DOP: / /2023 DOS: / /2023

**Experiment No: 01**

**Title**: Breaking the Mono-alphabetic Substitution Cipher using Frequency analysis method.

**Theory**:

* **Cryptography**

Cryptography is technique of securing information and communications through use of codes so that only those people for whom the information is intended can understand it and process it. Thus, preventing unauthorized access to information. The prefix “crypt” means “hidden” and suffix graph means “writing”

**Plaintext** can refer to anything which humans can understand and/or relate to. This may be as simple as English sentences, a script, or Java code. If you can make sense of what is written, then it is in plaintext.

**Ciphertex**t, or encrypted text, is a series of randomized letters and numbers which humans cannot make any sense of. An encryption algorithm takes in a plaintext message, runs the algorithm on the plaintext, and produces a ciphertext. The ciphertext can be reversed through the process of decryption, to produce the original plaintext.

There are two basic building blocks of all encryption techniques:

* SUBSTITUTION TECHNIQUES
* TRANSPOSITION TECHNIQUES

**⚫Substitution Techniques:**

A substitution technique is one in which the letters of plaintext are replaced by other letters or by numbers or symbols. If the plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with cipher text bit patterns.

**⚫Mono-alphabetic Substitution Cipher:**

The mono-alphanumeric substitution cipher is the oldest forms of encryption algorithms according to creates each character of a plaintext message and require a substitution process to restore it with a new character in the ciphertext.

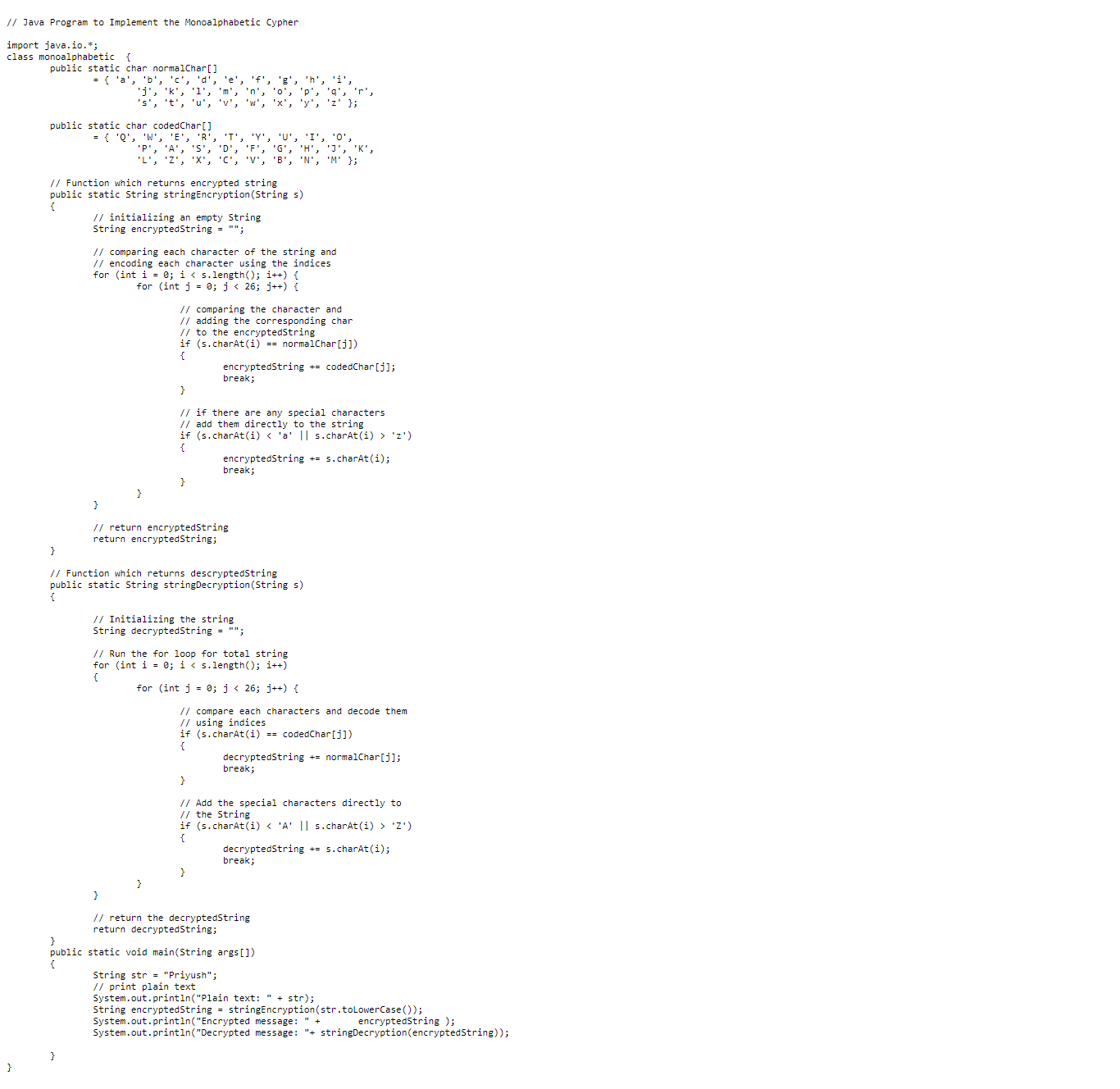
Monoalphabetic cipher is a substitution cipher in which for a given key, the cipher alphabet for each plain alphabet is fixed throughout the encryption process. For example, if ‘A’ is encrypted as ‘D’, for any number of occurrences in that plaintext, ‘A’ will always get encrypted to ‘D’.

