demonstrate setup a firewall on Operating System.
- To interpret the Windows Firewall with Advanced Security.

AIM: Study of the features of firewall in providing network security and to set Firewall Security in windows.

THEORY:

Firewall in Windows 10

Windows 10 comes with two firewalls that work together. One is the Windows Firewall, and the other is Windows Firewall with Advanced Security (WFAS). The main difference between them isthe complexity of the rules configuration. Windows Firewall uses simple rules that directly relate toa program or a service. The rules in WFAS can be configured based on protocols, ports, addresses and authentication. By default, both firewalls come with predefined set of rules that allow us to utilize network resources. This includes things like browsing the web, receiving e-mails, etc. Otherstandard firewall exceptions are File and Printer Sharing, Network Discovery, Performance Logs and Alerts, Remote Administration, Windows Remote Management, Remote Assistance, Remote Desktop, Windows Media Player, Windows Media Player Network Sharing Service With firewall in Windows 10 we can configure inbound and outbound rules. By default, all outbound traffic is allowed, and inbound responses to that traffic are also allowed. Inbound traffic initiated from external sources is automatically blocked. Sometimes we will see a notification about a blocked program which is trying to access network resources. In that case we will be able to add an exceptionto our firewall in order to allow traffic from the program in the future. Windows 10 comes with somenew features when it comes to firewall. For example, "full-stealth" feature blocks other computers from performing operating system fingerprinting. OS fingerprinting is a malicious technique used todetermine the operating system running on the host machine. Another feature is "boot-time filtering".This feature ensures that the firewall is working at the same time when the network interface becomes active, which was not the case in previous versions of Windows.

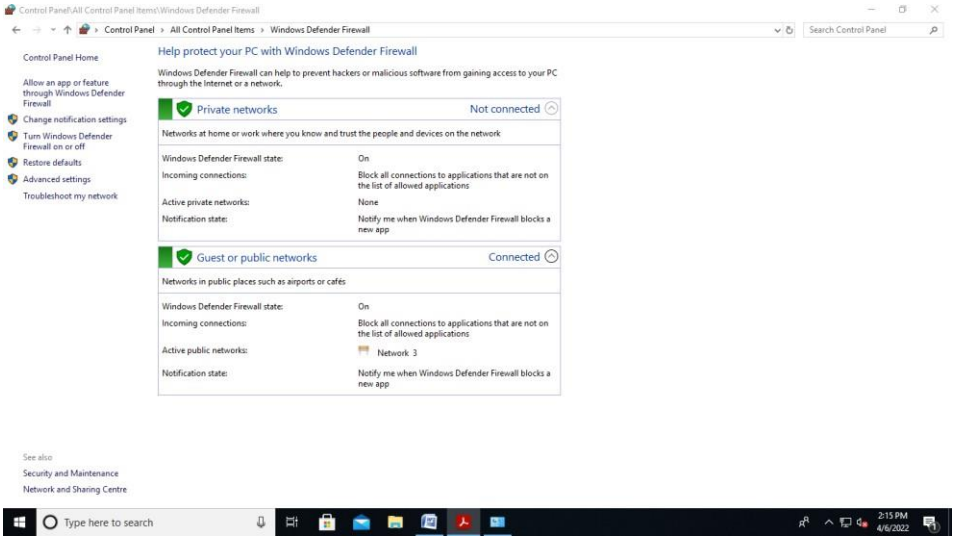
When we first connect to some network, we are prompted to select a network location. This featureis known as Network Location Awareness (NLA). This feature enables us to assign a network profileto the connection based on the location. Different network profiles contain different collections of firewall rules. In Windows 10, different network profiles can be configured on different interfaces. For example, our wired interface can have different profile than our wireless interface. There are three different network profiles available:

- Public
- Home/Work - private network
- Domain - used within a domain

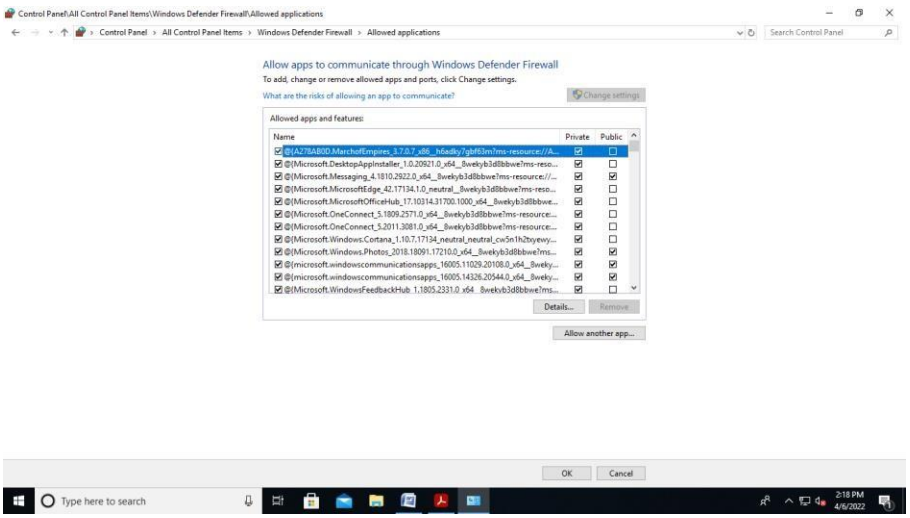
We choose those locations when we connect to a network. We can always change the location in the Network and Sharing Center, in Control Panel. The Domain profile can be automatically assigned by the NLA service when we log on to an Active Directory domain. Note that we must have administrative rights in order to configure firewall in Windows 10.

Configuring Windows Firewall:

To open Windows Firewall we can go to Start > Control Panel > Windows Defender Firewall.

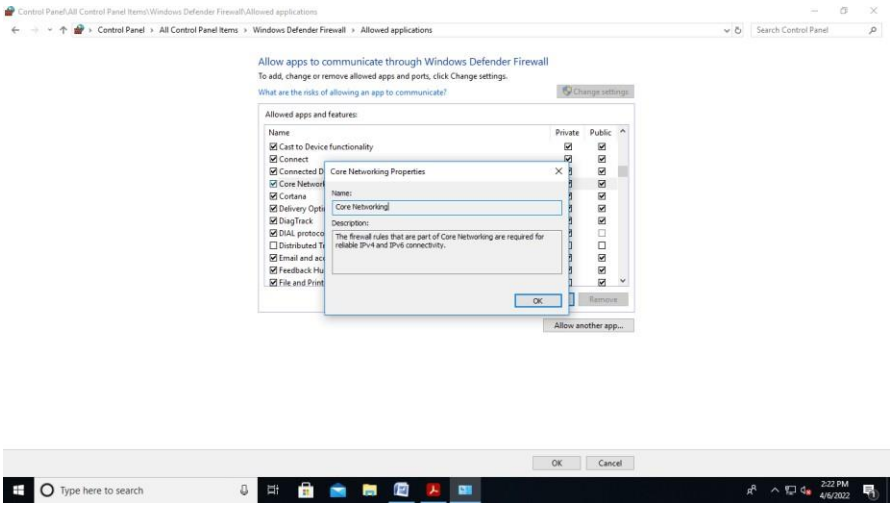


By default, Windows Firewall is enabled for both private (home or work) and public networks. It isalso configured to block all connections to programs that are not on the list of allowed programs. Toconfigure exceptions, we can go to the menu on the left and select "Allow a program or feature through Windows Firewall" option.



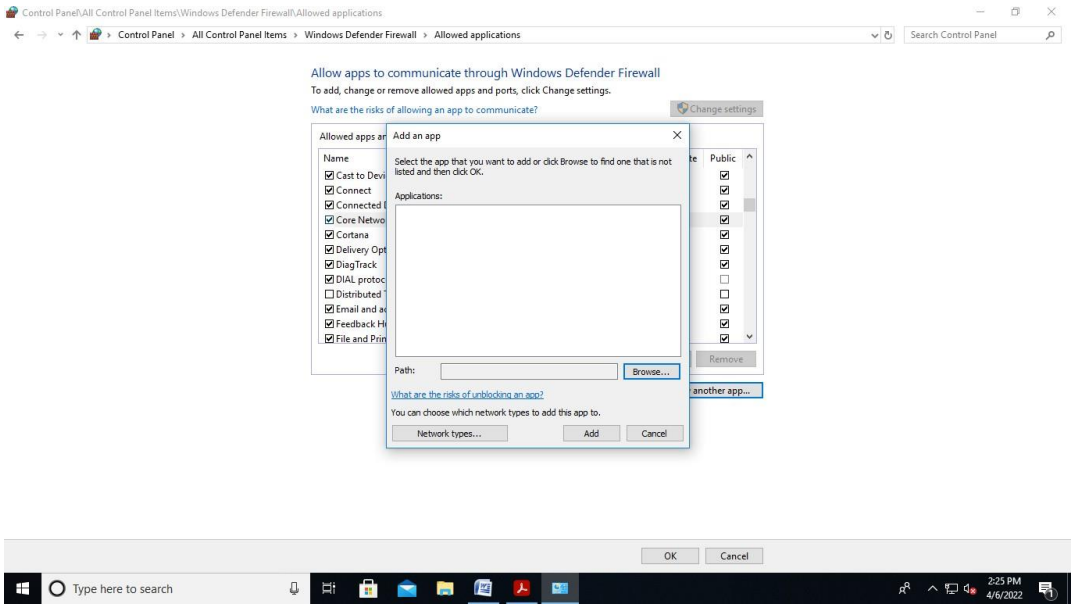
Exceptions

To change settings in this window we have to click the "Change settings" button. As you can see, here we have a list of predefined programs and features that can be allowed to communicate on private or public networks.
For example, notice that the Core Networking feature is allowed on both private and public networks,while the File and Printer Sharing is only allowed on private networks. We can also see the details of the items in the list by selecting it and then clicking the Details button.



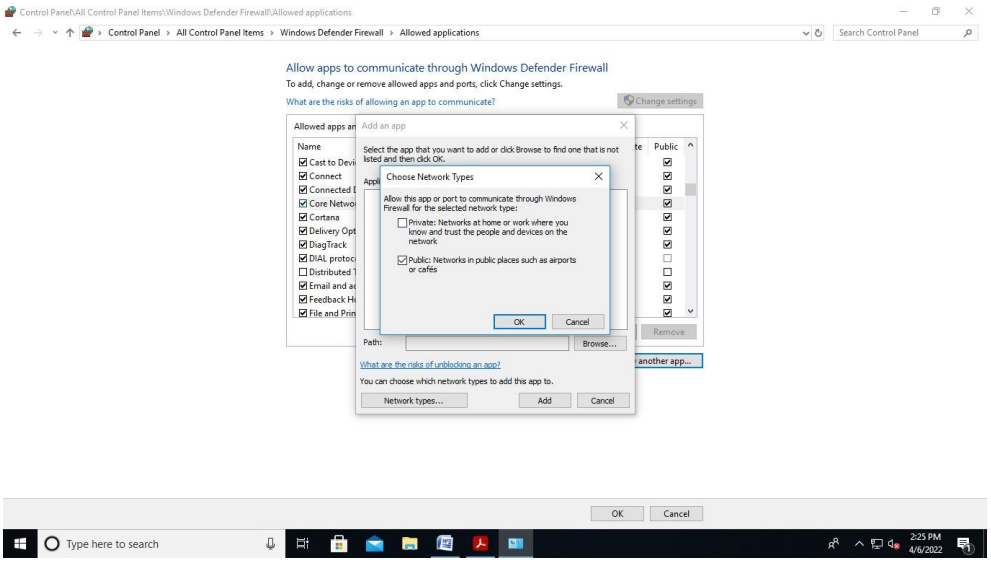
Details

If we have a program on our computer that is not in this list, we can manually add it by clicking onthe "Allow another program" button.



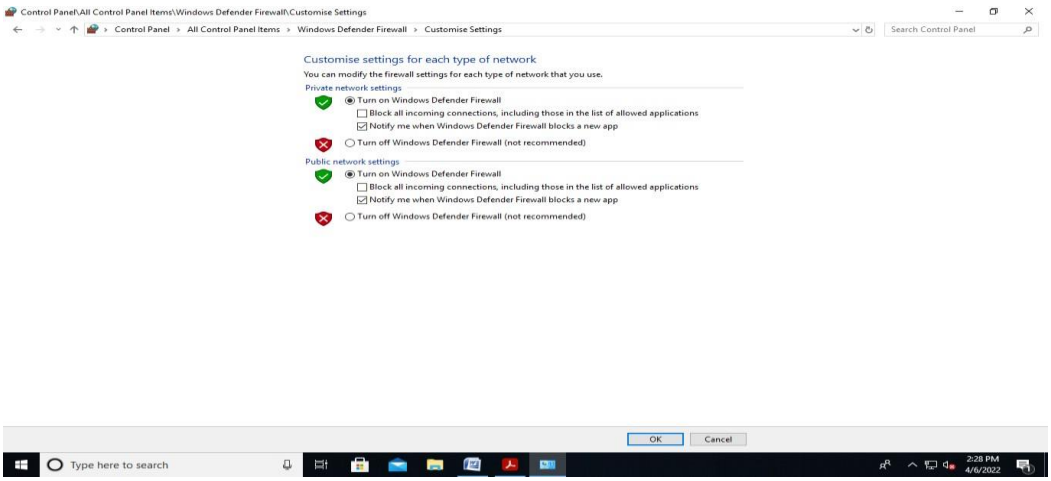
Add a Program

Here we have to browse to the executable of our program and then click the Add button. Notice thatwe can also choose location types on which this program will be allowed to communicate by clickingon the "Network location types" button.



Network Locations

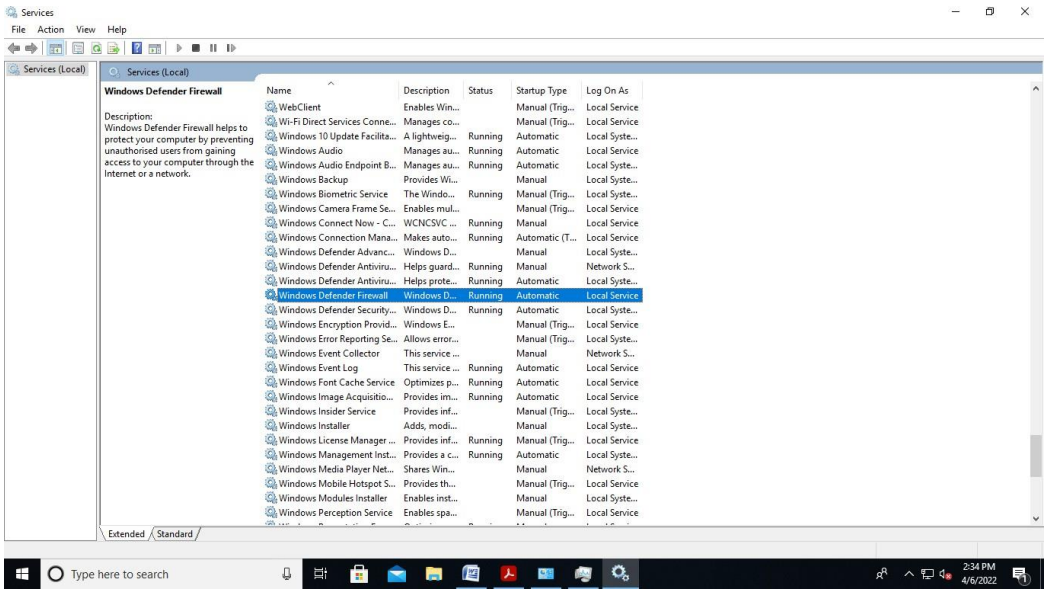
Many applications will automatically configure proper exceptions in Windows Firewall when we run them. For example, if we enable streaming from Media Player, it will automatically configure firewall settings to allow streaming. The same thing is if we enable Remote Desktop feature from the system properties window. By enabling Remote Desktop feature we actually create an exceptionin Windows Firewall.
Windows Firewall can be turned off completely. To do that we can select the "Turn Windows Firewall on or off" option from the menu on the left.



Firewall Customization

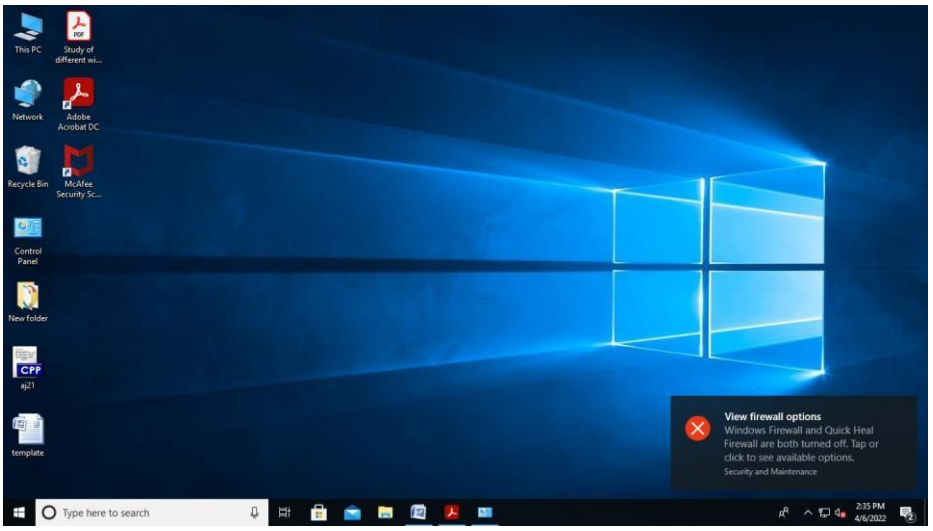
Note that we can modify settings for each type of network location (private or public). Interesting thing here is that we can block all incoming connections, including those in the list of allowed programs.

Windows Firewall is actually a Windows service. As you know, services can be stopped and started.If the Windows Firewall service is stopped, the Windows Firewall will not work.



Firewall Service

In our case the service is running. If we stop it, we will get a warning that we should turn on ourWindows Firewall.



Warning

Remember that with Windows Firewall we can only configure basic firewall settings, and this is enough for most day-to-day users. However, we can't configure exceptions based on ports in Windows Firewall any more. For that we have to use Windows Firewall with Advanced Security.

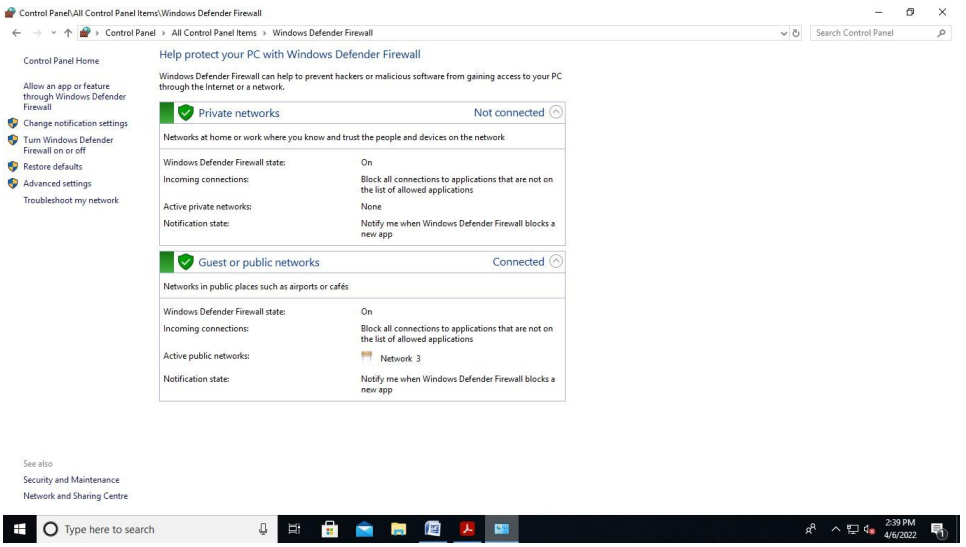
How to Start & Use the Windows Firewall with Advanced Security

The Windows Firewall with Advanced Security is a tool which gives you detailed control over therules that are applied by the Windows Firewall. You can view all the rules that are used by the Windows Firewall, change their properties, create new rules or disable existing ones. In this tutorial we will share how to open the Windows Firewall with Advanced Security, how to find your way around it and talk about the types of rules that are available and what kind of traffic theyfilter.

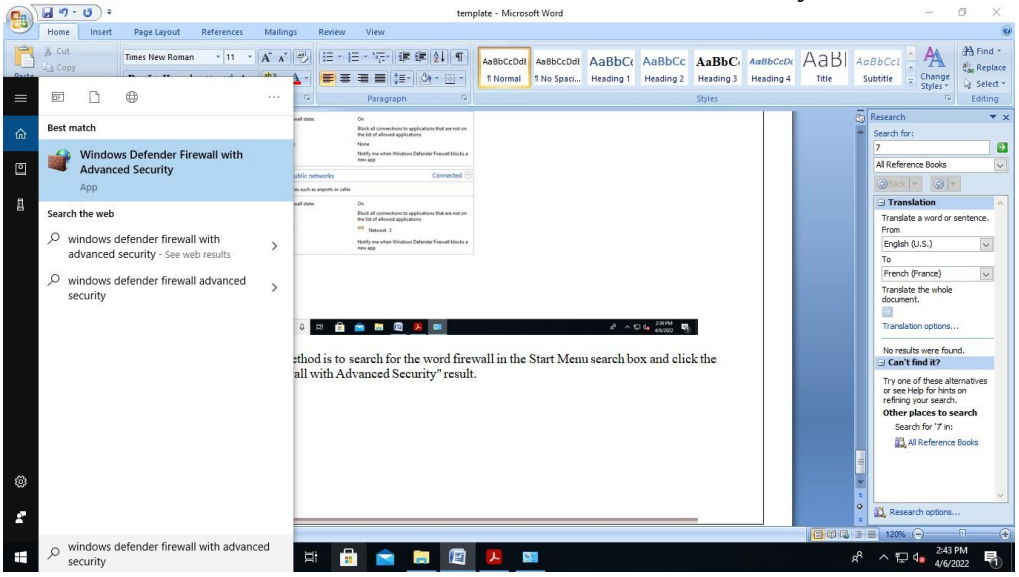
How to Access the Windows Firewall with Advanced Security

You have several alternatives to opening the Windows Firewall withAdvanced Security:

One is to open the standard Windows Firewall window, by going to "Control Panel -> System andSecurity > Windows Firewall". Then, click or tap Advanced settings.



In Windows 10, another method is to search for the word firewall in the Start Menu search box andclick the "Windows Defender Firewall with Advanced Security" result.



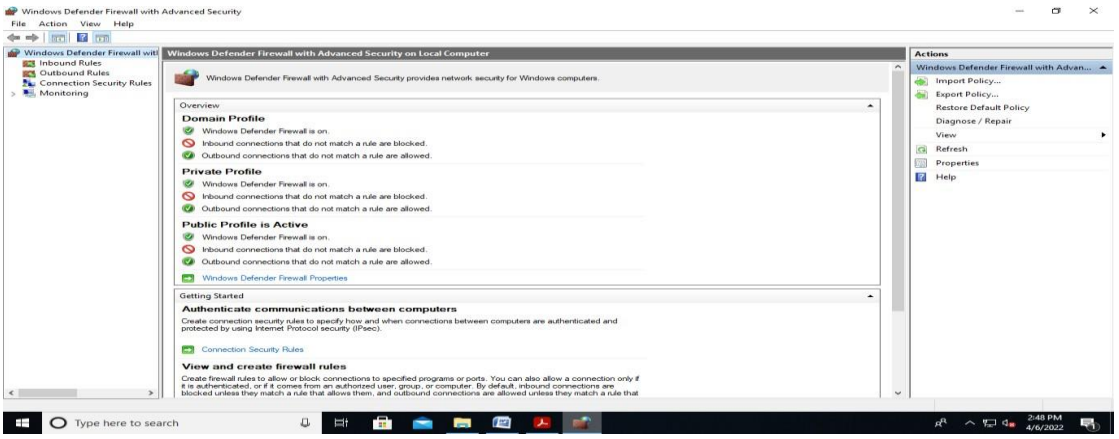
What Are The Inbound & Outbound Rules ?

In order to provide the security you need, the Windows Firewall has a standard set of inbound and outbound rules, which are enabled depending on the location of the network you are connected to.

Inbound rules are applied to the traffic that is coming from the network and the Internet to your computer or device. Outbound rules apply to the traffic from your computer to the network or the Internet.

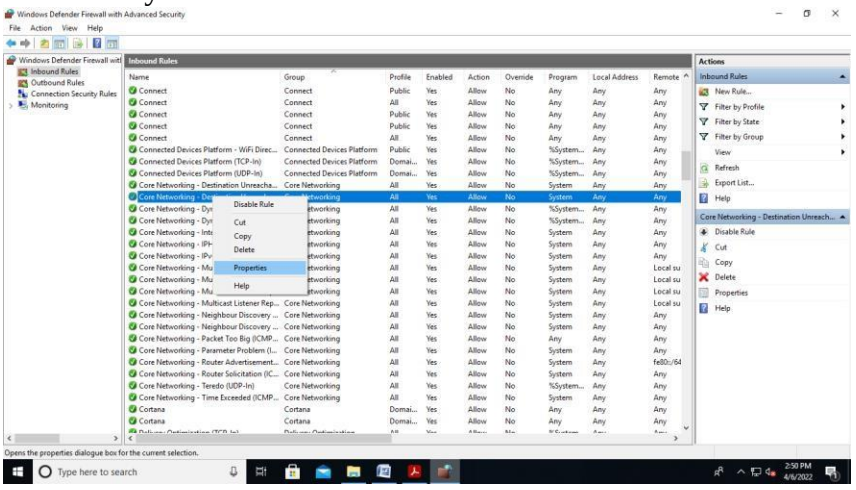
These rules can be configured so that they are specific to: computers, users, programs, services, portsor protocols. You can also specify to which type of network adapter (e.g. wireless, cable, virtual private network) or user profile it is applied to.

In the Windows Firewall with Advanced Security, you can access all rules and edit their properties.All you have to do is click or tap the appropriate unit in the left-side panel.

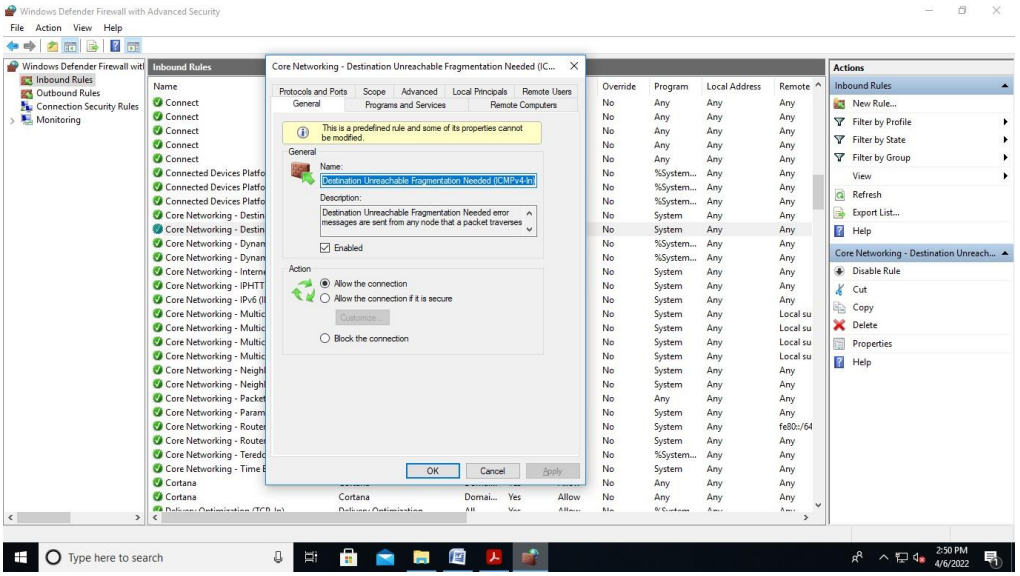


The rules used by the Windows Firewall can be enabled or disabled. The ones which are enabled oractive are marked with a green check-box in the Name column. The ones that are disabled are markedwith a gray check-box.

If you want to know more about a specific rule and learn its properties, right click on it and select Properties or select it and press Properties in the column on right, which lists the actions that are available for your selection.



In the Properties window, you will find complete information about the selected rule, what it doesand in when it is applied. You will also be able to edit its properties and change any of the available parameters.

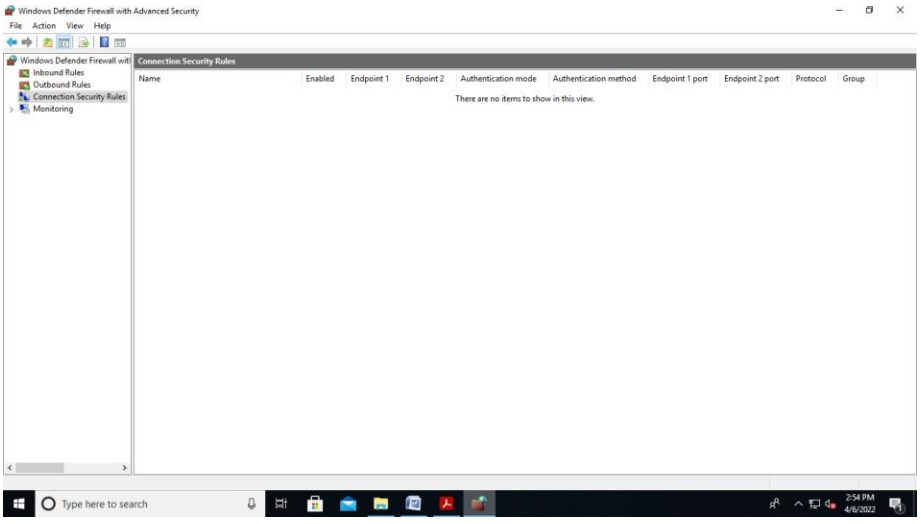


What Are The Connection Security Rules?

Connection security rules are used to secure traffic between two computers while it crosses the network. One example would be a rule which defines that connections between two specific computers must be encrypted.

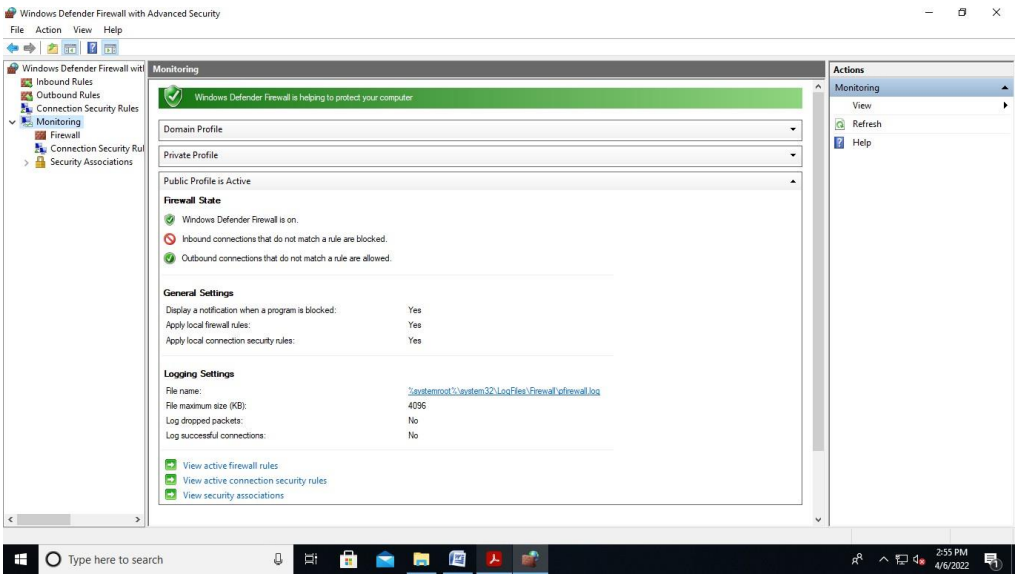
Unlike the inbound or outbound rules, which are applied only to one computer, connection securityrules require that both computers have the same rules defined and enabled.

If you want to see if there are any such rules on your computer, click or tap "Connection Security Rules" on the panel on the left. By default, there are no such rules defined on Windows computers and devices. They are generally used in business environments and such rules are set by the networkadministrator.



What Does the Windows Firewall with Advanced SecurityMonitor?

The Windows Firewall with Advanced Security includes some monitoring features as well. In the Monitoring section you can find the following information: the firewall rules that are active (both inbound and outbound), the connection security rules that are active and whether there are any activesecurity associations.



You should note that the Monitoring section shows only the active rules for the current networklocation.

CONCLUSION:

Subject : Network Security (Elective - II)

Experiment No : 08

Name : **Roll No :**

Title : Steps to ensure Security of any one web browser (Mozilla Firefox/Google Chrome)

Class : T.E **Date of Performance :** **Date of Submission :**

OBJECTIVE:

- To interpret the security vulnerabilities of browsers.
- To identify the security and privacy features and operation of Browsers
- To demonstrate how to avoid unsafe websites

AIM: Steps to ensure Security of web browser Google Chrome.

THEORY:

3.1 Browser security is an important part in keeping your information safe.

Browser security is an important part in keeping your information safe.
Your browser is the window to the internet and also the first line of defence against malware threats. Some small tweaks to your browser security settings are all that you need to make your time online that much safer.

3.2 Browser features and their security vulnerabilities

Browsers use many tools for various tasks, such as Java, Flash Player, ActiveX, etc. But these often come with security flaws, which cybercriminals exploit to get access to your PC. A quick rundown of these tools will help you figure out if you need them or not.

Deactivate ActiveX.

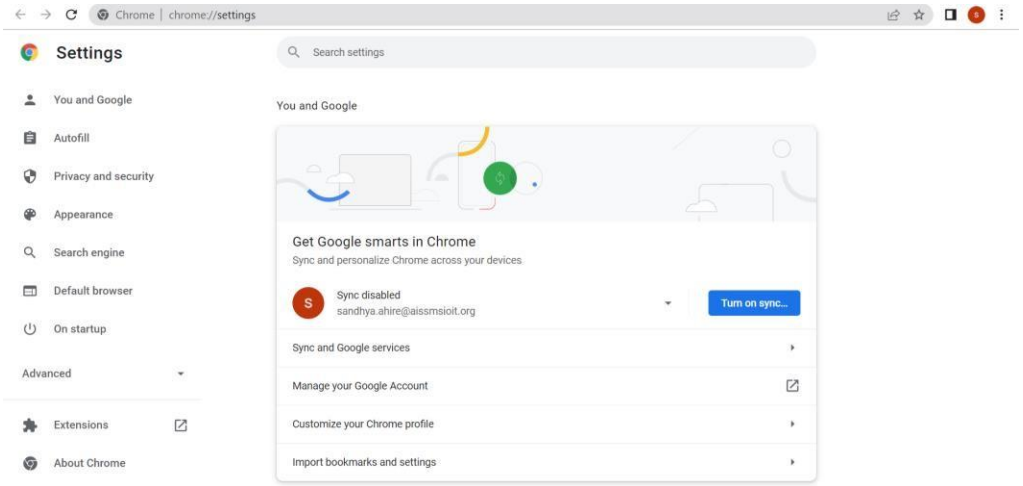
A browser add-on that comes preinstalled on Internet Explorer or Microsoft Edge and only works with these browsers. ActiveX acts as a middle man between your PC and Java/Flash based interactions in certain sites. This creates security problems by giving malicious websites a window into your PC. What's more, ActiveX is rarely used nowadays, so be on your guard if a site asks you to install it and accept the installation only if you are 150% sure that site is trustworthy.

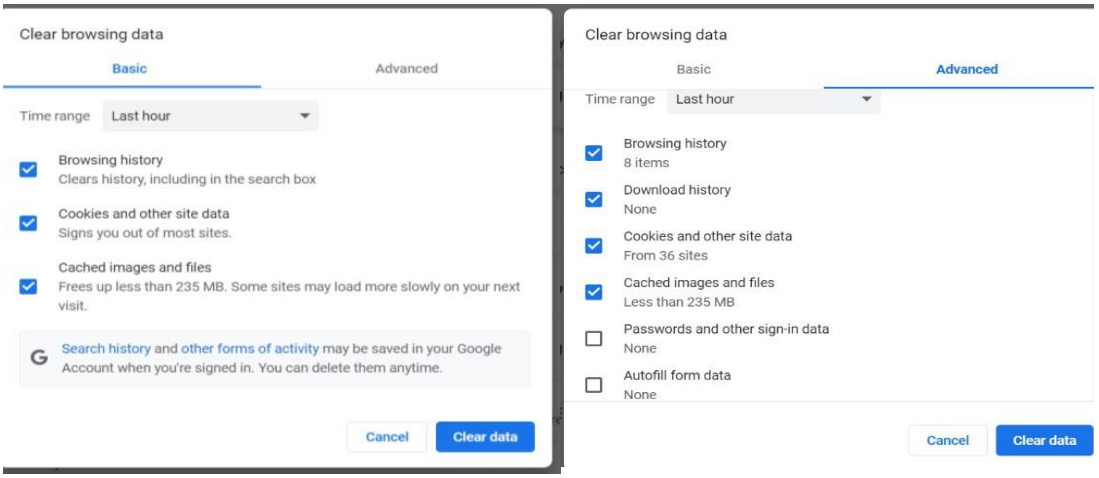
Try to disable JavaScript. JavaScript is a programming language used by websites to run various programs and features. Sites such as YouTube or Google Docs need it to function, but so do advertising, pop-up software and a whole host of other spammy elements from the internet.
Cybercriminals use JavaScript in malicious ways in order to infect your device with malware and other harmful software.
If you disable JavaScript altogether you will get a much quicker and simplified browser experience, with little to no ads, pop-ups, greatly improved page load times and generally a cleaner Internet experience at the cost of specialized tools such as Google Docs or YouTube.

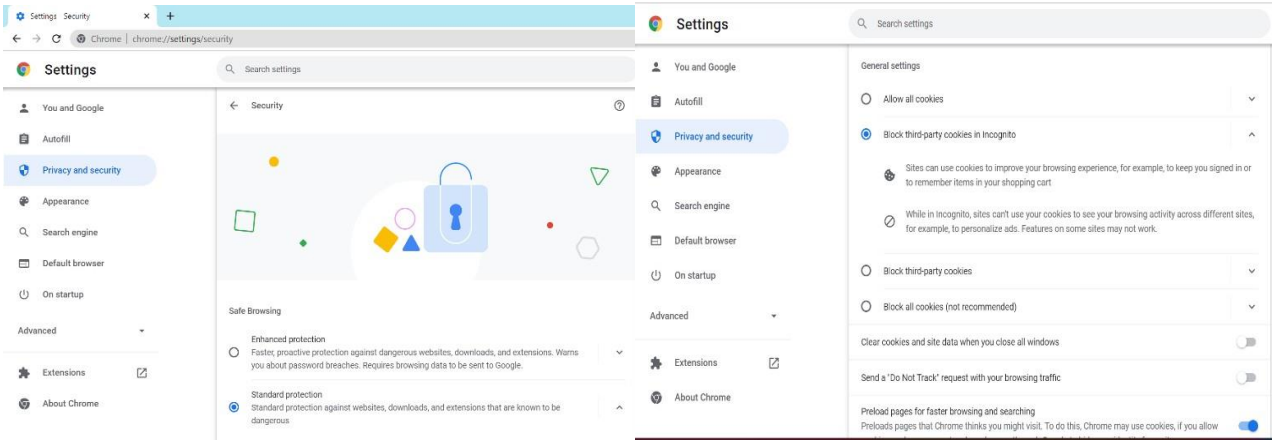
This doesn’t need to be as drastic as it sounds, since browsers do allow you to white list certain sites which can run JavaScript.
Delete Cookies. These are small data files stored on your browser. Websites use cookies in order to remember your accounts and passwords, browsing history and to track user behavior on their site. Because of the information they contain, cookies are prime targets for cybercriminals, especially the ones that contain emails, account names and passwords. When you disable and clear cookies you cut down on the personal data cybercriminals can obtain. One thing you will want to keep in mind is that there are two types of cookies:
First party and third-party cookies. First party cookies are placed by the site you visit, for instance you get a first party cookie by cnn.com while visiting cnn.com. Third party cookies are placed by other sites, for example you get a cookie from amazon.com while visiting cnn.com. First party cookies are frequently used to remember your login information so you don’t have to enter it every time you visit a site. But we can’t stress this enough, don’t allow your browser to save passwords! Third party cookies are almost always placed on your computer by advertisers or marketers interested in tracking your movement online, so nothing bad will happen if you block them.

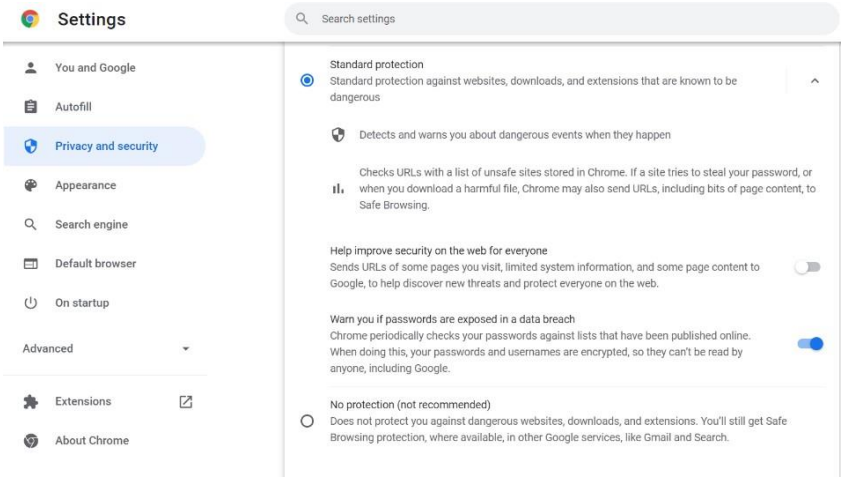
Browser extensions and add-ons add extra functionality to your browser such as ad blocking or search bars. However, these add-ons pose a security risk, since they can open up windows into your PC which can be exploited to inject malware.

3.3 Google chrome settings for better security





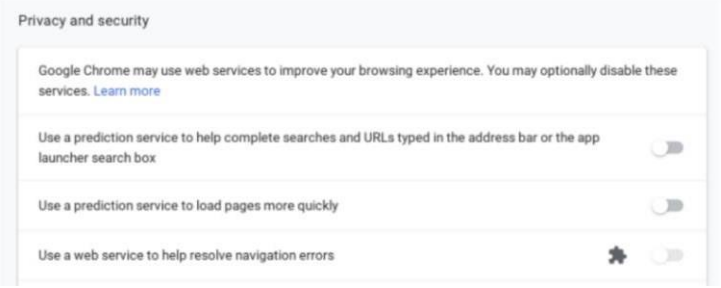




Procedure :

- Essential Privacy Settings for Chrome OS and Google Chrome
- Disable These "Privacy and Security" Settings.
- Enable "Safe Browsing" and "Do Not Track"
- Disable or Encrypt Data Syncing.
- Disable Location Tracking.
- Don't Save Addresses and Payment Methods.
- Limit Cookies.

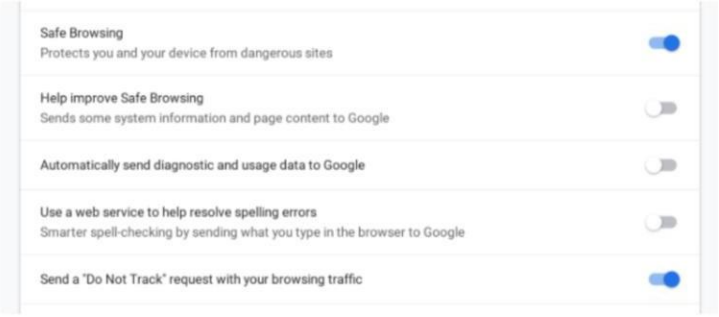
1. Disable These "Privacy and Security" Settings



Google has built several features into Chrome intended to improve your web browsing experience. The catch is that these services involve sending data to the company's servers, which it can add to your user account. Google then analyzes your data in order to sell increasingly personalized ads.
Some of these features send data to Google every time you enter a letter into the navigation bar. This means Google gets to see everything you search for and every website you visit, whether or notyou use the Google search engine, and even if you change your mind and decide not to visit a site orstart a search.
You can disable these options by opening Chrome settings and going to the Privacy and securitysection.
Features to disable:

- Use a prediction service to help complete searches and URLs typed in the address bar orthe app launcher search box
- Use a prediction service to load pages more quickly
- Use a web service to help resolve navigation errors
- Help improve Safe Browsing
- Automatically send diagnostic and usage data to Google
- Use a web service to help resolve spelling errors

2. Enable "Safe Browsing" and "Do Not Track"



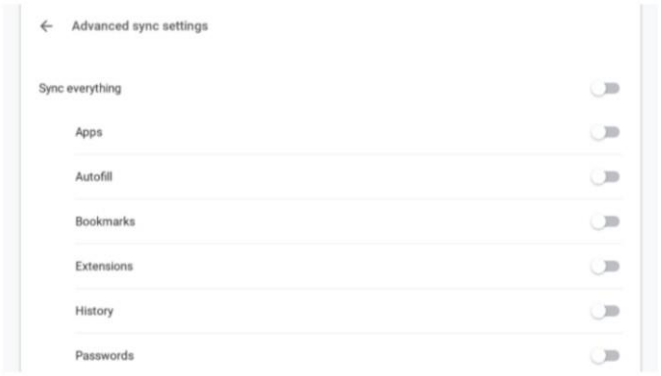
Under **Privacy and security**, there are also a few settings you will likely want to enable.

Safe Browsing is one of them. This feature can prevent certain malicious or poorly secured sites from opening in your browser.

Under Privacy and security, there are also a few settings you will likely want to enable.
Safe Browsing is one of them. This feature can prevent certain malicious or poorly secured sites from opening in your browser.