All of the substitution ciphers we have discussed earlier in this chapter are monoalphabetic; these ciphers are highly susceptible to cryptanalyst.

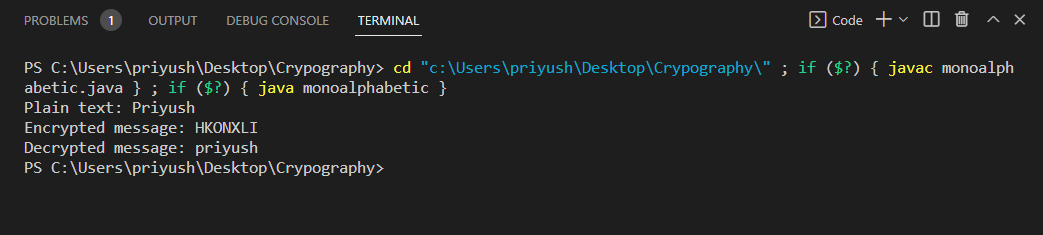
**⚫ALGORITHM**

* STEP-1: Read the plain text from the user.
* STEP-2: Read the plan text position from the user.
* STEP-3: comparing the character and adding the corresponding char to the encrypted String STEP-4: Run the for loop for total string.
* STEP-5: Display the cipher text obtained above.

**⚫Input:**

**

**⚫Output:**



**Advantages:**

The increased security possible with variant multilateral systems is the major advantage.

**Disadvantages.**

The major disadvantage is that by substituting more than one character of ciphertext for each plaintext value, the length of messages and resulting transmission times are increased.

**Conclusion: -**

Thus, we have implemented Breaking the Mono-alphabetic Substitution Cipher using Frequency analysis method.

**DOP: / /2023 DOS: / /2023**

**Experiment No: 02**

**Title**: Design and implement a product cipher using substitution ciphers

**Theory:**

**Product cipher:**

“Product cipher, data encryption scheme in which the ciphertext produced by encrypting a plaintext document is subjected to further encryption. By combining two or more simple transposition ciphers or substitution ciphers, a more secure encryption may result.”

This system used a 6 × 6 matrix to substitution-encrypt the 26 letters and 10 digits into pairs of the symbols A, D, F, G, V, and X. The resulting biliteral cipher was then written into a rectangular array and route encrypted by reading the columns in the order indicated by a key word, a.

**⚫Substitution ciphers:**

When plain text is encrypted, it becomes unreadable and is known as ciphertext. In a Substitution cipher, any character of plain text from the given fixed set of characters is substituted by some other character from the same set depending on a key. For example with a shift of 1, A would be replaced by B, B would become C, and so on.

***Algorithm for Substitution Cipher:***

Input:

* A String of both lower and upper case letters, called Plaintext.
* An Integer denoting the required key.

Procedure:

* Create a list of all the characters.
* Create a dictionary to store the substitution for all characters.
* For each character, transform the given character as per the rule, depending on whether we’re encrypting or decrypting the text.
* Print the new string generated.

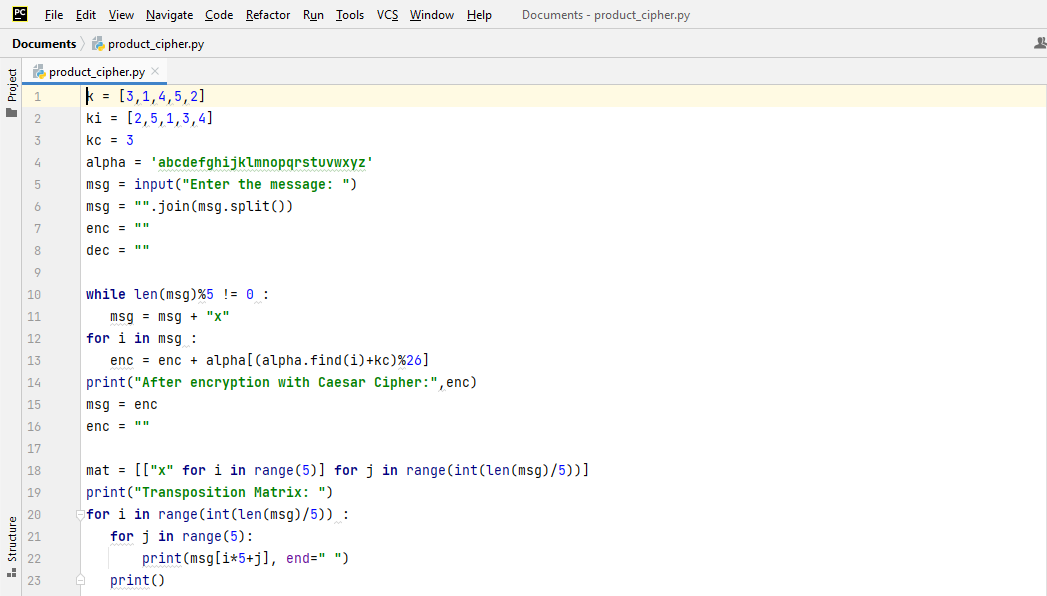
**⚫Transposition Cipher:**

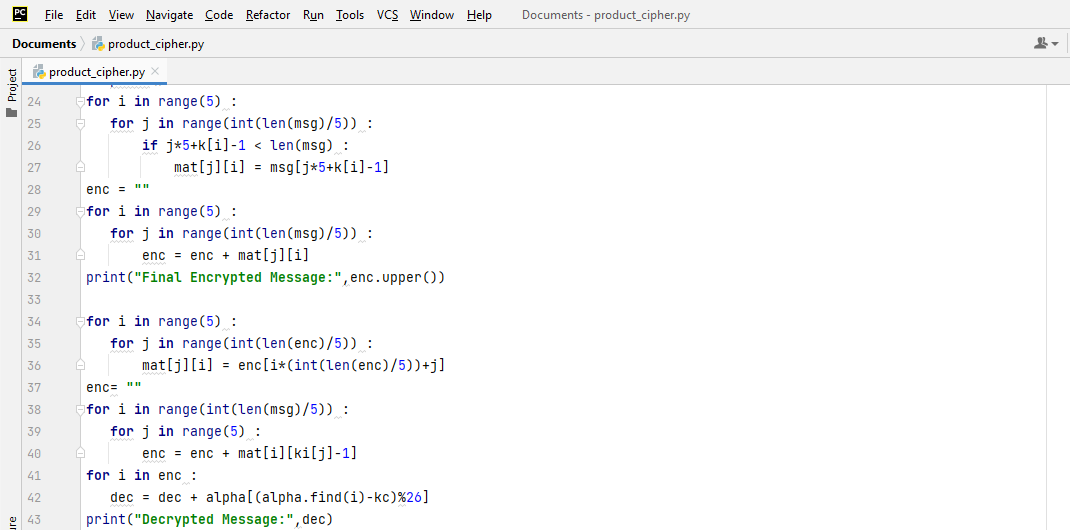
“Transposition cipher, simple data encryption scheme in which plain text characters are shifted in some regular pattern to form cipher text. In manual systems transpositions are generally carried out with the aid of an easily remembered mnemonic.”