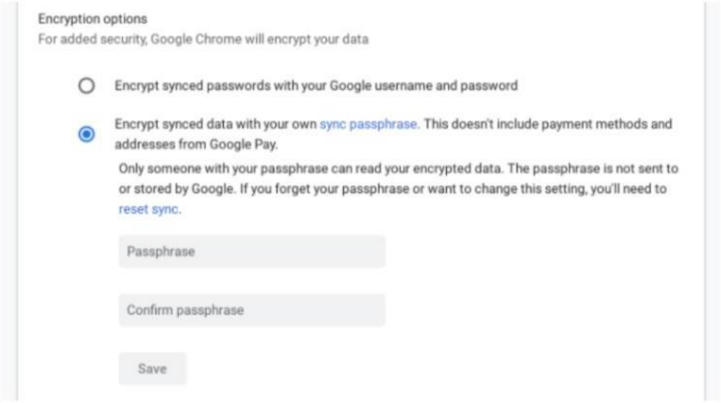
Do Not Track is another. Websites sometimes monitor your behavior. They may know how much time you spend on any given page and what type of information most interests you.
Sometimes they do this to provide you with a better experience, but in the process, they're able to build a profile that you might prefer they didn't have. With Do Not Track enabled, you're telling websites not to track your behavior.

3. Disable or Encrypt Data Syncing



Web browsers may primarily serve to connect us to the web, but your bookmarks and browsing history are typically saved on your computer. You don't have to store everything online. If you enable syncing, you're taking data from your computer and giving it to Google.

Web browsers may primarily serve to connect us to the web, but your bookmarks and browsing history are typically saved on your computer. You don't have to store everything online. If you enablesyncing, you're taking data from your computer and giving it to Google. Disable syncing to keep a copy off of Google's servers.
You can disable syncing by going to People > Sync. There you can turn off Sync everything and untoggled various categories.
If you use numerous devices and value having your browsing data synced across all of them, you can instead choose to encrypt all of your synced data with a passphrase. You can find this option underneath all of the toggles mentioned above.

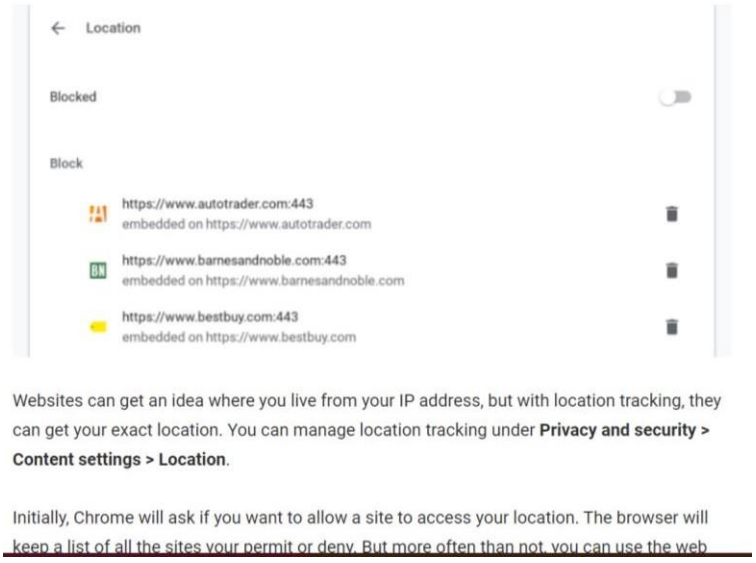


Chrome will ask you to create a passphrase that you will need to enter on every device you choose to sync. To keep this data private, make sure the passphrase is not the same as the

Chrome will ask you to create a passphrase that you will need to enter on every device you chooseto sync. To keep this data private, make sure the passphrase is not the same as the one you choosefor your Google account. This way Google's servers will store your data, but the company won't have the passphrase needed to decrypt your files.

Warning: Be careful not to forget your passphrase. Since your passphrase is not stored online,Google cannot help you recover it. This means you will lose your synced data.

4. Disable Location Tracking



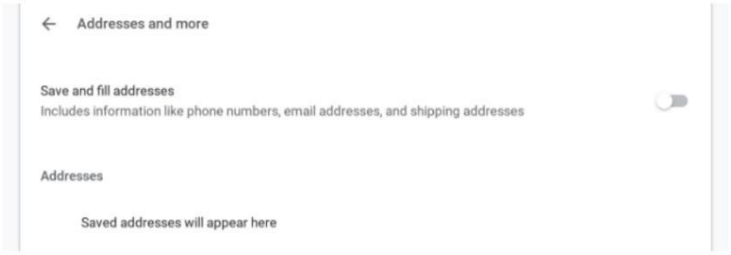
Websites can get an idea where you live from your IP address, but with location tracking, they canget your exact location. You can manage location tracking under Privacy and security > Content settings > Location.
Initially, Chrome will ask if you want to allow a site to access your location. The browser will keepa list of all the sites your permit or deny. But more often than not, you can use the web just fine by blocking this functionality entirely.
You can use Google Maps and similar sites by entering your address manually, just like in the daysbefore our devices came with GPS built-in.

5. Don't Save Addresses and Payment Methods

Whether you love the internet or prefer to stay offline, these days it's hard to avoid filling out online forms. Chrome will try to make this task easier for you by remembering information you fill out often, such as your email address, your physical address, phone numbers, and credit cards.

As tempting as this may be, it means you're creating a record of your personal information that isn'tnecessary. Even if you disable syncing, someone with access to your computer can peak at this information. This could be dangerous if you leave your computer in a public place, but it can also lead to unintended consequences when sharing your device with friends or family members.

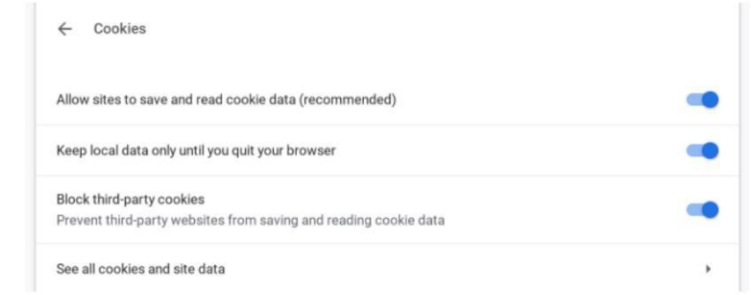
You can tell Chrome not to remember most of this information by going to People > Addresses andmore.
To stop Chrome from storing your credit cards, go to People > Payment methods. Both locations allow you to delete any information that Chrome may have already stored.



Whether you love the internet or prefer to stay offline, these days it's hard to avoid filling out online forms. Chrome will try to make this task easier for you by remembering information you fill out often, such as your email address, your physical address, phone numbers, and credit cards.

1.Limit Cookies

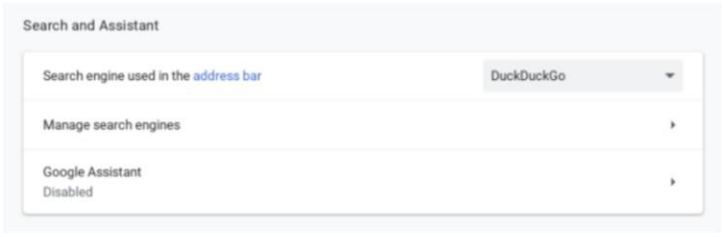
When we talk about websites and ad networks tracking your behavior, we're typically talking aboutthe use of cookies. Your browser stores these files so that websites function as you would expect. Without them, you're starting from a clean slate whenever you visit a page.
Cookies are important for sites that let you sign into an account or add items into a cart.
But sites can store whatever they want in these files. So can ad networks. That's why it's a good practice to limit which cookies are permitted onto your computer.
To do this, go to Privacy and security > Content settings > Cookies. Enable Block third-party cookies. To better cover your tracks, you can also enable Keep local data only until you quit your browser, but know that this means you will have to sign into sites again the next time you open Chrome.
You can see all of the cookies Chrome has saved by selecting See all cookies and site data. Here youcan delete cookies one at a time or clear them all.



When we talk about websites and ad networks tracking your behavior, we're typically talking about the use of cookies. Your browser stores these files so that websites function as you would expect. Without them, you're starting from a clean slate whenever you visit a page.

Cookies are important for sites that let you sign into an account or add items into a cart.

1. Change Default Search Engine



Chrome defaults to the Google search engine. That provides Google with every search that we enter into the navigation bar. This information is so personal, it means Google knows certain things about us that would surprise our loved ones or closest colleagues.

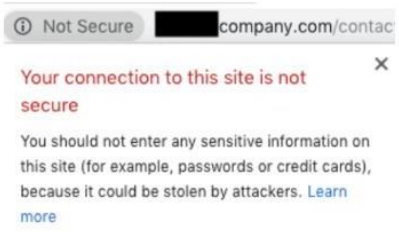
You can cut Google off from this information by changing your default search engine. You could try Bing, if you prefer, though that requires giving your data to Microsoft rather than Google. Alternatively, you can try a search engine that prioritizes privacy.

Chrome defaults to the Google search engine. That provides Google with every search that we enterinto the navigation bar. This information is so personal, it means Google knows certain things aboutus that would surprise our closest colleagues.
You can cut Google off from this information by changing your default search engine. You could try Bing, if you prefer, though that requires giving your data to Microsoft rather than Google. Alternatively, you can try a search engine that prioritizes privacy.
You can change your default search engine by going to Search and Assistant > Manage search engines. You can also get here by right-clicking the navigation bar and selecting Edit search engines... in the context menu.
Under Search and Assistant, you also have the option to change "Search engine used in the addressbar." You can also choose to disable Google Assistant, if your computer supports that feature.

Chrome Privacy and Security: Using Google Chrome Safely

Chrome is frequently updated, and updates often fix security vulnerabilities, so install them as soonas possible. Chrome automatically updates itself when you open it, but if you leave Chrome open for a long time (days) without closing it, it won't be able to update itself. When an update is available,you'll see an arrow icon in the top right corner of Chrome. Click it to update.

Chrome has made a big deal about securing data between the browser and websites via HTTPS. Theaddress bar will warn you when the site you're on is Not Secure. Don't enter sensitive info (financial,medical, personally-identifiable) in pages that don't show the padlock icon and https in the addressbar.
However, not all sites that use HTTPS are legitimate! Malicious sites, such as phishing and scam sites, frequently use HTTPS. So you should still ensure that the site you're on is legitimate, regardlessof whether it uses HTTPS.



CONCLUSION:

Subject : Network Security (Elective - II)

Experiment No : 09

Name : **Roll No :**

Title : Implementation of Symmetric and Asymmetric cryptography

Class : T.E **Date of Performance :** **Date of Submission :**

Aim : Implementation of Symmetric and Asymmetric cryptography.

Theory :

What is Symmetric Key Cryptography?

Unauthorized access to all types of data is an ever-present risk in today's cyber world. Financial and payment system data are the most vulnerable data, which may reveal consumers' and clients' personal identifying information (PII) or payment card records.

Encryption is critical for securing personally identifiable information and mitigating the threats for companies that perform payment transactions every minute of the day. This makes cryptography crucial. There are mainly two types of cryptography: symmetric and asymmetric cryptography.

Symmetric Key Cryptography

Symmetric Key Cryptography, or Symmetric Encryption, uses a secret key for both encryption and decryption. This approach is the inverse of Asymmetric Encryption, which uses one key to encrypt and another to decrypt. Data is translated to a format that cannot be interpreted or inspected by someone who does not have the secret key used to encrypt it during this phase.

The strength of the random number generator used to generate the secret key determines the effectiveness of this method. Symmetric Key Cryptography, commonly used on the Internet today, comprises two kinds of algorithms: Block and Stream. The Advanced Encryption Standard (AES) and the Data Encryption Standard (DES) are two common encryption algorithms. This type of encryption is typically much faster than Asymmetric Encryption, but it allows the secret key to be held by both the sender and the data receiver. Symmetric cryptography is based on a single shared key that all parties are aware of and can use to encrypt and decrypt data. Secret-key, single-key, shared-key, one-key, and private-key encryption are other words for symmetric-key cryptography. The usage of the last and first words will lead to misunderstanding compared to the related language used in public-key cryptography.

Symmetric key encryption employs one of the following encryption techniques:

Stream ciphers: Encrypt a message's digits or letters one at a time.

Block ciphers: Encrypt a group of bits as a single entity, inserting the plaintext to make it a block size multiple. 64-bit blocks are widely used. The NIST-approved Advanced Encryption Standard (AES) algorithm and the GCM block cipher mode of operation all use 128-bit blocks.

Asymmetric Key Cryptography :

Asymmetric cryptography, better known as public-key cryptography, encrypts and decrypts a message using a pair of similar keys. In asymmetric key cryptography, the private key is kept by one public key and one private key — to prevent unauthorized entry or usage. Anybody can use a public key to encrypt a document so that only the expected receiver can decrypt it with their private key. A private key or secret key is only known to the key's generator.

When anyone tries to submit an encrypted message, they will use a shared directory to retrieve the recipient's public key and use it to encrypt the message until submitting it. The message will then be decrypted by the receiver using their associated private key.

However, when the sender encrypts the message using their private key, the message may only be decrypted using the sender's public key, thus authenticating the sender. These encryption and decryption procedures are automatic; users don't need to lock and unlock the message manually. Numerous protocols, including the transport layer security (TLS) and safe sockets layer (SSL) protocols that allow HTTPS, depend on asymmetric cryptography. Encryption is often used in browsers that need to create a stable link over an unstable network, such as the Internet, or to verify a digital signature. The key advantage of asymmetric cryptography is increased data security. Since users are never expected to disclose or exchange their private keys, the risks of cyber activity on a user's private key during transmission are reduced.

Write a program to encryption and decryption of Symmetric key Cryptography

```
# Hexadecimal to binary conversion
def hex2bin(s):
    mp = {'0' : "0000",
          '1' : "0001",
          '2' : "0010",
          '3' : "0011",
          '4' : "0100",
          '5' : "0101",
          '6' : "0110",
          '7' : "0111",
          '8' : "1000",
          '9' : "1001",
          'A' : "1010",
          'B' : "1011",
          'C' : "1100",
          'D' : "1101",
          'E' : "1110",
          'F' : "1111" }
    bin = ""
    for i in range(len(s)):
        bin = bin + mp[s[i]]
    return bin

# Binary to hexadecimal conversion
def bin2hex(s):
    mp = {"0000" : '0',
          "0001" : '1',
```

```
                "0010" : '2',
                "0011" : '3',
                "0100" : '4',
                "0101" : '5',
                "0110" : '6',
                "0111" : '7',
                "1000" : '8',
                "1001" : '9',
                "1010" : 'A',
                "1011" : 'B',
                "1100" : 'C',
                "1101" : 'D',
                "1110" : 'E',
                "1111" : 'F' }
        hex = ""
        for i in range(0,len(s),4):
                ch = ""
                ch = ch + s[i]
                ch = ch + s[i + 1]
                ch = ch + s[i + 2]
                ch = ch + s[i + 3]
                hex = hex + mp[ch]

        return hex

# Binary to decimal conversion
def bin2dec(binary):

        binary1 = binary
        decimal, i, n = 0, 0, 0
        while(binary != 0):
                dec = binary % 10
                decimal = decimal + dec * pow(2, i)
                binary = binary//10
                i += 1
        return decimal

# Decimal to binary conversion
def dec2bin(num):
        res = bin(num).replace("0b", "")
        if(len(res)%4 != 0):
                div = len(res) / 4
                div = int(div)
                counter =(4 * (div + 1)) - len(res)
                for i in range(0, counter):
                        res = '0' + res
        return res

# Permute function to rearrange the bits
```

```
def permute(k, arr, n):
	permutation = ""
	for i in range(0, n):
		permutation = permutation + k[arr[i] - 1]
	return permutation

# shifting the bits towards left by nth shifts
def shift_left(k, nth_shifts):
	s = ""
	for i in range(nth_shifts):
		for j in range(1,len(k)):
			s = s + k[j]
		s = s + k[0]
		k = s
		s = ""
	return k

# calculating xow of two strings of binary number a and b
def xor(a, b):
	ans = ""
	for i in range(len(a)):
		if a[i] == b[i]:
			ans = ans + "0"
		else:
			ans = ans + "1"
	return ans
```

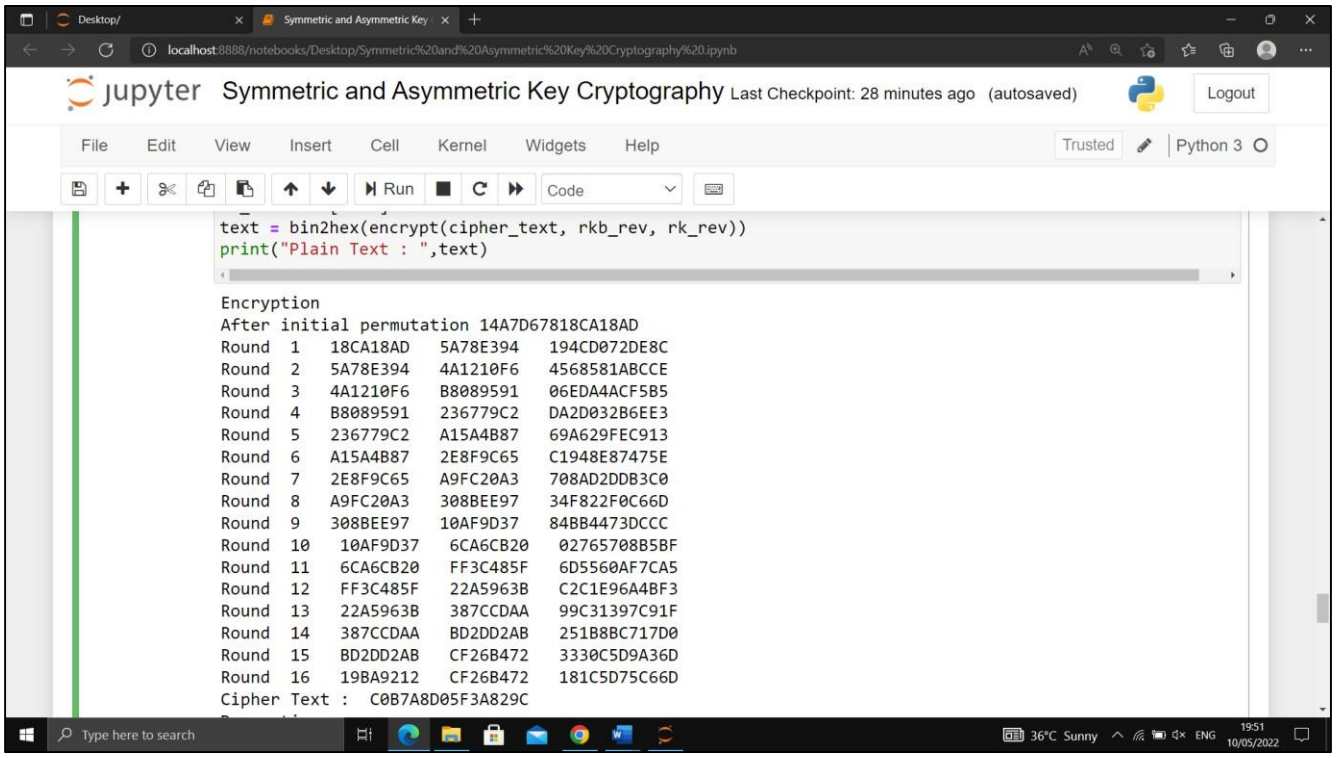
```
# Table of Position of 64 bits at initial level: Initial Permutation Table
initial_perm = [58, 50, 42, 34, 26, 18, 10, 2,
                60, 52, 44, 36, 28, 20, 12, 4,
                62, 54, 46, 38, 30, 22, 14, 6,
                64, 56, 48, 40, 32, 24, 16, 8,
                57, 49, 41, 33, 25, 17, 9, 1,
                59, 51, 43, 35, 27, 19, 11, 3,
                61, 53, 45, 37, 29, 21, 13, 5,
                63, 55, 47, 39, 31, 23, 15, 7]

# Expansion D-box Table
exp_d = [32, 1 , 2 , 3 , 4 , 5 , 4 , 5,
         6 , 7 , 8 , 9 , 8 , 9 , 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, 30, 31, 32, 1 ]

# Straight Permutation Table
per = [ 16, 7, 20, 21,
        29, 12, 28, 17,
        1, 15, 23, 26,
        5, 18, 31, 10,
        2, 8, 24, 14,
        32, 27, 3, 9,
        19, 13, 30, 6,
        22, 11, 4, 25 ]

# S-box Table
sbox = [[[14, 4, 13, 1, 2, 15, 11, 8, 3, 10, 6, 12, 5, 9, 0, 7],
         [ 0, 15, 7, 4, 14, 2, 13, 1, 10, 6, 12, 11, 9, 5, 3, 8],
         [ 4, 1, 14, 8, 13, 6, 2, 11, 15, 12, 9, 7, 3, 10, 5, 0],
         [15, 12, 8, 2, 4, 9, 1, 7, 5, 11, 3, 14, 10, 0, 6, 13 ]],

         [[15, 1, 8, 14, 6, 11, 3, 4, 9, 7, 2, 13, 12, 0, 5, 10],
             [3, 13, 4, 7, 15, 2, 8, 14, 12, 0, 1, 10, 6, 9, 11, 5],
             [0, 14, 7, 11, 10, 4, 13, 1, 5, 8, 12, 6, 9, 3, 2, 15],
         [13, 8, 10, 1, 3, 15, 4, 2, 11, 6, 7, 12, 0, 5, 14, 9 ]],

         [ [10, 0, 9, 14, 6, 3, 15, 5, 1, 13, 12, 7, 11, 4, 2, 8],
         [13, 7, 0, 9, 3, 4, 6, 10, 2, 8, 5, 14, 12, 11, 15, 1],
         [13, 6, 4, 9, 8, 15, 3, 0, 11, 1, 2, 12, 5, 10, 14, 7],
             [1, 10, 13, 0, 6, 9, 8, 7, 4, 15, 14, 3, 11, 5, 2, 12 ]],

         [ [7, 13, 14, 3, 0, 6, 9, 10, 1, 2, 8, 5, 11, 12, 4, 15],
         [13, 8, 11, 5, 6, 15, 0, 3, 4, 7, 2, 12, 1, 10, 14, 9],
         [10, 6, 9, 0, 12, 11, 7, 13, 15, 1, 3, 14, 5, 2, 8, 4],
             [3, 15, 0, 6, 10, 1, 13, 8, 9, 4, 5, 11, 12, 7, 2, 14] ],
```

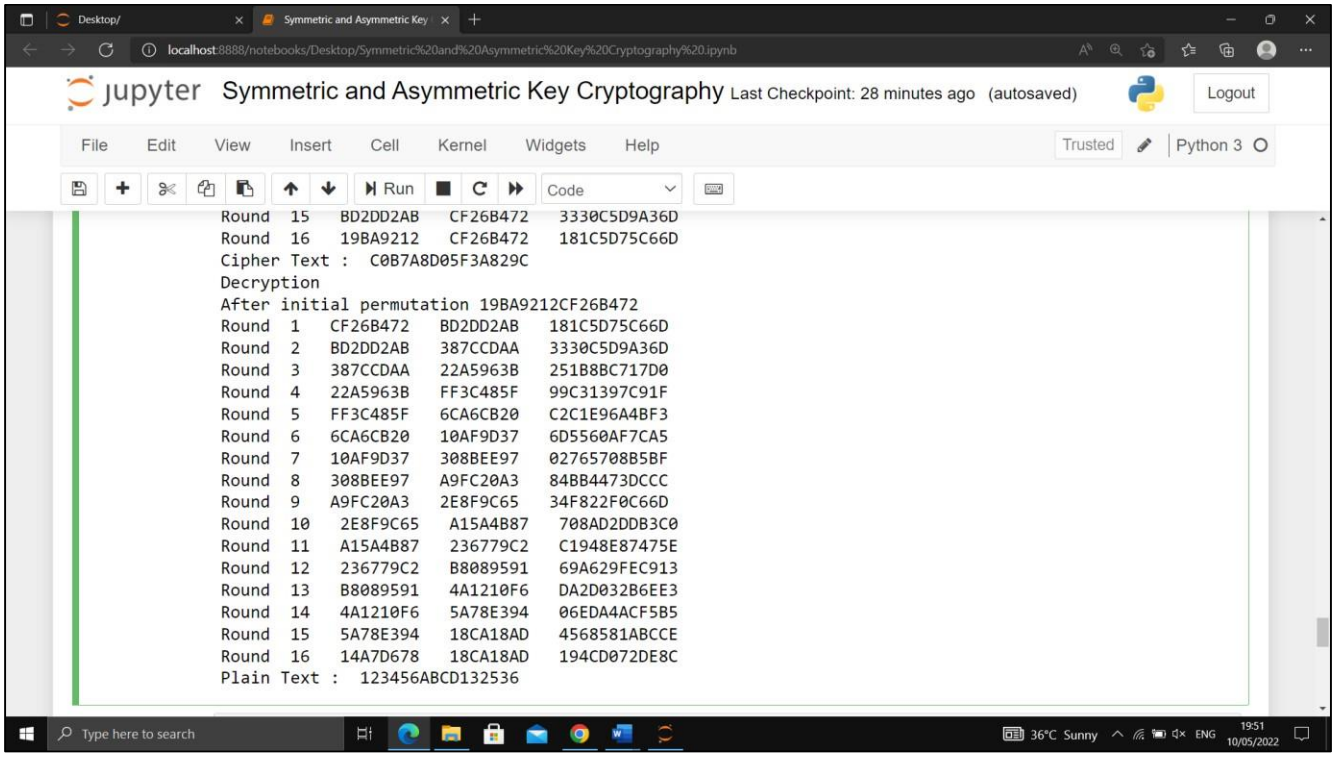
```
                [ [2, 12, 4, 1, 7, 10, 11, 6, 8, 5, 3, 15, 13, 0, 14, 9],
                [14, 11, 2, 12, 4, 7, 13, 1, 5, 0, 15, 10, 3, 9, 8, 6],
                        [4, 2, 1, 11, 10, 13, 7, 8, 15, 9, 12, 5, 6, 3, 0, 14],
                [11, 8, 12, 7, 1, 14, 2, 13, 6, 15, 0, 9, 10, 4, 5, 3 ]],

                [ [12, 1, 10, 15, 9, 2, 6, 8, 0, 13, 3, 4, 14, 7, 5, 11],
                [10, 15, 4, 2, 7, 12, 9, 5, 6, 1, 13, 14, 0, 11, 3, 8],
                        [9, 14, 15, 5, 2, 8, 12, 3, 7, 0, 4, 10, 1, 13, 11, 6],
                        [4, 3, 2, 12, 9, 5, 15, 10, 11, 14, 1, 7, 6, 0, 8, 13] ],

                [ [4, 11, 2, 14, 15, 0, 8, 13, 3, 12, 9, 7, 5, 10, 6, 1],
                [13, 0, 11, 7, 4, 9, 1, 10, 14, 3, 5, 12, 2, 15, 8, 6],
                        [1, 4, 11, 13, 12, 3, 7, 14, 10, 15, 6, 8, 0, 5, 9, 2],
                        [6, 11, 13, 8, 1, 4, 10, 7, 9, 5, 0, 15, 14, 2, 3, 12] ],

                [ [13, 2, 8, 4, 6, 15, 11, 1, 10, 9, 3, 14, 5, 0, 12, 7],
                        [1, 15, 13, 8, 10, 3, 7, 4, 12, 5, 6, 11, 0, 14, 9, 2],
                        [7, 11, 4, 1, 9, 12, 14, 2, 0, 6, 10, 13, 15, 3, 5, 8],
                        [2, 1, 14, 7, 4, 10, 8, 13, 15, 12, 9, 0, 3, 5, 6, 11] ] ]

# Final Permutation Table
final_perm = [ 40, 8, 48, 16, 56, 24, 64, 32,
                        39, 7, 47, 15, 55, 23, 63, 31,
                        38, 6, 46, 14, 54, 22, 62, 30,
                        37, 5, 45, 13, 53, 21, 61, 29,
                        36, 4, 44, 12, 52, 20, 60, 28,
                        35, 3, 43, 11, 51, 19, 59, 27,
                        34, 2, 42, 10, 50, 18, 58, 26,
                        33, 1, 41, 9, 49, 17, 57, 25 ]

def encrypt(pt, rkb, rk):
        pt = hex2bin(pt)

        # Initial Permutation
        pt = permute(pt, initial_perm, 64)
        print("After initial permutation", bin2hex(pt))

        # Splitting
        left = pt[0:32]
        right = pt[32:64]
        for i in range(0, 16):
                # Expansion D-box: Expanding the 32 bits data into 48 bits
                right_expanded = permute(right, exp_d, 48)

                # XOR RoundKey[i] and right_expanded
                xor_x = xor(right_expanded, rkb[i])
```

```
* 6 + 4]))       # S-boxex: substituting the value from s-box table by calculating row and column
                sbox_str = ""
                for j in range(0, 8):
                        row = bin2dec(int(xor_x[j * 6] + xor_x[j * 6 + 5]))
                        col = bin2dec(int(xor_x[j * 6 + 1] + xor_x[j * 6 + 2] + xor_x[j * 6 + 3] + xor_x[j

                        val = sbox[j][row][col]
                        sbox_str = sbox_str + dec2bin(val)

                # Straight D-box: After substituting rearranging the bits
                sbox_str = permute(sbox_str, per, 32)

                # XOR left and sbox_str
                result = xor(left, sbox_str)
                left = result

                # Swapper
                if(i != 15):
                        left, right = right, left
                print("Round ", i + 1, " ", bin2hex(left), " ", bin2hex(right), " ", rk[i])

        # Combination
        combine = left + right
        # Final permutation: final rearranging of bits to get cipher text
        cipher_text = permute(combine, final_perm, 64)
        return cipher_text

pt = "123456ABCD132536"
key = "AABB09182736CCDD"

# Key generation
# --hex to binary
key = hex2bin(key)

# --parity bit drop table
keyp = [57, 49, 41, 33, 25, 17, 9,
                1, 58, 50, 42, 34, 26, 18,
                10, 2, 59, 51, 43, 35, 27,
                19, 11, 3, 60, 52, 44, 36,
                63, 55, 47, 39, 31, 23, 15,
                7, 62, 54, 46, 38, 30, 22,
                14, 6, 61, 53, 45, 37, 29,
                21, 13, 5, 28, 20, 12, 4 ]

# getting 56 bit key from 64 bit using the parity bits
key = permute(key, keyp, 56)
```

```
# Number of bit shifts
shift_table = [1, 1, 2, 2,
                2, 2, 2, 2,
               1, 2, 2, 2,
               2, 2, 2, 1 ]
 # Key- Compression Table : Compression of key from 56 bits to 48 bits
 key_comp = [14, 17, 11, 24, 1, 5,
                3, 28, 15, 6, 21, 10,
                23, 19, 12, 4, 26, 8,
                16, 7, 27, 20, 13, 2,
                41, 52, 31, 37, 47, 55,
                30, 40, 51, 45, 33, 48,
                44, 49, 39, 56, 34, 53,
                46, 42, 50, 36, 29, 32 ]

 # Splitting
 left = key[0:28] # rkb for RoundKeys in binary
 right = key[28:56] # rk for RoundKeys in hexadecimal

 rkb = []
 rk = []
 for i in range(0, 16):
        # Shifting the bits by nth shifts by checking from shift table
        left = shift_left(left, shift_table[i])

        right = shift_left(right, shift_table[i])

        # Combination of left and right string
        combine_str = left + right

        # Compression of key from 56 to 48 bits
        round_key = permute(combine_str, key_comp, 48)

        rkb.append(round_key)
        rk.append(bin2hex(round_key))

 print("Encryption")
 cipher_text = bin2hex(encrypt(pt, rkb, rk))
 print("Cipher Text : ",cipher_text)

 print("Decryption")
 rkb_rev = rkb[::-1]
 rk_rev = rk[::-1]
 text = bin2hex(encrypt(cipher_text, rkb_rev, rk_rev))
 print("Plain Text : ",text)
```

Output :

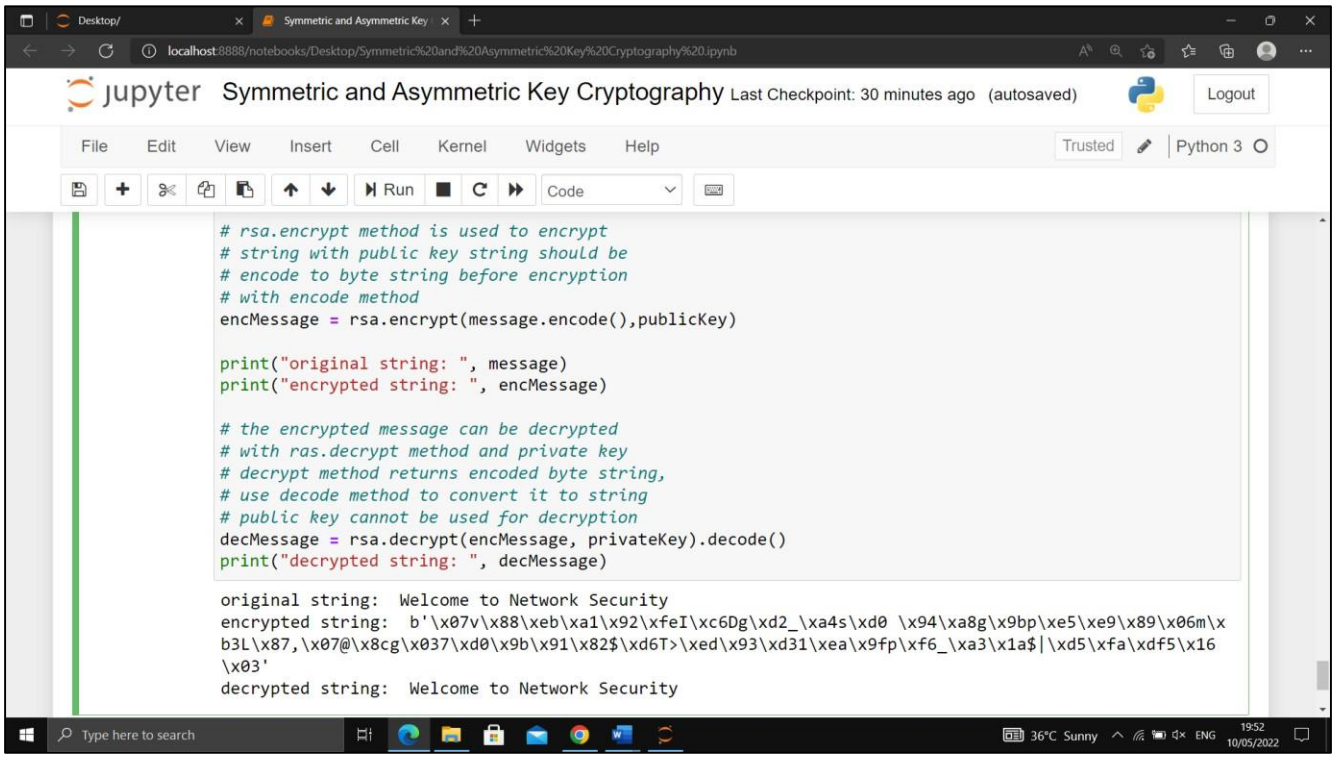

Desktop/
Symmetric and Asymmetric Key
localhost:8888/notebooks/Desktop/Symmetric%20and%20Asymmetric%20Key%20Cryptography%20.ipynb
jupyter Symmetric and Asymmetric Key Cryptography Last Checkpoint: 28 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
text = bin2hex(encrypt(cipher_text, rkb_rev, rk_rev))
print("Plain Text : ",text)
Encryption
After initial permutation 14A7D67818CA18AD
Round 1 18CA18AD 5A78E394 194CD072DE8C
Round 2 5A78E394 4A1210F6 4568581ABCCE
Round 3 4A1210F6 B8089591 06EDA4ACF5B5
Round 4 B8089591 236779C2 DA2D032B6EE3
Round 5 236779C2 A15A4B87 69A629FEC913
Round 6 A15A4B87 2E8F9C65 C1948E87475E
Round 7 2E8F9C65 A9FC20A3 708AD2DDB3C0
Round 8 A9FC20A3 308BEE97 34F822F0C66D
Round 9 308BEE97 10AF9D37 84BB4473DCCC
Round 10 10AF9D37 6CA6CB20 02765708B5BF
Round 11 6CA6CB20 FF3C485F 6D5560AF7CA5
Round 12 FF3C485F 22A5963B C2C1E96A4BF3
Round 13 22A5963B 387CCDAA 99C31397C91F
Round 14 387CCDAA BD2DD2AB 251B8BC717D0
Round 15 BD2DD2AB CF26B472 3330C5D9A36D
Round 16 19BA9212 CF26B472 181C5D75C66D
Cipher Text : C0B7A8D05F3A829C
Type here to search
36°C Sunny
ENG
19:51
10/05/2022


Desktop/
Symmetric and Asymmetric Key
localhost:8888/notebooks/Desktop/Symmetric%20and%20Asymmetric%20Key%20Cryptography%20.ipynb
jupyter Symmetric and Asymmetric Key Cryptography Last Checkpoint: 28 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Round 15 BD2DD2AB CF26B472 3330C5D9A36D
Round 16 19BA9212 CF26B472 181C5D75C66D
Cipher Text : C0B7A8D05F3A829C
Decryption
After initial permutation 19BA9212CF26B472
Round 1 CF26B472 BD2DD2AB 181C5D75C66D
Round 2 BD2DD2AB 387CCDAA 3330C5D9A36D
Round 3 387CCDAA 22A5963B 251B8BC717D0
Round 4 22A5963B FF3C485F 99C31397C91F
Round 5 FF3C485F 6CA6CB20 C2C1E96A4BF3
Round 6 6CA6CB20 10AF9D37 6D5560AF7CA5
Round 7 10AF9D37 308BEE97 02765708B5BF
Round 8 308BEE97 A9FC20A3 84BB4473DCCC
Round 9 A9FC20A3 2E8F9C65 34F822F0C66D
Round 10 2E8F9C65 A15A4B87 708AD2DDB3C0
Round 11 A15A4B87 236779C2 C1948E87475E
Round 12 236779C2 B8089591 69A629FEC913
Round 13 B8089591 4A1210F6 DA2D032B6EE3
Round 14 4A1210F6 5A78E394 06EDA4ACF5B5
Round 15 5A78E394 18CA18AD 4568581ABCCE
Round 16 14A7D678 18CA18AD 194CD072DE8C
Plain Text : 123456ABCD132536
Type here to search
36°C Sunny
ENG
19:51
10/05/2022

Write a program to encryption and decryption of Asymmetric key Cryptography

```
import rsa
# generate public and private keys with rsa.newkeys method,this method accepts, key length as its parameter, key length should be atleast 16
publicKey, privateKey = rsa.newkeys(512)
# this is the string that we will be encrypting
message = "Welcome to Network Security"
# rsa.encrypt method is used to encrypt
# string with public key string should be
# encode to byte string before encryption with encode method
encMessage = rsa.encrypt(message.encode(),publicKey)
print("original string: ", message)
print("encrypted string: ", encMessage)
# the encrypted message can be decrypted
# with ras.decrypt method and private key
# decrypt method returns encoded byte string,
# use decode method to convert it to string
# public key cannot be used for decryption
decMessage = rsa.decrypt(encMessage, privateKey).decode()
print("decrypted string: ", decMessage)
```

Output :



Conclusion :

AISSMS
COLLEGE OF ENGINEERING
ज्ञानम् सकलजनहिताय
Accredited by NAAC with "A+" Grade

Subject : Network Security (Elective - II)

Experiment No : 10

Name : **Roll No :**

Title : Case Study of AES

Class : T.E **Date of Performance :** **Date of Submission :**

Aim : Case Study of AES.

Theory :

Origins :

- A clear replacement for DES was needed
  - Since the Key size was too small
  - The variants are just patches
- It can use Triple-DES – but slow, has small blocks
- US NIST issued call for ciphers in 1997
- 15 candidates accepted in Jun 98
- 5 were shortlisted in Aug-99

AES Competition Requirements :

- private key symmetric block cipher
- 128-bit data, 128/192/256-bit keys
- Stronger & faster than Triple-DES
- provide full specification & design details
- both C & Java implementations
- NIST have released all submissions & unclassified analyses

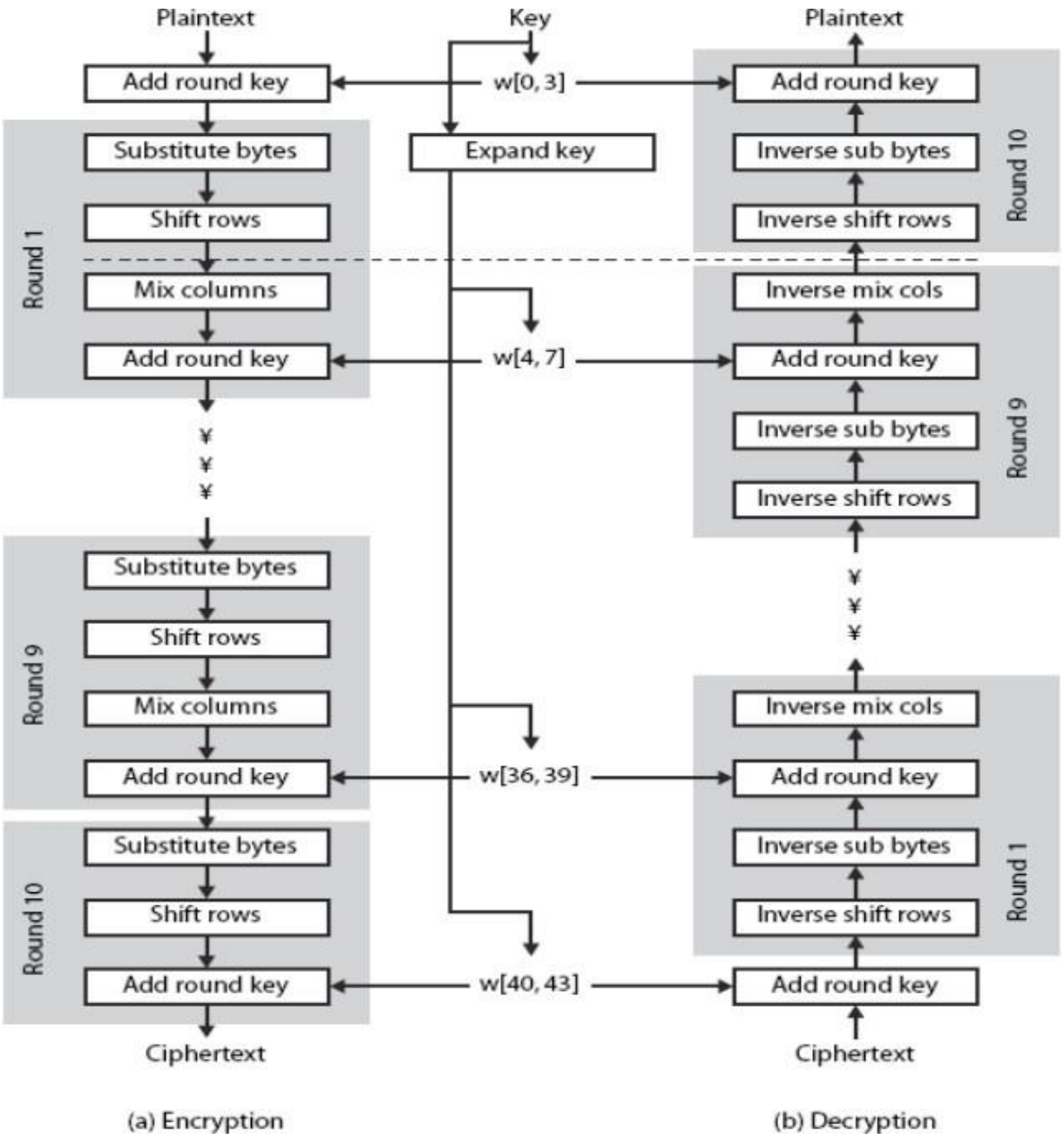
AES Evaluation Criteria :

- Initial criteria:
  - security – effort for practical cryptanalysis
  - cost – in terms of computational efficiency
  - algorithm & implementation characteristics

- Final criteria
    - general security
    - ease of software & hardware implementation
    - implementation attacks
    - flexibility (in en/decrypt, keying, other factors)

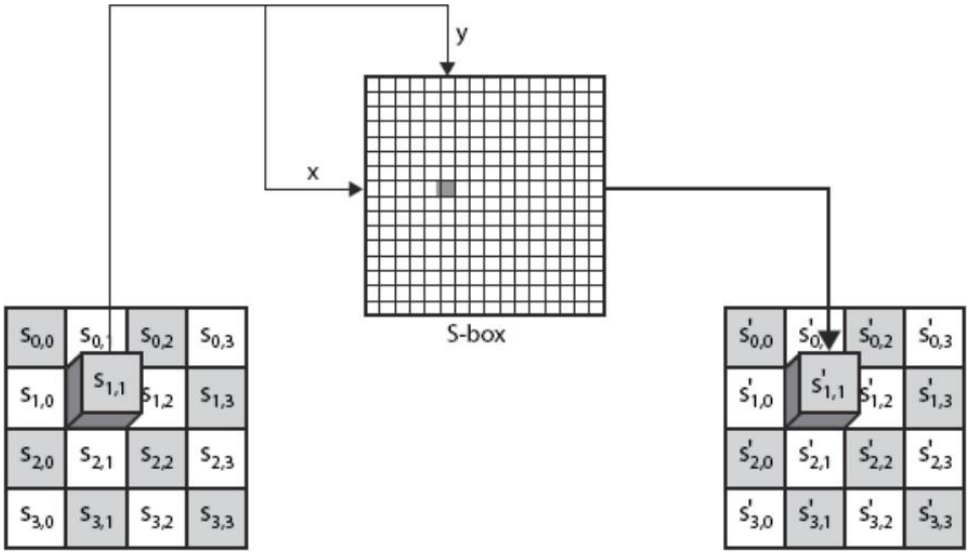
The AES Cipher - Rijndael :

- Rijndael was selected as the AES in Oct-2000. Designed by Joan Rijmen and Vincent Daemen in Belgium
- It has 128/192/256 bit keys, 128 bit data
- It is an iterative rather than Feistel cipher
    - processes data as block of 4 columns of 4 bytes
    - operates on entire data block in every round
- Designed to be:
    - resistant against known attacks
    - speed and code compactness on many CPUs
    - design simplicity
- Data block viewed as 4-by-4 table of bytes
- Such a table is called the current state
- key is expanded to array of words
- It has 10 rounds in which state the following transformations (called `layers'):
    - BS- byte substitution (1 S-box used on every byte)
    - SR- shift rows (permute bytes between groups/columns)
    - MC- mix columns (uses matrix multiplication in GF(256))
    - ARK- add round key (XOR state with round key)
- First and last round are a little different



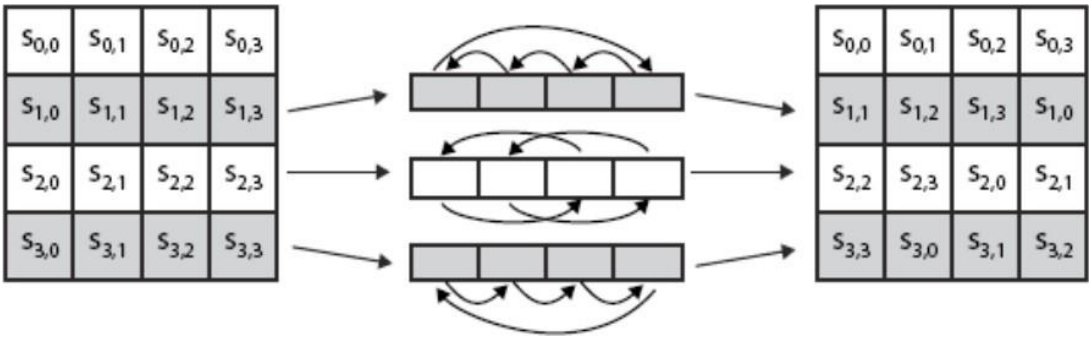
Byte Substitution :

- A simple substitution of each byte
- Uses one s-box of 16x16 bytes containing a permutation of all 256 8-bit values
- Each byte of state is replaced by byte indexed by row (left 4-bits) & column (right 4-bits)
    - Eg. Byte {95} is replaced by byte in row 9 column 5 which has value {2A}
- S-box constructed using defined transformation of values in GF(256)
- S-box constructed using a simple math formula using a non-linear function : 1/x.
- Construction of S-Box (on board)



Shift Rows:

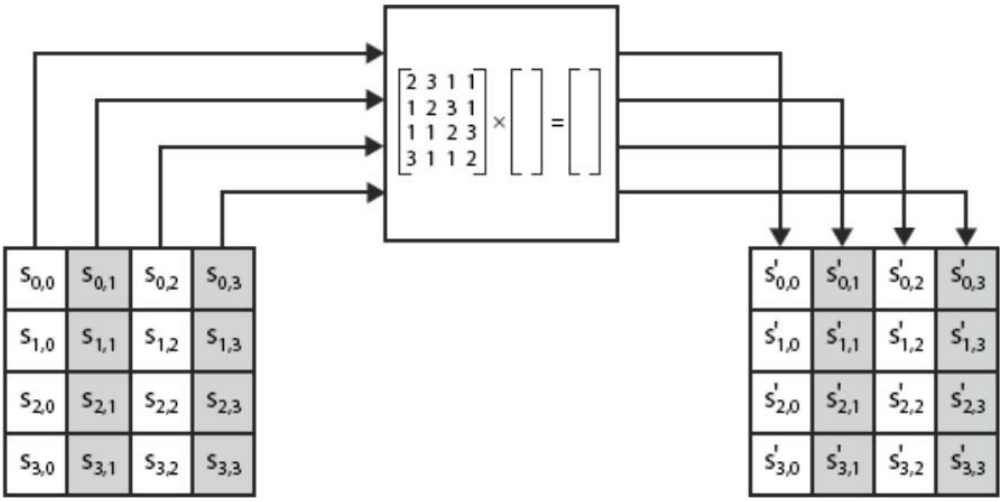
- A circular byte shift in each row
  - 1st row is unchanged
  - 2nd row does 1 byte circular shift to left
  - 3rd row does 2 byte circular shift to left
  - 4th row does 3 byte circular shift to left
- Decrypt inverts using shifts to right.
- Since state is processed by columns, this step permutes bytes between the columns.



Mix Columns :

- Each column is processed separately.
- Each byte is replaced by a value dependent on all 4 bytes in the column.
- Effectively a matrix multiplication in GF(28) using prime poly m(x) =x8+x4+x3+x+1

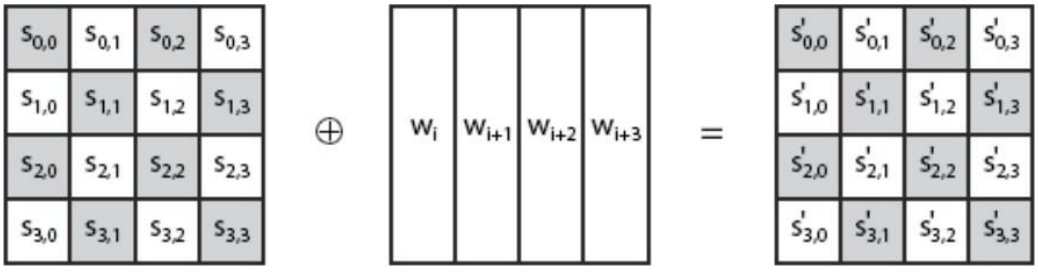
$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$



- Expresses each col of the new state as 4 equations.
  - One equation to derive each new byte in col
- Decryption requires use of inverse matrix with larger coefficients, hence a little harder
- Have an alternate characterization
  - each column a 4-term polynomial
  - with coefficients in GF(28)
  - and polynomials multiplied modulo (x4+1)

Add Round Key :

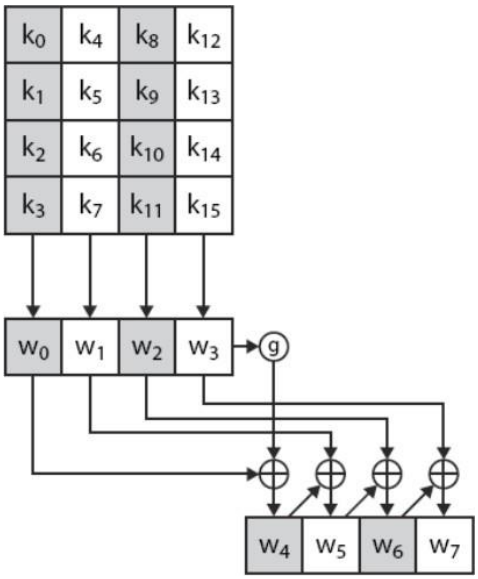
- Xor state with 128-bits of the round key.
- Again processed by column (though effectively a series of byte operations).
- Inverse for decryption identical
  - since XOR own inverse, with reversed keys
- Designed to be as simple as possible.



AES Round :

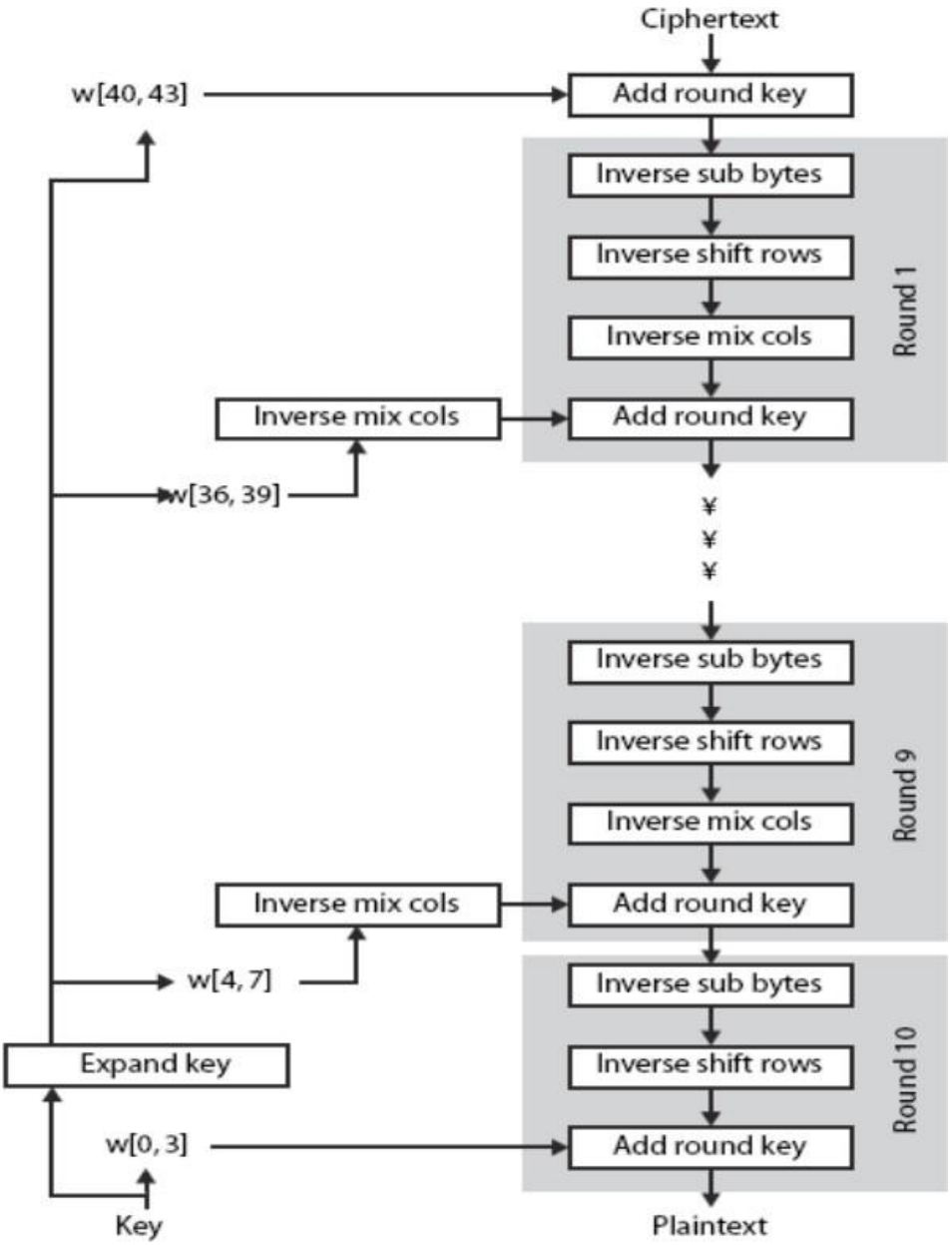
AES Key Scheduling : Takes 128-bit (16-byte) key and expands intoarray of 44 32-bit words

AES Key Expansion :



AES Decryption :

- AES decryption is not identical to encryption since steps done in reverse.
- But can define an equivalent inverse cipher with steps as for encryption.
  - but using inverses of each step
  - with a different key schedule
- It works since result is unchanged when
  - swap byte substitution & shift rows
  - swap mix columns & add (tweaked) round key



AES - Design Considerations :

- Not a Feistel scheme: so diffusion is faster, but it's a new scheme, so less analyzed.
- S-box: mathematically constructed: based on the x → x^(-1) transformation.
- Shift row- to resist two recent attack: truncated differential and the square attack.
- Key scheduling – nonlinear (uses the S-box) mixing of the key bits.
- 10 rounds: there are attacks better than brutesearch for Rijndael-with-7-rounds, so extra 3 rounds for safety.

Conclusion :

Subject : Network Security (Elective - II)

Experiment No : 11

Name : **Roll No :**

Title : Case Study of Hash Functions

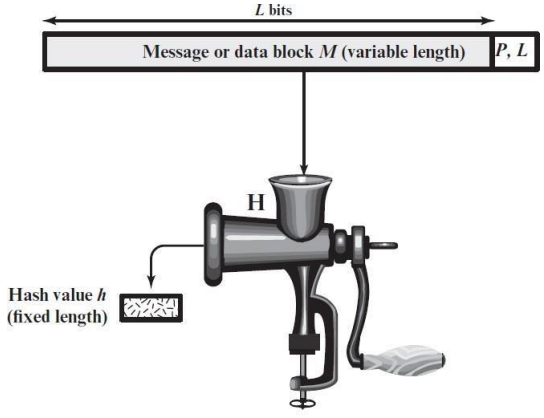
Class : T.E **Date of Performance :** **Date of Submission :**

Aim : Case Study of Hash Functions.

Theory :

Cryptographic Hash Functions:

- A **hash function** H accepts a variable-length block of data *M* as input and produces a fixed-size hash value *h* = H(*M*).
- A "good" hash function has the property that the results of applying the function to a large set of inputs will produce outputs that are evenly distributed and apparently random. In general terms, the principal object of a hash function is data integrity. A change to any bit or bits in *M* results, with high
- probability, in a change to the hash value.
- The kind of hash function needed for security applications is referred to as a **cryptographic hash function**.
- A cryptographic hash function is an algorithm for which it is computationally infeasible (because no attack is significantly more efficient than brute force) to find either
    - a data object that maps to a pre-specified hash result (the one-way property) or
    - two data objects that map to the same hash result (the collision-free property).
- Because of these characteristics, hash functions are often used to determine whether or not data has changed.



P, L = padding plus length field

Cryptographic Hash function h=H(M)

The above figure depicts the general operation of a cryptographic hash function.

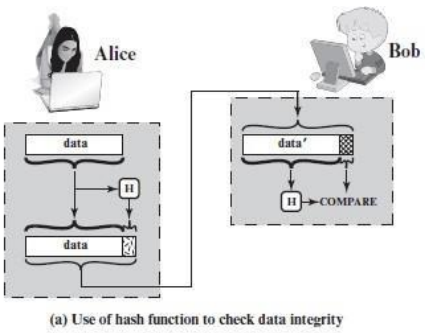
- Typically, the input is padded out to an integer multiple of some fixed length (e.g., 1024 bits), and thepadding includes the value of the length of the original message in bits.
- The length field is a security measure to increase the difficulty for an attacker to produce an alternativemessage with the same hash value.

Applications of Cryptographic Hash Functions:

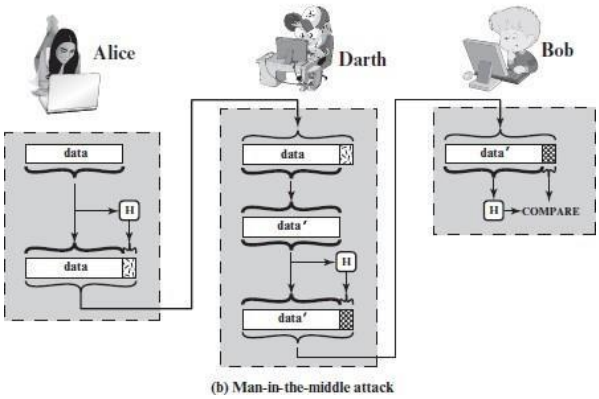
The most versatile cryptographic algorithm is the cryptographic hash function. It is used in a wide variety ofsecurity applications and Internet protocols. The following are various applications where it is employed.

Message Authentication:

- Message authentication is a mechanism or service used to verify the integrity of a message.
- Message authentication assures that data received are exactly as sent (i.e., there is no modification,insertion, deletion, or replay).
- When a hash function is used to provide message authentication, the hash function value is often referredto as a **message digest**.
- The essence of the use of a hash function for message integrity is as follows.
    - The sender computes a hash value as a function of the bits in the message and transmits both the hashvalue and the message.
    - The receiver performs the same hash calculation on the message bits and compares this value with the incoming hash value.
    - If there is a mismatch, the receiver knows that the message (or possibly the hash value) has been altered (Figure a).
    - The hash value must be transmitted in a secure fashion. That is, the hash value must be protected sothat if an adversary alters or replaces the message, it is not feasible for adversary to also alter the hashvalue to fool the receiver. This type of attack is shown in Figure b.



(a) Use of hash function to check data integrity

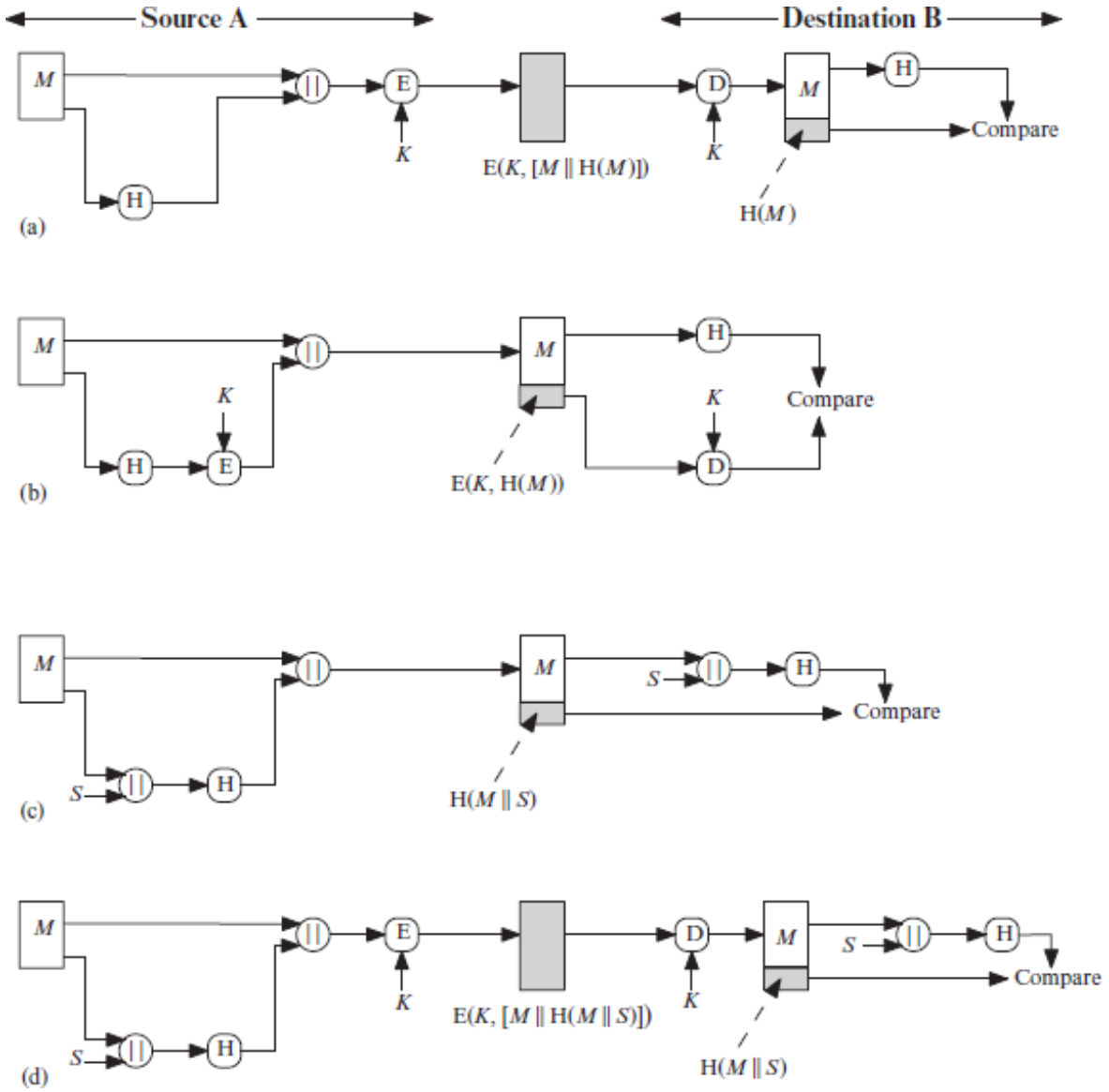


(b) Man-in-the-middle attack

Attach against Hash function

The following are a variety of ways in which a hash code can be used to provide message authentication.

a. The message plus concatenated hash code is encrypted using symmetric encryption. Because only A and B share the secret key, the message must have come from A and has not been altered. The hash codeprovides the structure or redundancy required to achieve authentication. Because encryption is applied to the entire message plus hash code, confidentiality is also provided.
b. Only the hash code is encrypted, using symmetric encryption. This reduces the processing burden for those applications that do not require confidentiality.
c. It is possible to use a hash function but no encryption for message authentication. The technique assumesthat the two communicating parties share a common secret value *S*. A computes the hash value over theconcatenation of *M* and *S* and appends the resulting hash value to *M*. Because B possesses *S*, it can re- compute the hash value to verify. Because the secret value itself is not sent, an opponent cannot modifyan intercepted message and cannot generate a false message.
d. Confidentiality can be added to the approach of method (c) by encrypting the entire message plus the hash code.

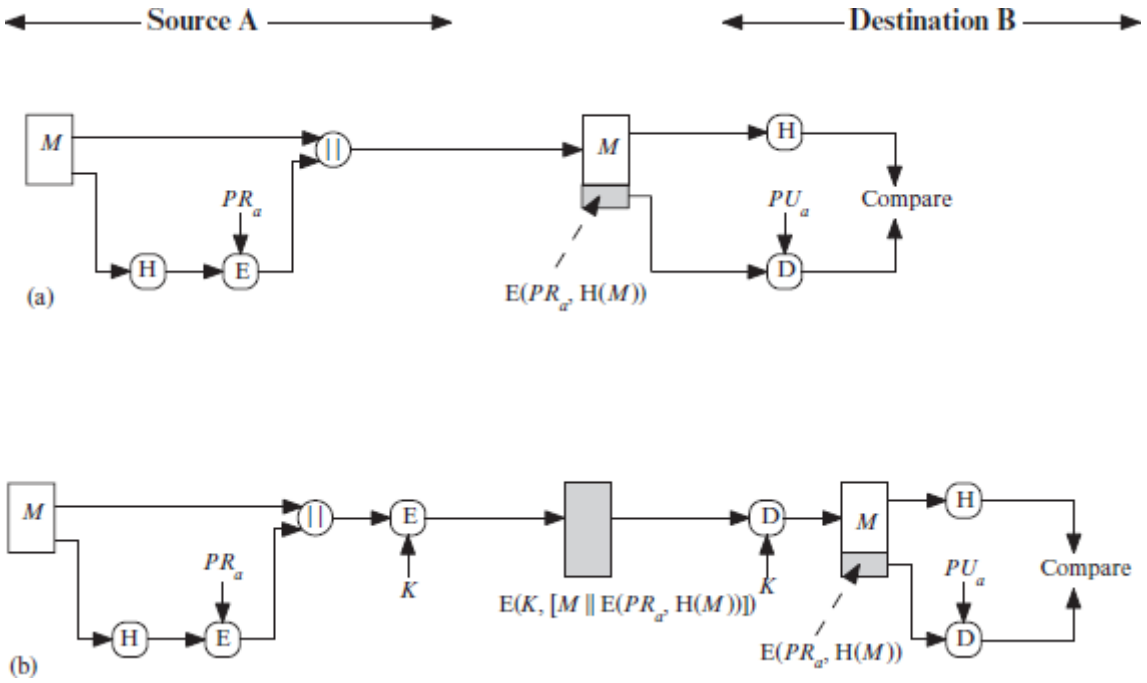


- More commonly, message authentication is achieved using a **message authentication code (MAC)**, alsoknown as a **keyed hash function**.
- Typically, MACs are used between two parties that share a secret key to authenticate informationexchanged between those parties.
- A MAC function takes as input a secret key and a data block and produces a hash value, referred to as theMAC, which is associated with the protected message.
- If the integrity of the message needs to be checked, the MAC function can be applied to the message andthe result compared with the associated MAC value.
- An attacker who alters the message will be unable to alter the associated MAC value without knowledgeof the secret key.

Digital Signatures:

- Another important application, which is similar to the message authentication application, is the **digitalsignature**.
- The operation of the digital signature is similar to that of the MAC.
- In the case of the digital signature, the hash value of a message is encrypted with a user's private key.
- Anyone who knows the user's public key can verify the integrity of the message that is associated withthe digital signature.

- In this case, an attacker who wishes to alter the message would need to know the user's private key.
- Following figures illustrates, in a simplified fashion, how a hash code is used to provide a digital signature.
  a. The hash code is encrypted, using public-key encryption with the sender's private key. As with Figure b, this provides authentication. It also provides a digital signature, because only the sendercould have produced the encrypted hash code. In fact, this is the essence of the digital signature technique.
  b. If confidentiality as well as a digital signature is desired, then the message plus the private-key- encrypted hash code can be encrypted using a symmetric secret key. This is a common technique.



Other Applications:

- Hash functions are commonly used to create a **one-way password file**.
- Hash functions can be used for **intrusion detection** and **virus detection**.
- A cryptographic hash function can be used to construct a **pseudorandom function (PRF)** or a **pseudorandom number generator (PRNG)**.

Two-Simple Hash Functions:

- To get the understanding of security considerations involved in cryptographic hash functions, we presenttwo simple, insecure hash functions in this section.
- All hash functions operate using the following general principles.
  - The input (message, file, etc.) is viewed as a sequence of *n* -bit blocks.
  - The input is processed one block at a time in an iterative fashion to produce an *n*-bit hash function.
- One of the simplest hash functions is the bit-by-bit exclusive-OR (XOR) of every block. This can beexpressed as:

$C_i = b_{i1} \oplus b_{i2} \oplus \ldots \oplus b_{im}$

where

$C_i = i^{th}$ bit of the hash code, $1 \ldots i \ldots n$*m* = number of *n*-bit blocks

in the inputb_{ij} = i^{th} bit in j^{th} block
$\oplus$ = XOR operation

- This operation produces a simple parity bit for each bit position and is known as a longitudinalredundancy check.
- It is reasonably effective for random data as a data integrity check. Each *n*-bit hash value is equallylikely.
- Thus, the probability that a data error will result in an unchanged hash value is 2-*n*.
- With more predictably formatted data, the function is less effective.
- For example, in most normal text files, the high-order bit of each octet is always zero.
- So if a 128-bit hash value is used, instead of an effectiveness of 2-128, the hash function on this type ofdata has an effectiveness of 2-112.
- A simple way to improve matters is to perform a one-bit circular shift, or rotation, on the hash valueafter each block is processed. The procedure can be summarized as follows.
    1. Initially set the *n*-bit hash value to zero.
    2. Process each successive *n*-bit block of data as follows:
        a. Rotate the current hash value to the left by one bit.
        b. XOR the block into the hash value.
- This has the effect of "randomizing" the input more completely and overcoming any regularities thatappear in the input.
- Although the second procedure provides a good measure of data integrity, it is virtually useless for datasecurity when an encrypted hash code is used with a plaintext message.
- Although a simple XOR or rotated XOR (RXOR) is insufficient if only the hash code is encrypted, youmay still feel that such a simple function could be useful when the message together with the hash codeis encrypted

Message Authentication Requirements:

In the context of communications across a network, the following attacks can beidentified.

1. **Disclosure:** Release of message contents to any person or process not possessing
2. the appropriate cryptographic key.
3. **Traffic analysis:** Discovery of the pattern of traffic between parties. In a connection oriented application, the frequency and duration of connections could be determined. In either a connection- oriented or connectionless environment, the number and length of messages between parties could bedetermined.
4. **Masquerade:** Insertion of messages into the network from a fraudulent source. This includes the creation of messages by an opponent that are purported to come from an authorized entity. Also included are fraudulent acknowledgments of message receipt or nonreceipt by someone other than themessage recipient.
5. **Content modification:** Changes to the contents of a message, including insertion, deletion, transposition, and modification.
6. **Sequence modification:** Any modification to a sequence of messages between parties, including insertion, deletion, and reordering.
7. **Timing modification:** Delay or replay of messages. In a connection-oriented application, an entire session or sequence of messages could be a replay of some previous valid session, or individual messages in the sequence could be delayed or replayed. In a connectionless application, an individualmessage (e.g., datagram) could be delayed or replayed.
8. **Source repudiation:** Denial of transmission of message by source.
9. **Destination repudiation:** Denial of receipt of message by destination.

- Measures to deal with the first two attacks are in the realm of message confidentiality and are dealt within Encryption techniques.
- Measures to deal with items (3) through (6) in the foregoing list are generally regarded as messageauthentication.
- Mechanisms for dealing specifically with item (7) come under the heading of digital signatures.
- Generally, a digital signature technique will also counter some or all of the attacks listed under items (3)through (6). Dealing with item (8) may require a combination of the use of digital signatures and a protocol designed to counter this attack.
- In summary, message authentication is a procedure to verify that received messages come from the allegedsource and have not been altered.
- Message authentication may also verify sequencing and timeliness.
- A digital signature is an authentication technique that also includes measures to counter repudiation by the source.

Message Authentication Functions:

- Any message authentication or digital signature mechanism has two levels of functionality.
- At the lower level, there must be some sort of function that produces an authenticator: a value to be usedto authenticate a message.
- This lower-level function is then used as a primitive in a higher-level authentication protocol thatenables
- a receiver to verify the authenticity of a message.
- We are concerned with the types of functions that may be used to produce an authenticator.

These maybe grouped into three classes.

- **Hash function:** A function that maps a message of any length into a fixed-length hash value,which serves as the authenticator
- **Message encryption:** The ciphertext of the entire message serves as its authenticator
- **Message authentication code (MAC):** A function of the message and a secret key that producesa fixed-length value that serves as the authenticator

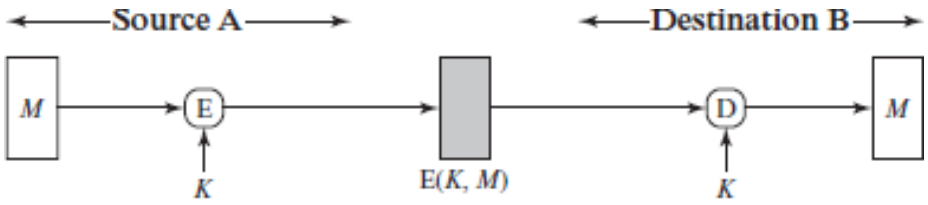
Message Encryption:

Message encryption by itself can provide a measure of authentication. The analysis differs for symmetric and public-key encryption schemes.

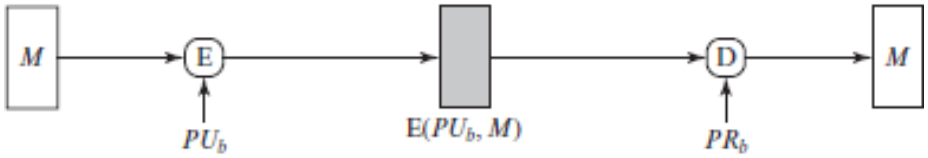
Symmetric Encryption:

Consider the straightforward use of symmetric encryption (Figure a).

- A message *M* transmitted from source A to destination B is encrypted using a secret key *K* shared by A and B. If no other party knows the key, then confidentiality is provided: No other party can recoverthe plaintext of the message.

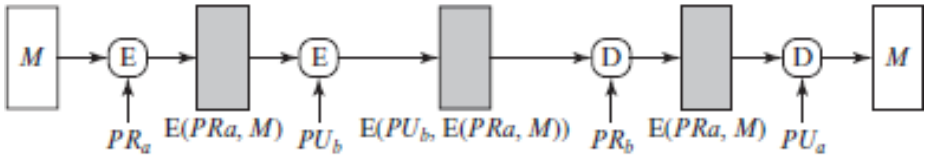


(a) Symmetric encryption: confidentiality and authentication



(b) Public-key encryption: confidentiality

(c) Public-key encryption: authentication and signature



(d) Public-key encryption: confidentiality, authentication, and signature

Public-Key Encryption:

- The straightforward use of public-key encryption (Figure b) provides confidentiality but notauthentication.
- The source (A) uses the public key PU_b of the destination (B) to encrypt *M*. Because only B has the corresponding private key PR_b, only B can decrypt the message. This scheme provides no authentication, because any opponent could also use B's public key to encrypt a message and claim tobe A.
- To provide authentication, A uses its private key to encrypt the message, and B uses A's public

key to decrypt (Figure c). This provides authentication using the same type of reasoning as in the symmetric encryption case: The message must have come from A because A is the only party that possesses PR_a and therefore the only party with the information necessary to construct ciphertext that can be decryptedwith PU_a.

- There must be some internal structure to the plaintext so that the receiver can distinguish between
- well-formed plaintext and random bits.
- Assuming there is such structure, then the scheme of Figure c does provide authentication. It alsoprovides what is known as digital signature.
- Only A could have constructed the ciphertext because only A possesses PR_a. Not even B, the recipient, could have constructed the ciphertext. Therefore, if B is in possession of the ciphertext, B has the meansto prove that the message must have come from A.
- In effect, A has "signed" the message by using its private key to encrypt.
- Note that this scheme does not provide confidentiality. Anyone in possession of A's public key candecrypt the ciphertext.
- To provide both confidentiality and authentication, A can encrypt *M* first using its private key, which provides the digital signature, and then using B's public key, which provides confidentiality (Figure d).
- The disadvantage of this approach is that the public-key algorithm, which is complex, must be exercisedfour times rather than two in each communication.

Message Authentication Code

- An alternative authentication technique involves the use of a secret key to generate a small fixed-sizeblock of data, known as a **cryptographic checksum** or MAC, that is appended to the message.
- This technique assumes that two communicating parties, say A and B, share a common secret key *K*.
- When A has a message to send to B, it calculates the MAC as a function of the message and the key:

MAC = C(*K*, *M*)
where
M = input message C = MAC function
K = shared secret key
MAC = message authentication code

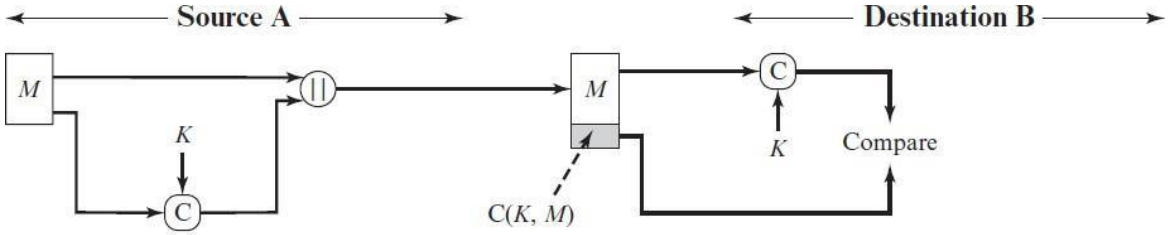
- The message plus MAC are transmitted to the intended recipient.
- The recipient performs the same calculation on the received message, using the same secret key, to generate a new MAC.
- The received MAC is compared to the calculated MAC (Figure a). If we assume that only the receiver and the sender know the identity of the secret key, and if the received MAC matches the calculated MAC,then
    1. The receiver is assured that the message has not been altered. If an attacker alters the message but does not alter the MAC, then the receiver's calculation of the MAC will differ from the received MAC. Because the attacker is assumed not to know the secret key, the attacker cannot alter the MACto correspond to the alterations in the message.

2. The receiver is assured that the message is from the alleged sender. Because no one else knows the secret key, no one else could prepare a message with a proper MAC.
3. If the message includes a sequence number (such as is used with HDLC, X.25, and TCP), then the receiver can be assured of the proper sequence because an attacker cannot successfully alter the sequence number.

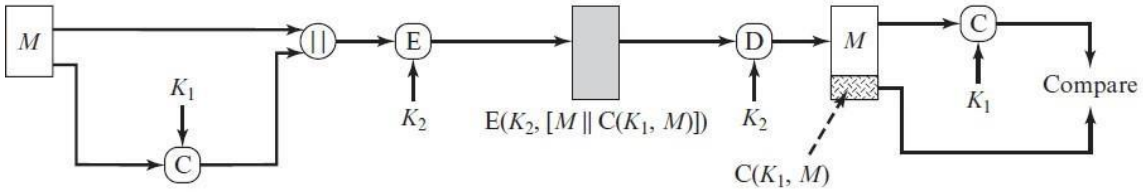
- A MAC function is similar to encryption. One difference is that the MAC algorithm need not bereversible, as it must be for decryption.
- In general, the MAC function is a many-to-one function.
- The domain of the function consists of messages of some arbitrary length, whereas the range consists ofall possible MACs and all possible keys.
- If an *n*-bit MAC is used, then there are 2*n* possible MACs, whereas there are *N* possible messages with

N >> 2*n*. Furthermore, with a *k*-bit key, there are 2*k* possible keys.

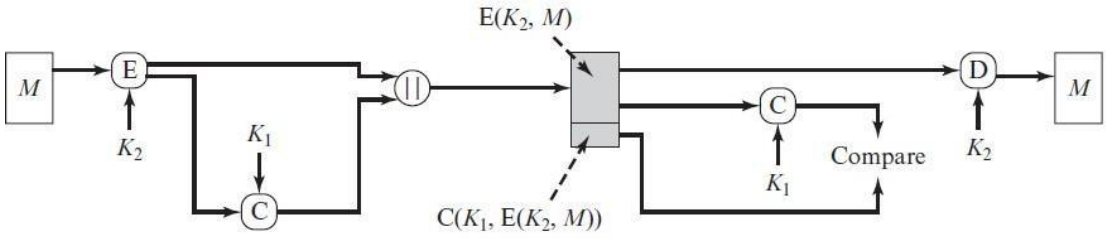
- For example, suppose that we are using 100-bit messages and a 10-bit MAC. Then, there are a total of 2100 different messages but only 210 different MACs. So, on average, each MAC value is generated bya total of 2100/210 = 290 different messages. If a 5-bit key is used, then there are 25 = 32 different mappings from the set of messages to the set of MAC values.



(a) Message authentication



(b) Message authentication and confidentiality; authentication tied to plaintext



(c) Message authentication and confidentiality; authentication tied to ciphertext

- The process depicted in Figure (a) provides authentication but not confidentiality, because the messageas a whole is transmitted in the clear.
- Confidentiality can be provided by performing message encryption either after (Figure b) or before(Figure c) the MAC algorithm.
- In both these cases, two separate keys are needed, each of which is shared by the sender and the receiver.In the first case, the MAC is calculated with the message as input and is then concatenated to the message.
- The entire block is then encrypted. In the second case, the message is encrypted first.
- Then the MAC is calculated using the resulting ciphertext and is concatenated to the ciphertext

to formthe transmitted block.

- Typically, it is preferable to tie the authentication directly to the plaintext, so the method of Figure b isused.

MACs based on hash functions: HMAC

HMAC Design Objectives

RFC 2104 lists the following design objectives for HMAC.

- To use, without modifications, available hash functions. In particular, to use hash functions that performwell in software and for which code is freely and widely available.
- To allow for easy replaceability of the embedded hash function in case faster or more secure hashfunctions are found or required.
- To preserve the original performance of the hash function without incurring a significant degradation.
- To use and handle keys in a simple way.
- To have a well understood cryptographic analysis of the strength of the authentication mechanism basedon reasonable assumptions about the embedded hash function.

HMAC Algorithm

H = embedded hash function (e.g., MD5, SHA-1, RIPEMD-160)
IV = initial value input to hash function
M = message input to HMAC (including the padding specified in the embeddedhash function)
Yi _ *i* th block of *M*, $0 \leq i \leq (L - 1)$
L _ number of blocks in *M*
b _ number of bits in a block
n _ length of hash code produced by embedded hash function
K _ secret key; recommended length is $\geq n$; if key length is greater than *b*, the key is input to the hash functionto produce an *n*-bit key
K+ _ *K* padded with zeros on the left so that the result is *b* bits
in lengthipad _ 00110110 (36 in hexadecimal) repeated *b*/8 times
opad _ 01011100 (5C in hexadecimal) repeated *b*/8 times

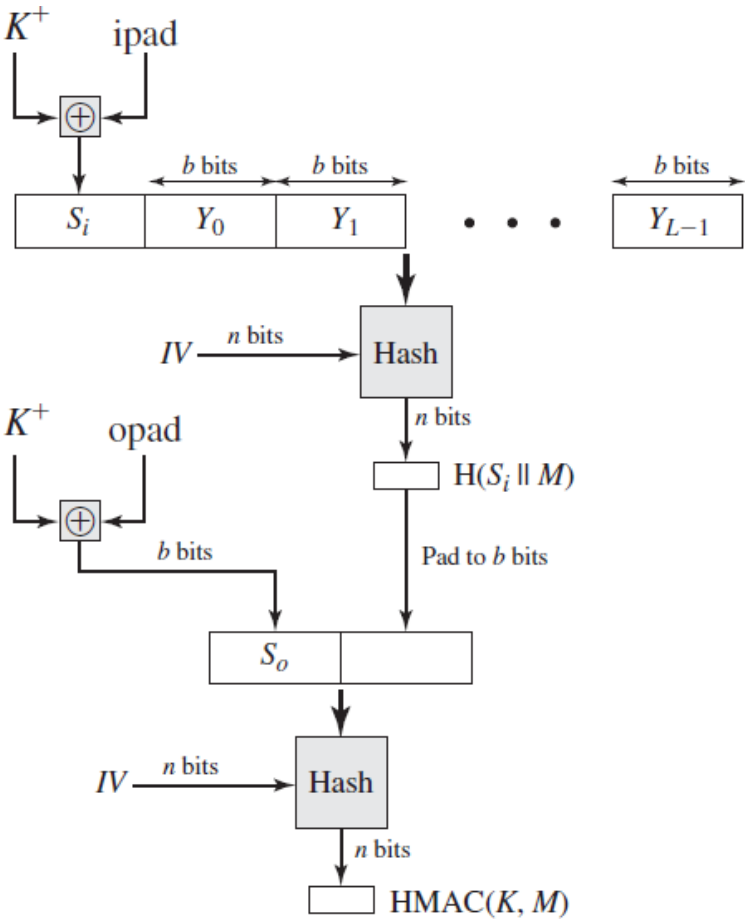


fig: HMAC Structure

HMAC can be expressed as: HMAC(K, M) = H[(K+ ⊕ opad) || H[(K+ ⊕ ipad) || M]]

The algorithm is as follows:

1. Append zeros to the left end of K to create a b-bit string K^+ (e.g., if K is oflength 160 bits and , then will be appended with 44 zeroes).
2. XOR (bitwise exclusive-OR) with ipad to produce the b-bit block S_i.
3. Append M to S_i.
4. Apply H to the stream generated in step 3.
5. XOR K^+ with opad to produce the b-bit block S_o
6. Append the hash result from step 4 to S_o.
   - Apply H to the stream generated in step 6 and output the result.

Conclusion :

Subject : Network Security (Elective - II)

Experiment No : 12

Name : **Roll No :**

Title : Write a program to implement digital Signature

Class : T.E **Date of Performance :** **Date of Submission :**

Aim : Digital Signatures Scheme.

A Digital Signature is an authentication mechanism that enables the creator of the message to attack a code that acts as a signature. The signature is formed by taking the hash of the message and encrypting the message with the creator's private key. The signature guarantees the source and integrity of the message.

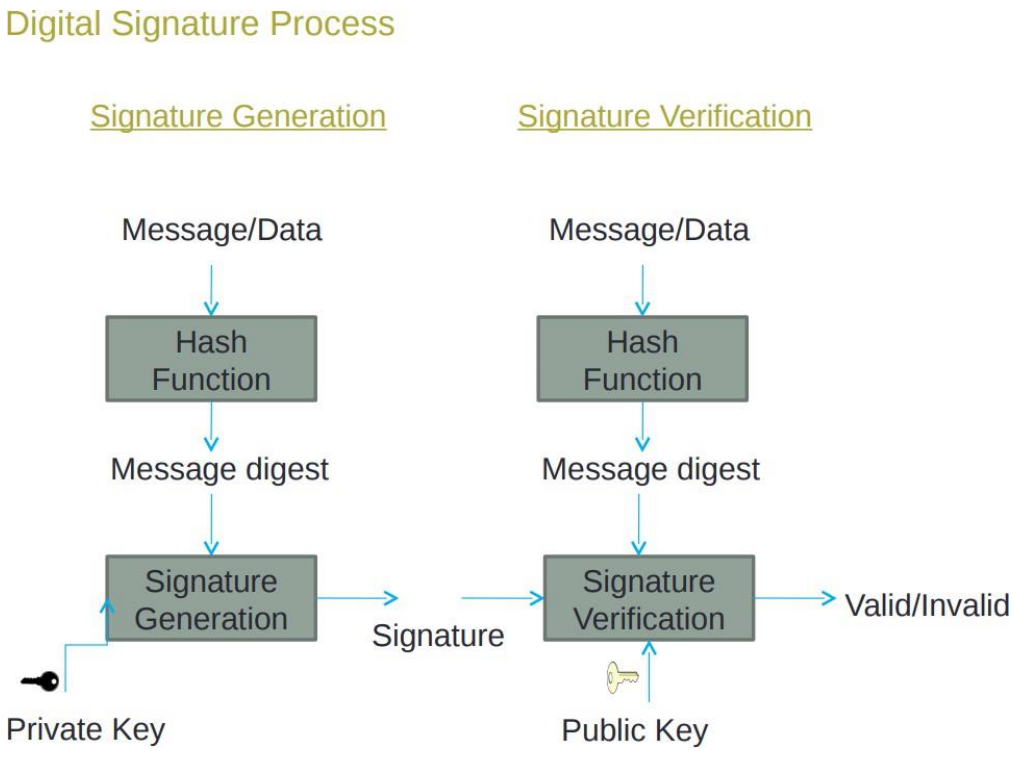
About the experiment:

In Public key setting, it becomes difficult to verify for a receiver whether message is originated from claimed source.

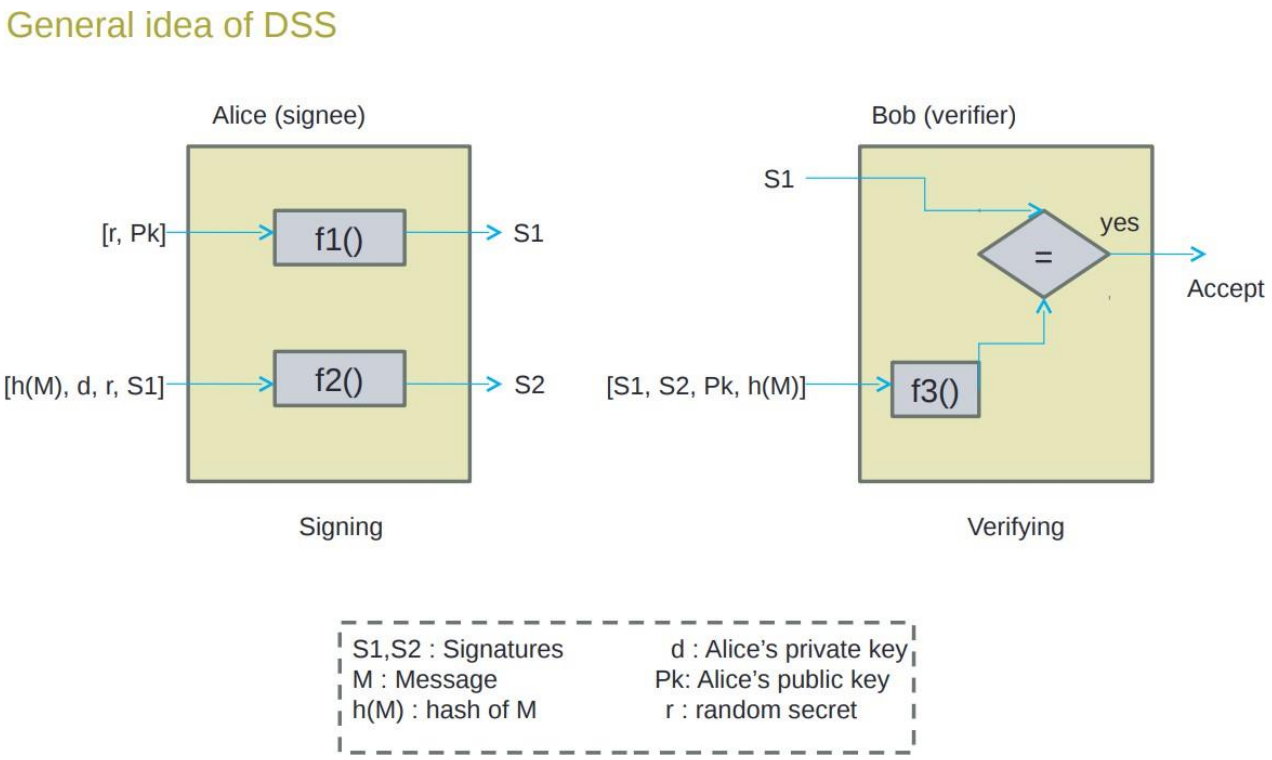
In this experiment, we show how can a receiver verify integrity of the message in public key setting.

Theory :

Digital Signatures :



Digital Signature Standard :

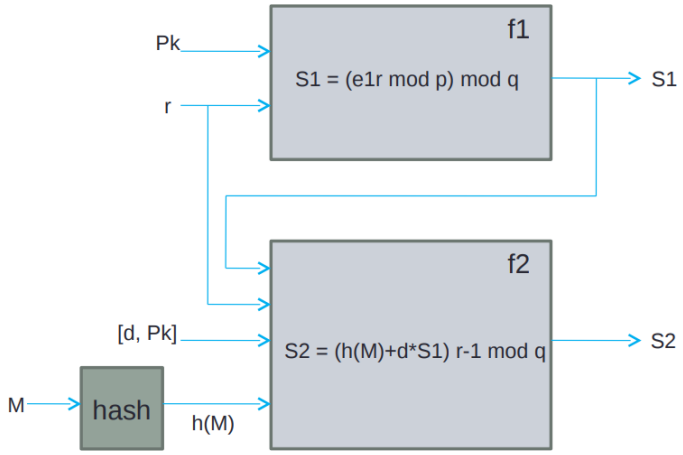


Key Generation :

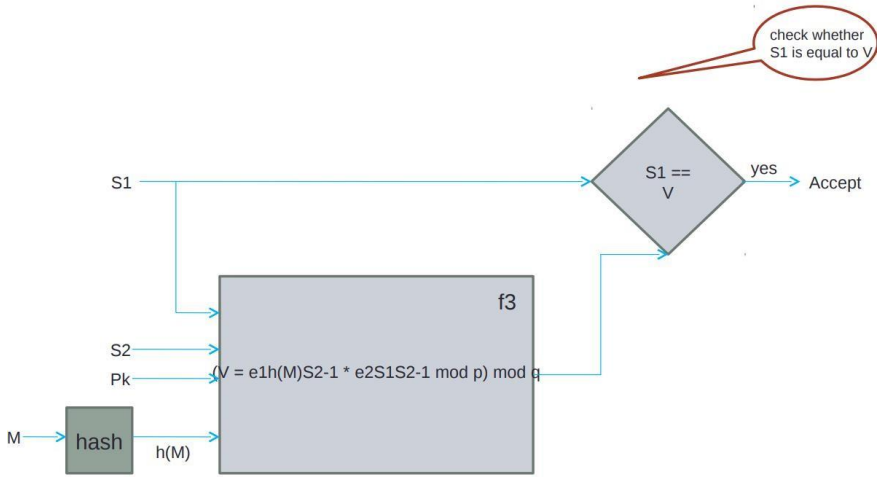
Before signing a message to any entity, Alice(the signee) must generate keys and announce the public keys to the public

Choose a prime p, between 512 and 1024 bits in length. The number of bits in p must be a multiple of 64
Choose a 160-bit prime q in such a way that q divides (p-1)
Choose a primitive root e0 in Zp
Create e1 such that : e1 = e0(p-1)/q mod p
Chose d as private key and calculate e2 = e1d
Alice's public key Pk is (e1, e2, p, q); Private key is d

Signing



Verifying :



Procedure :

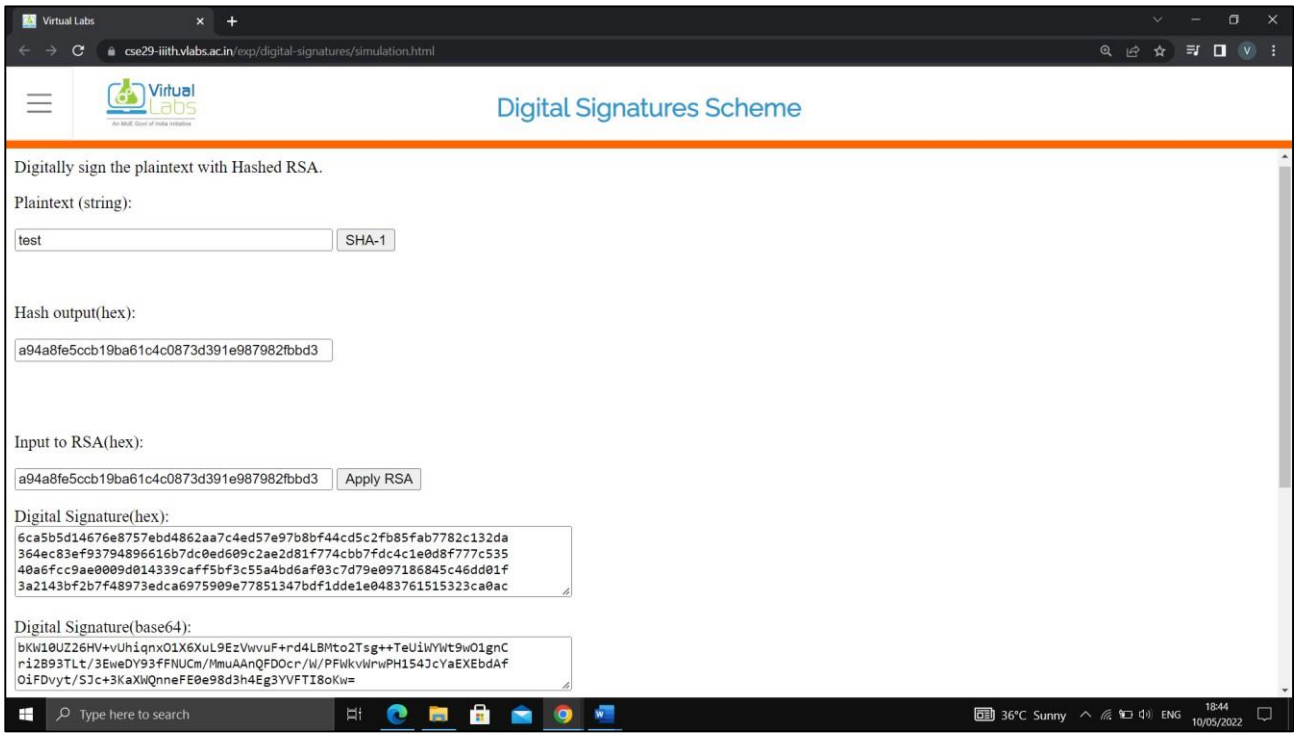
Step 1 : Enter the input text to be encrypted in the 'Plaintext' area and generate hash value for message by clicking on the SHA-1 button

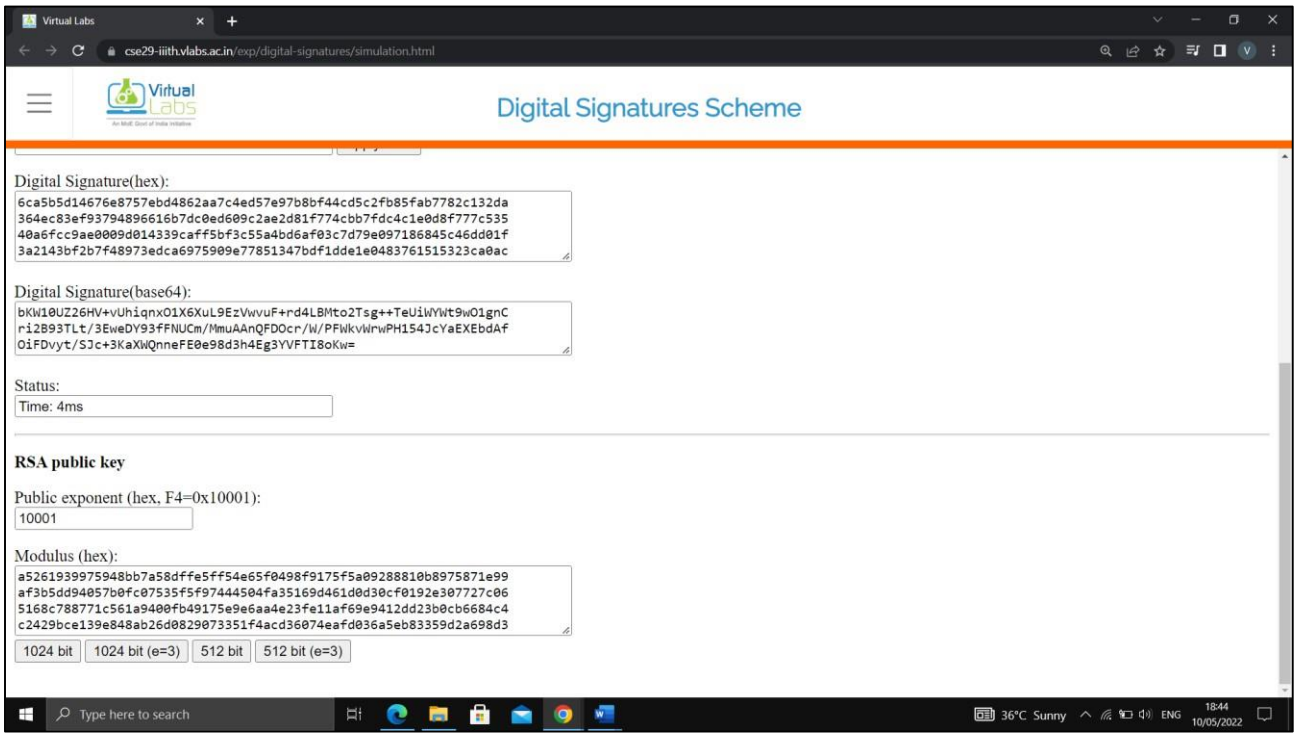
Step 2 : Copy content of Hash Output(hex) field and paste it in Input to RSA(hex) field.

Step 3 : Select keysize of public key from RSA Public key section by clicking on any key button.

Step 4 : Click on Apply RSA button to generate a digital signature.

Simulation :




Virtual Labs
cse29-iiith.vlabs.ac.in/exp/digital-signatures/simulation.html
Digital Signatures Scheme
Digital Signature(hex):
6ca5b5d14676e8757ebd4862aa7c4ed57e97b8bf44cd5c2fb85fab7782c132da
364ec83ef937948966l6b7dc0ed609c2ae2d81f774cbb7fdc4c1e0d8f777c535
40a6fcc9ae0009d014339caff5bf3c55a4bd6af03c7d79e097186845c46dd01f
3a2143bf2b7f48973edca6975909e77851347bdf1dde1e0483761515323ca0ac
Digital Signature(base64):
bKW10UZ26HV+vUhiqnxO1X6XuL9EzVwvuF+rd4LBMto2Tsg++TeUiWYWt9wO1gnC
ri2B93TLt/3EweDY93fFNUCm/MmuAAnQFDOcr/W/PFWkvWrwPH154JcYaEXEbdAf
OiFDvyt/SJc+3KaXWQnneFE0e98d3h4Eg3YVFTI8oKw=
Status:
Time: 4ms
RSA public key
Public exponent (hex, F4=0x10001):
10001
Modulus (hex):
a5261939975948bb7a58dffe5ff54e65f0498f9175f5a0928881Ob8975871e99
af3b5dd94057b0fc07535f5f97444504fa35169d461d0d30cf0192e307727c06
5168c788771c561a9400fb49175e9e6aa4e23fe11af69e9412dd23b0cb6684c4
c2429bce139e848ab26d0829073351f4acd36074eafd036a5eb83359d2a698d3
1024 bit
1024 bit (e=3)
512 bit
512 bit (e=3)
Type here to search
36°C Sunny
ENG
18:44
10/05/2022

Conclusion :

Content beyond Syllabus

Subject : Network Security (Elective - II)

Experiment No : 1

Name : **Roll No :**

Title : One-Time Pad and Perfect Secrecy

Class : T.E **Date of Performance :** **Date of Submission :**

Aim : Vernam Cipher and Perfect secrecy.

Here we see a perfectly secure cryptosystem which was developed by Gilbert Vernam in 1918. Vernam sees a trade off between security and convenience, and the trade-off has to do with the length of the key tape as its security depends on the fact that the key is used only once

About the experiment:

The main idea is to realize that any perfectly secure encryption scheme (and not just the One-Time Pad) requires the keyspace to be at least as large as the message space, and is therefore impractical.

Theory :

Vernam proposed a bit-wise exclusive or of the message stream with a truly random zero-one stream which was shared by sender and recipient.

```
Example:

            SENDING
            -------
            message: 0 0 1 0 1 1 0 1 0 1 1 1 ...
            pad:     1 0 0 1 1 1 0 0 1 0 1 1 ...
            XOR      --------------------------
            cipher:  1 0 1 1 0 0 0 1 1 1 0 0 ...

            RECEIVING
            ---------
            cipher:  1 0 1 1 0 0 0 1 1 1 0 0 ...
            pad:     1 0 0 1 1 1 0 0 1 0 1 1 ...
            XOR      --------------------------
            message: 0 0 1 0 1 1 0 1 0 1 1 1 ...
```

This cipher is unbreakable in a very strong sense. The intuition is that any message can be transformed into any cipher (of the same length) by a pad, and all transformations are equally likely. Given a two letter message, there is a pad which adds to the message to give OK, and another pad which adds to the message to give NO. Since either of these pads are equally likely, the message is equally likely to be OK or NO.

Procedure :

STEP 1 : Select a plain text and a key by clicking on the Next buttons next to the respective fields.

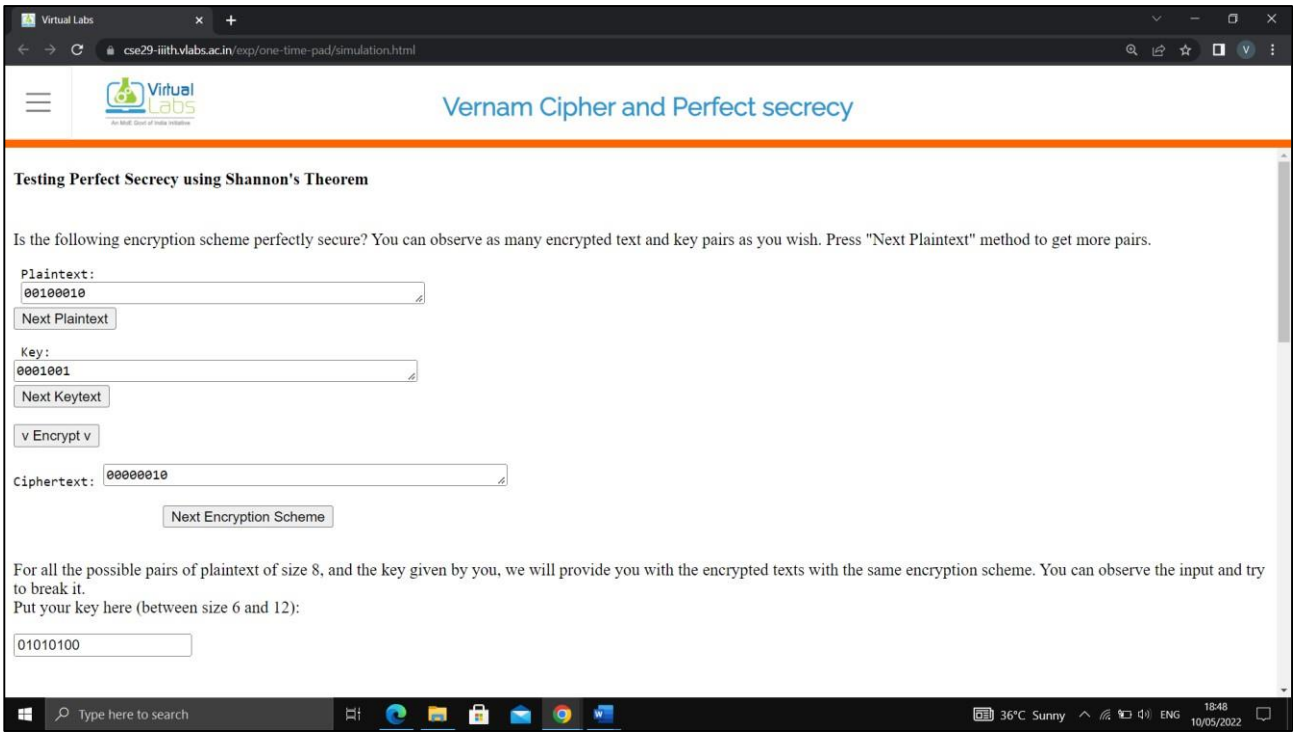
STEP 2 : If the generated key is biased, make it unbiased in 0 and 1, and use it. You can use the idea given in theory part.

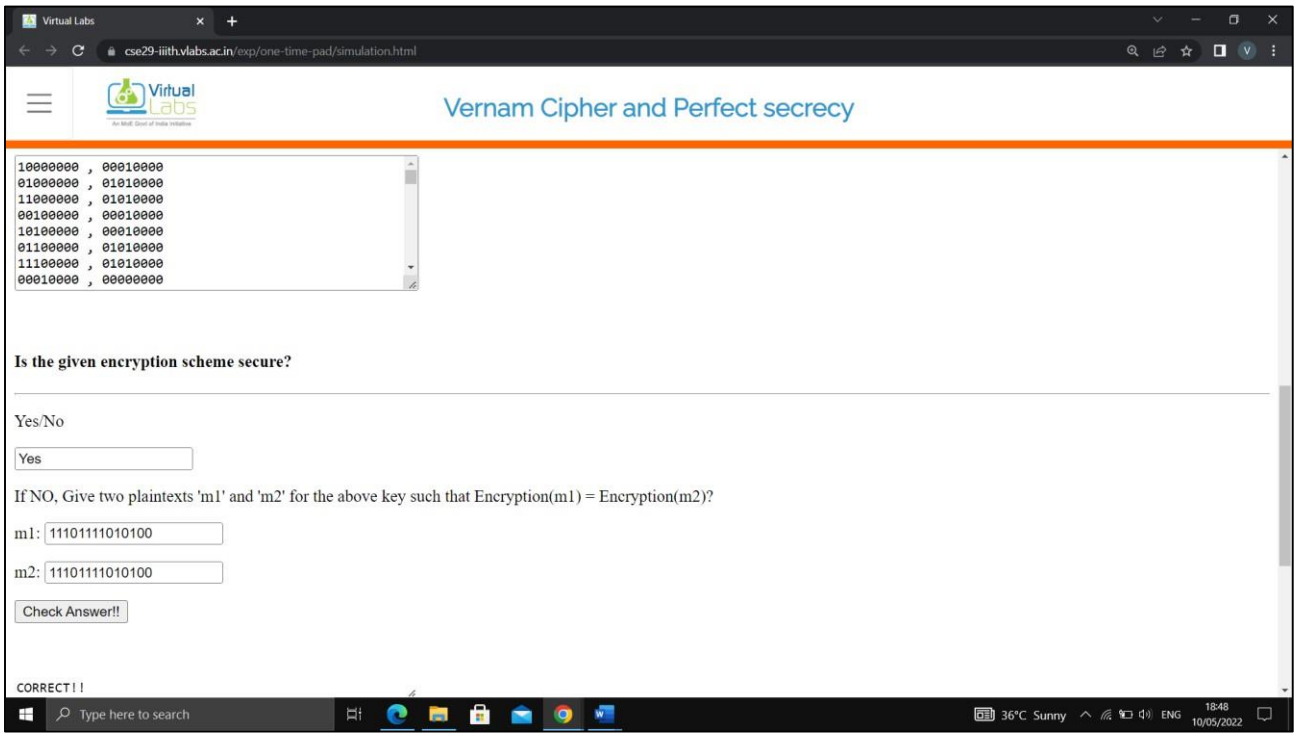
STEP 3 : On clicking on "Encrypt" button, you will get the ciphertext for the same encryption scheme. You can get the cipher text for as many plaintext and key pairs as you like for the same encryption scheme. You can change the encryption scheme too if you want.

STEP 4 : You can take a look at all the possible 2^(length of plaintext + length of key) tuples of plaintext, key and ciphertext in the next block.

STEP 5 : By observing the tuples obtained above, you need to tell if the encryption scheme being used is secure or not. If not, you need to find the message, m and ciphertext, c such that P(M=m|C=c) = P(M=m). In the experiment the size of the message space and that of the key space are equal. Thus from Shannon's theorem, to show that a scheme is not perfect, it is enough if you locate two distinct plaintexts that are encrypted to obtain the same ciphertext for the same key

Simulation :




Virtual Labs
cse29-iiith.vlabs.ac.in/exp/one-time-pad/simulation.html
Vernam Cipher and Perfect secrecy
10000000 , 00010000
01000000 , 01010000
11000000 , 01010000
00100000 , 00010000
10100000 , 00010000
01100000 , 01010000
11100000 , 01010000
00010000 , 00000000
Is the given encryption scheme secure?
Yes/No
Yes
If NO, Give two plaintexts 'm1' and 'm2' for the above key such that Encryption(m1) = Encryption(m2)?
m1: 11101111010100
m2: 11101111010100
Check Answer!!
CORRECT!!
Type here to search
36°C Sunny
ENG
18:48
10/05/2022

Conclusion :

Subject : Network Security (Elective - II)

Experiment No : 2

Name : **Roll No :**

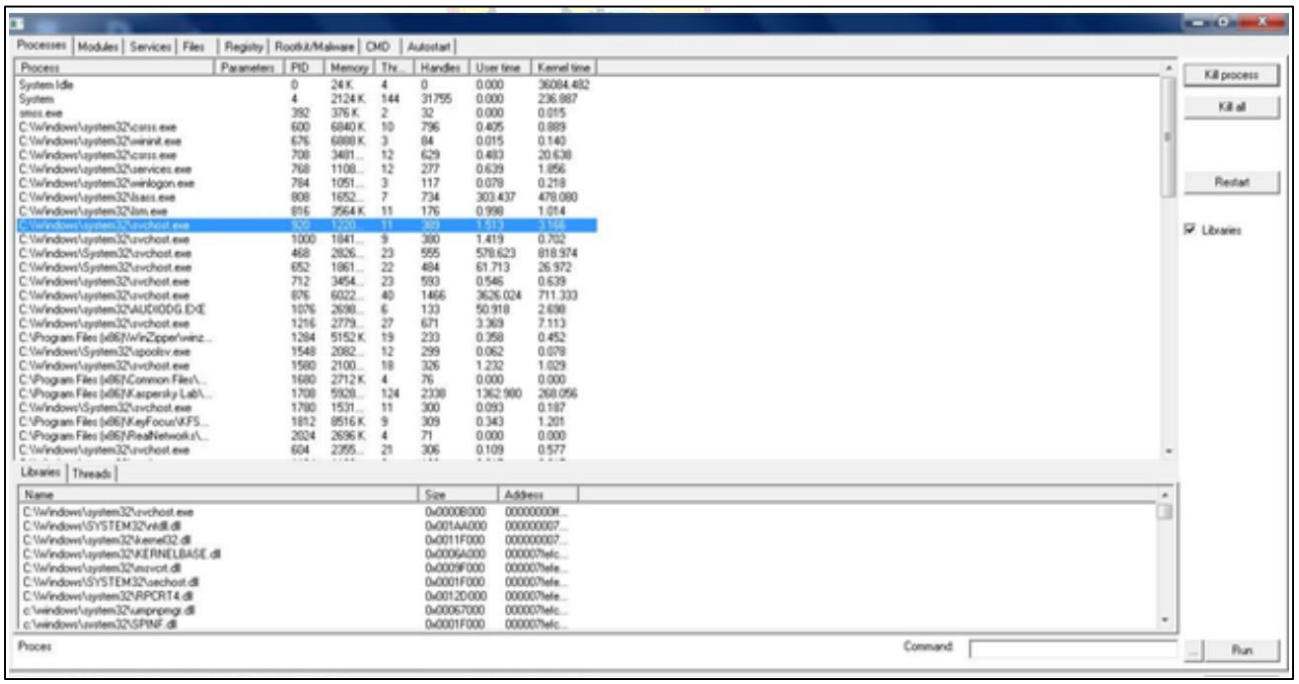
Title : Defeating Malware - Rootkit Hunter

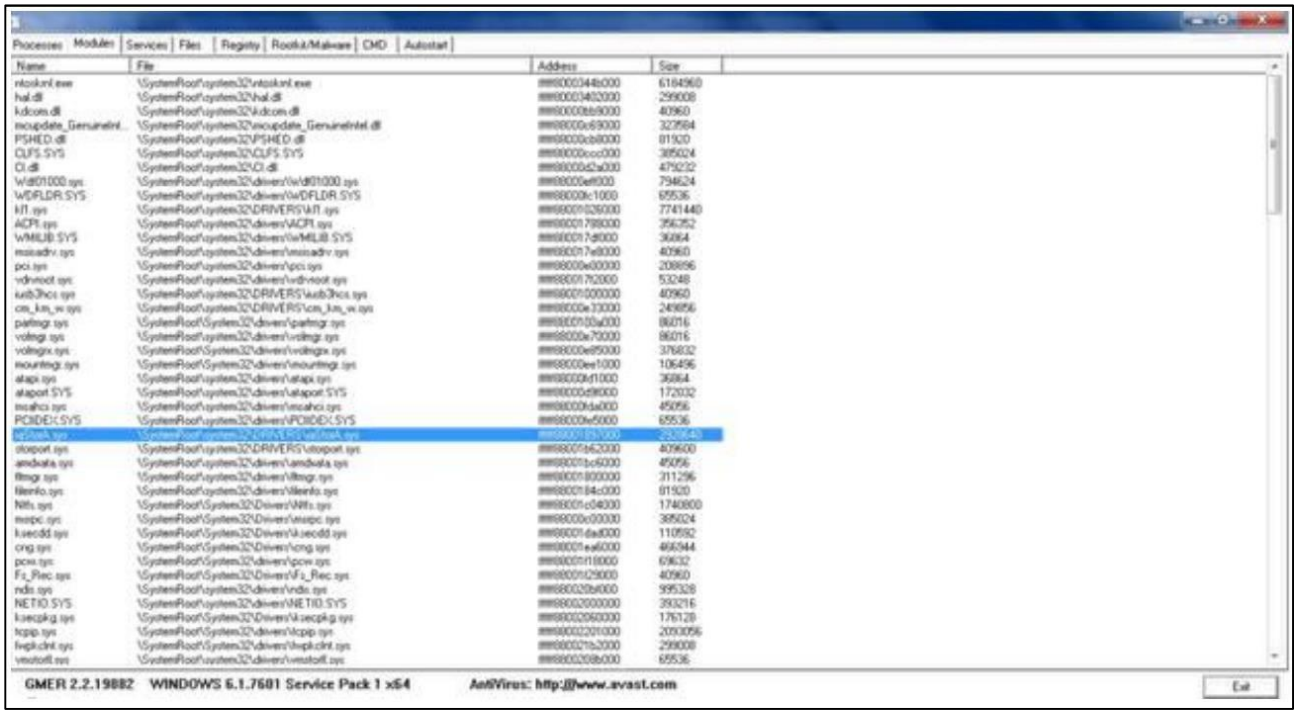
Class : T.E **Date of Performance :** **Date of Submission :**

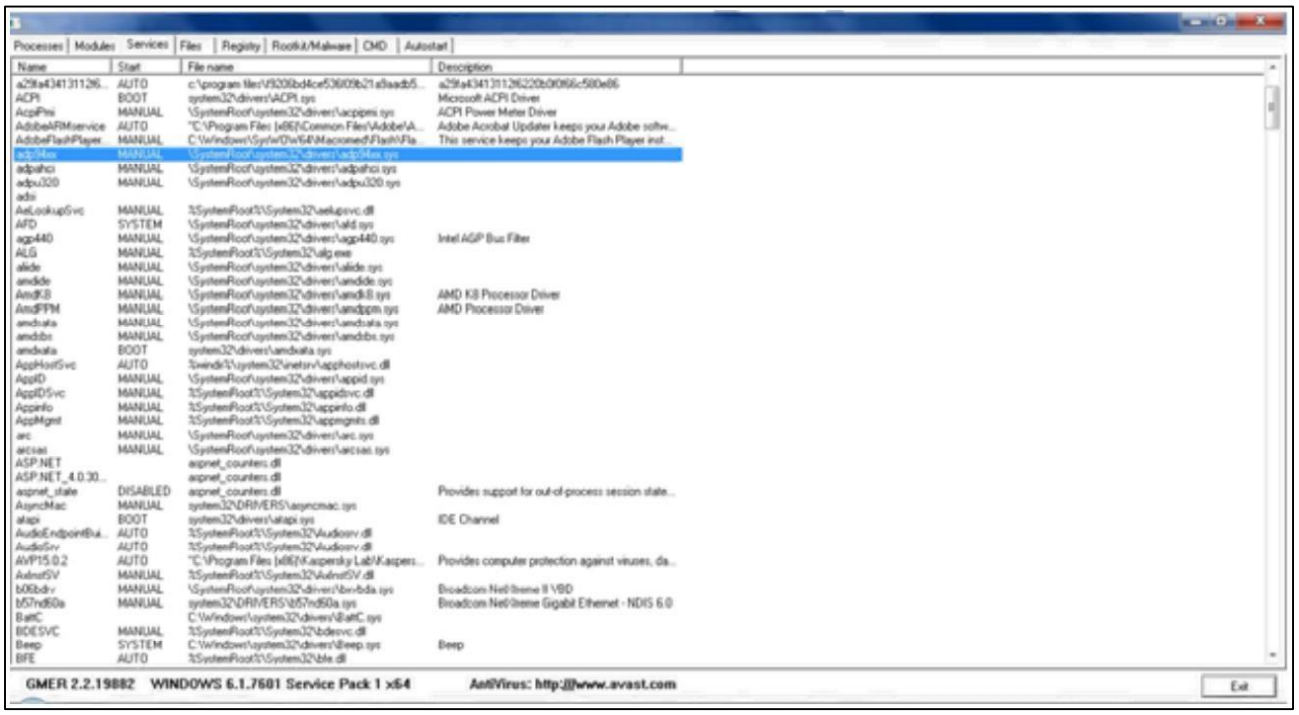
Aim : Root kit is a stealth type of malicious software designed to hide the existence of certain process from normal methods of detection and enables continued privileged access to a computer.

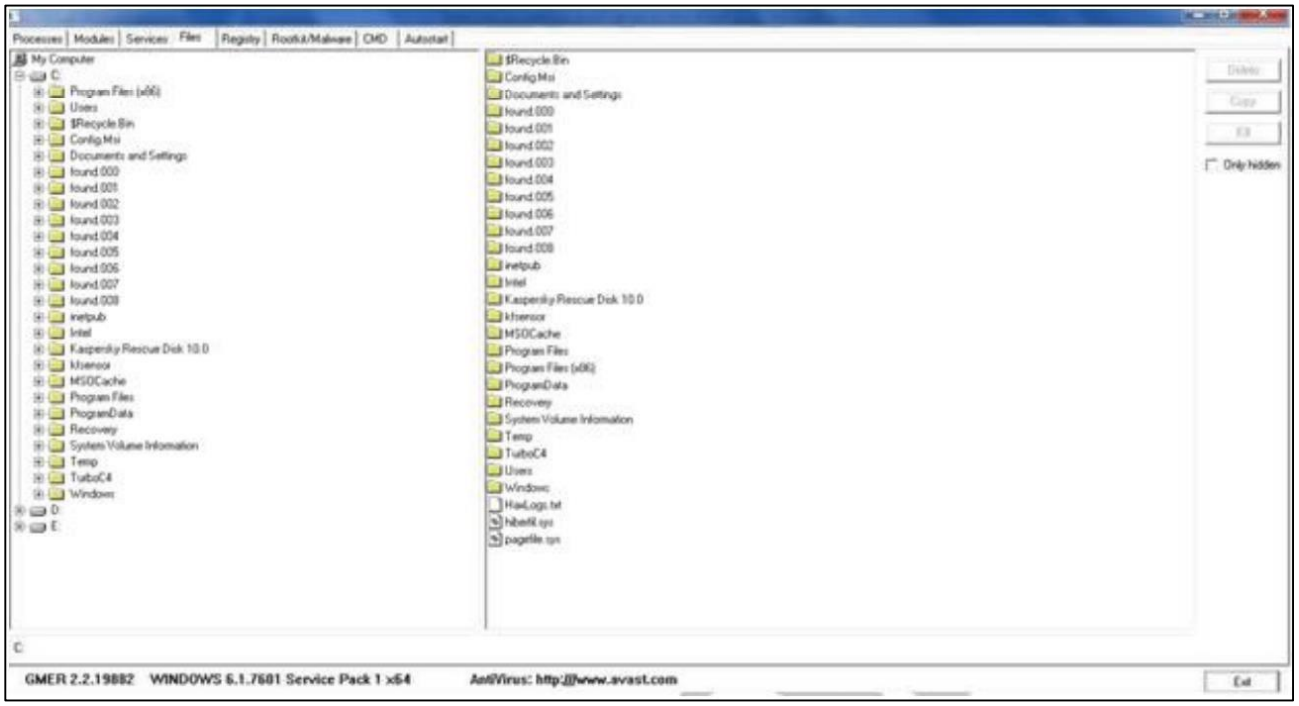
Procedure :

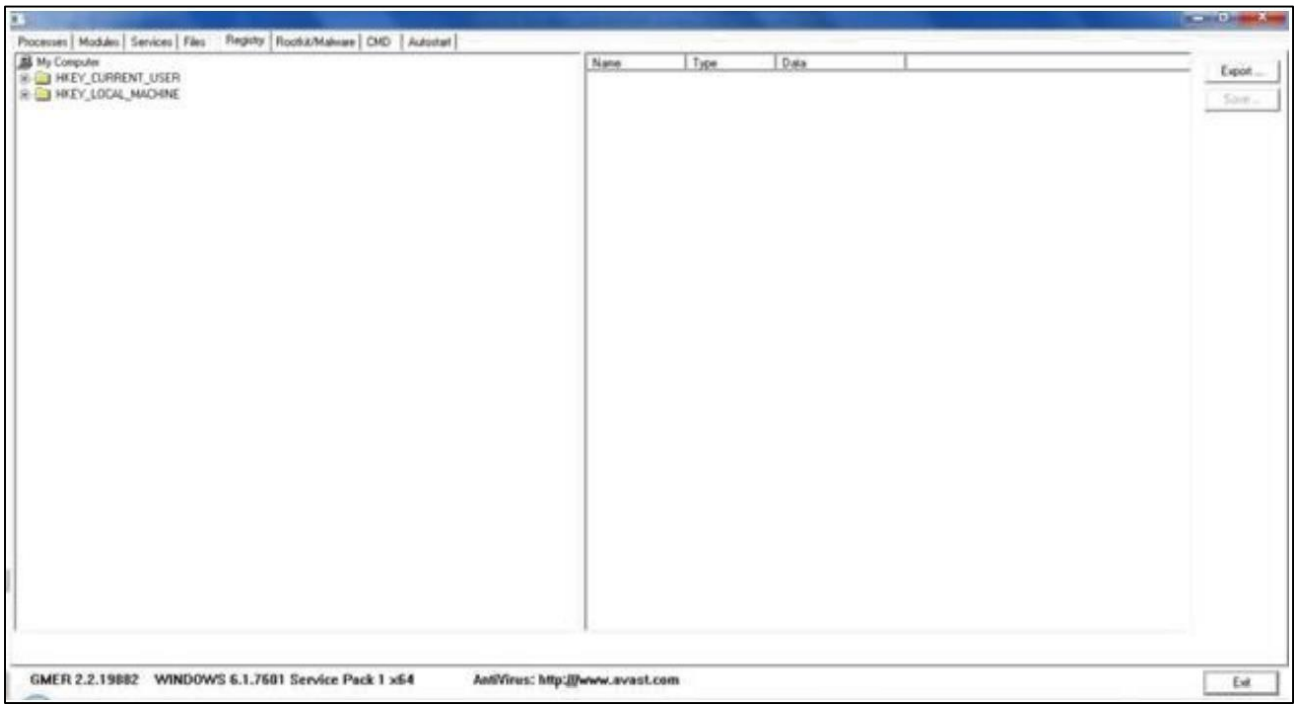
- Download Rootkit Tool from GMER website. www.gmer.net
- This displays the Processes, Modules, Services, Files, Registry, RootKit/Malwares, Autostart, CMD of local host.
- Select Processes menu and kill any unwanted process if any. Modules menu displays the various system files like .sys, .dll
- Services menu displays the complete services running with Autostart, Enable, Disable, System, Boot.
- Files menu displays full files on Hard-Disk volumes.
- Registry displays Hkey_Current_user and Hkey_Local_Machine. Rootkits/Malawares scans the local drives selected.
- Autostart displays the registry base Autostart applications.
- CMD allows the user to interact with command line utilities or Registry

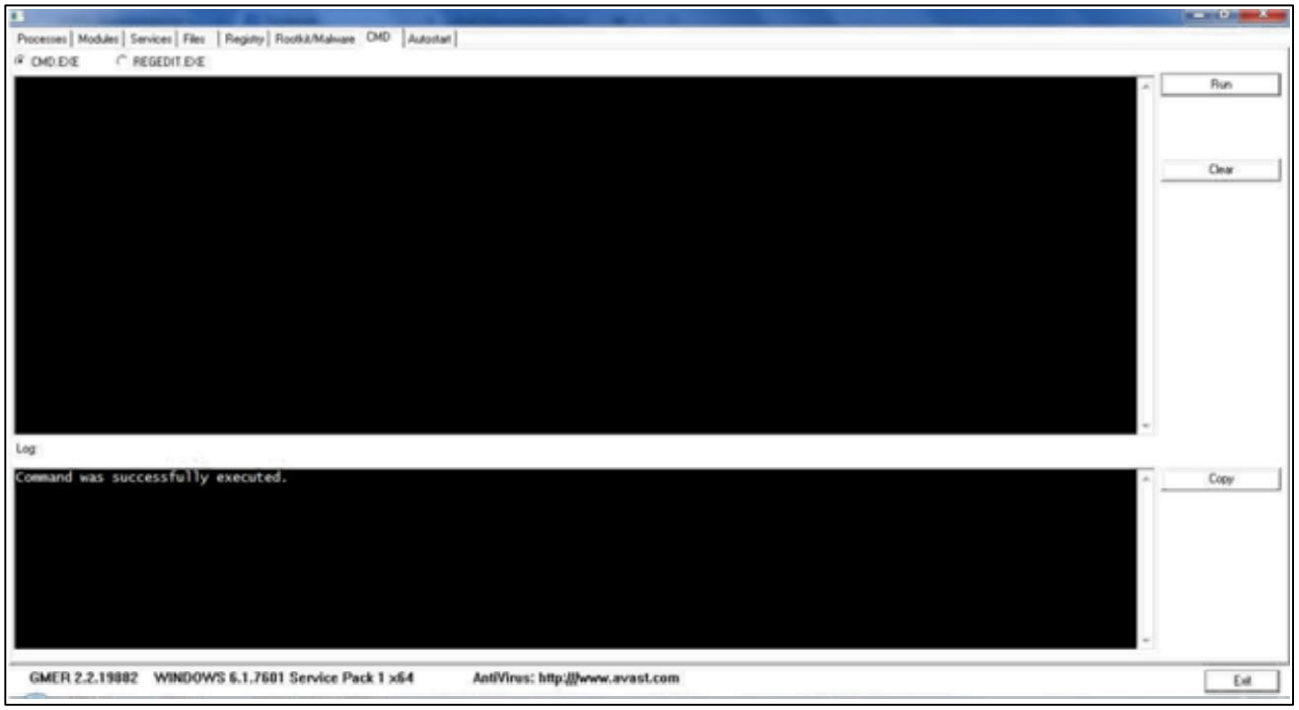

Processes
Modules
Services
Files
Registry
Rootkit/Malware
CMD
Autostart
Kill process
Kill all
Restart
Libraries
Threads
Run

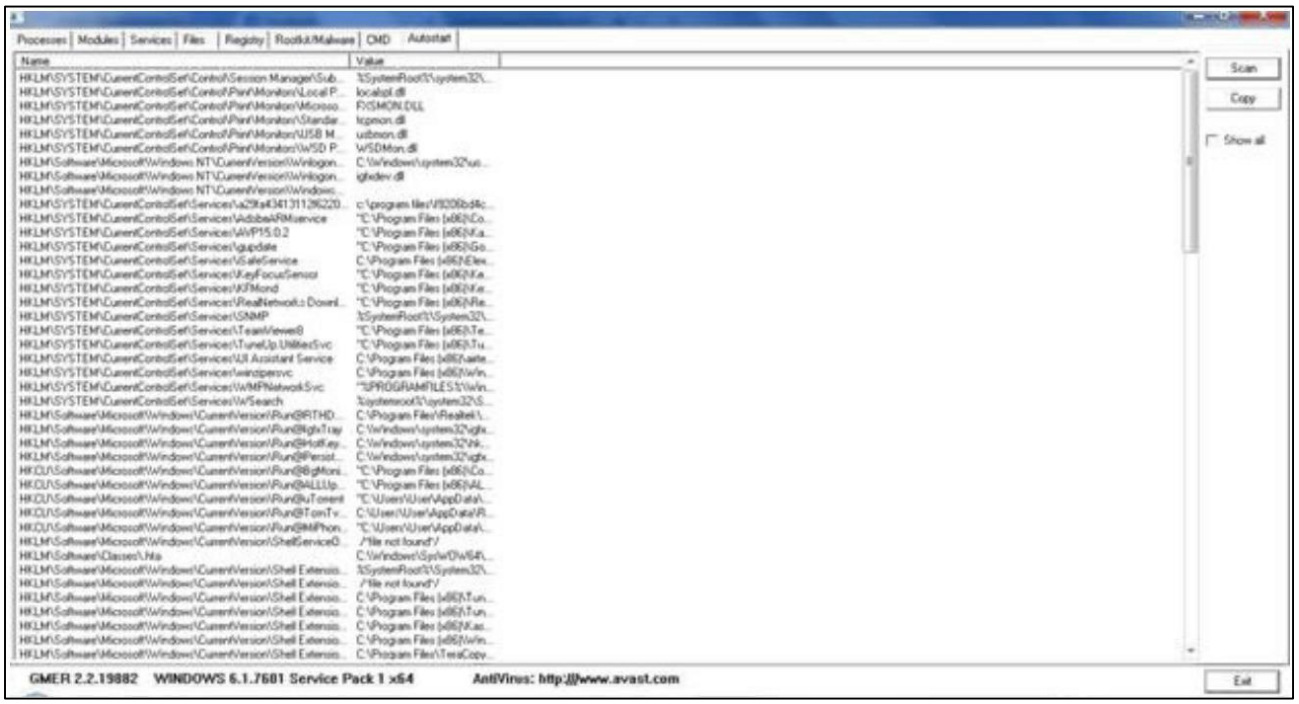

Processes
Modules
Services
Files
Registry
Rootkit/Malware
CMD
Autostart
GMER 2.2.19882 WINDOWS 6.1.7601 Service Pack 1 x64
AntiVirus: http://www.avast.com
Exit


Processes | Modules | Services | Files | Registry | RootKit/Malware | CMD | Autostart
Name | Start | File name | Description
GMER 2.2.19882 WINDOWS 6.1.7601 Service Pack 1 x64
AntiVirus: http://www.avast.com
Exit


Processes | Modules | Services | Files | Registry | RootKit/Malware | CMD | Autostart
My Computer
C:
Program Files (x86)
Users
$Recycle.Bin
Config.Msi
Documents and Settings
found.000
found.001
found.002
found.003
found.004
found.005
found.006
found.007
found.008
inetpub
Intel
Kaspersky Rescue Disk 10.0
Ksensor
MSOCache
Program Files
ProgramData
Recovery
System Volume Information
Temp
TurboC4
Windows
D:
E:
HiveLogs.txt
NtIoLog.sys
pagefile.sys
Delete
Copy
Kill
Only hidden
GMER 2.2.19882 WINDOWS 6.1.7601 Service Pack 1 x64
AntiVirus: http://www.avast.com
Exit


Processes | Modules | Services | Files | Registry | Rootkit/Malware | CMD | Autostart
My Computer
HKEY_CURRENT_USER
HKEY_LOCAL_MACHINE
Name
Type
Data
Export ...
Save ...
GMER 2.2.19882 WINDOWS 6.1.7601 Service Pack 1 x64 AntiVirus: http://www.avast.com
Exit


Processes | Modules | Services | Files | Registry | Rootkit/Malware | CMD | Autostart
CMD.EXE
REGEDIT.EXE
Run
Clear
Log
Command was successfully executed.
Copy
GMER 2.2.19882 WINDOWS 6.1.7601 Service Pack 1 x64 AntiVirus: http://www.avast.com
Exit



Result :

Thus the Rootkits tool was installed and its various options were verified successfully.