**⚫ALGORITHM**

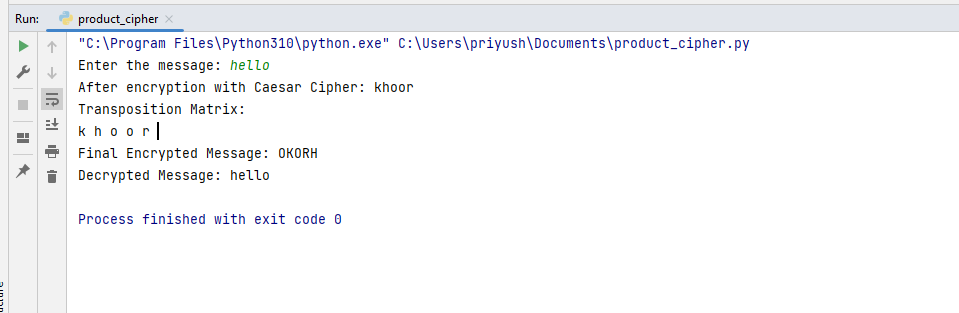
* STEP-1: Read the plain text from the user.
* STEP-2: Read the plan text position from the user.
* STEP-3: comparing the character and adding the corresponding char to the encrypted String
* STEP-4: Run the for loop for total string.
* STEP-5: Display the Transposition Matrix
* STEP-6: Display the cipher text obtained above.

**⚫Input:**





**⚫Output:**



**Conclusion: -**

Thus, we have studied, design and implement a product cipher using substitution ciphers.

**DOP: / /2023 DOS: / /2023**

**Experiment No: 03**

**Aim**: Cryptanalysis or decoding Playfair, Vigenère cipher.

**Theory:**

**🔹Vigenere Cipher:**

**Vigenere Cipher** is an encryption and decryption algorithm. It is a type of polyalphabetic substitution cipher, which means that the cipher alphabet is changed regularly during the encryption process. Due to this, the cipher becomes less vulnerable to cryptanalysis.

The Vigenere Cipher was developed in 1585 by Blaise de Vigenere. He used a Vigenere table or square to encode messages

**Encryption:**

The plaintext(P) and key(K) are added modulo 26.

Ei = (Pi + Ki) mod 26

**Decryption:**

Di = (Ei - Ki + 26) mod 26

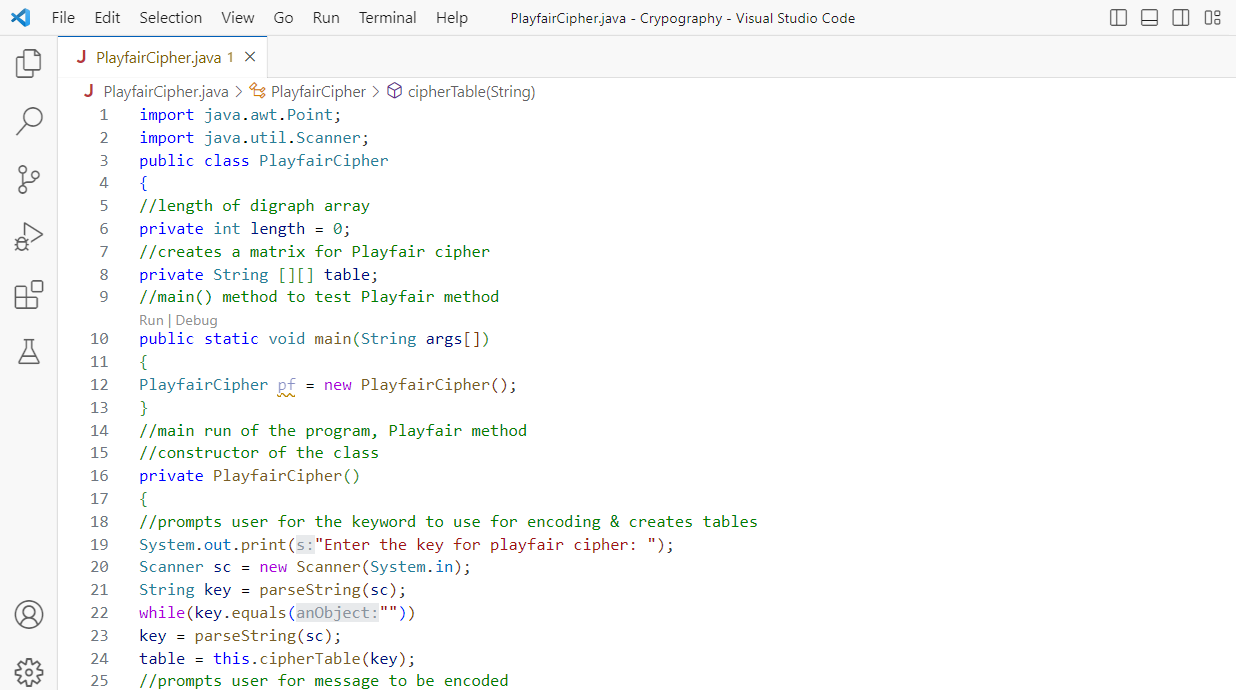
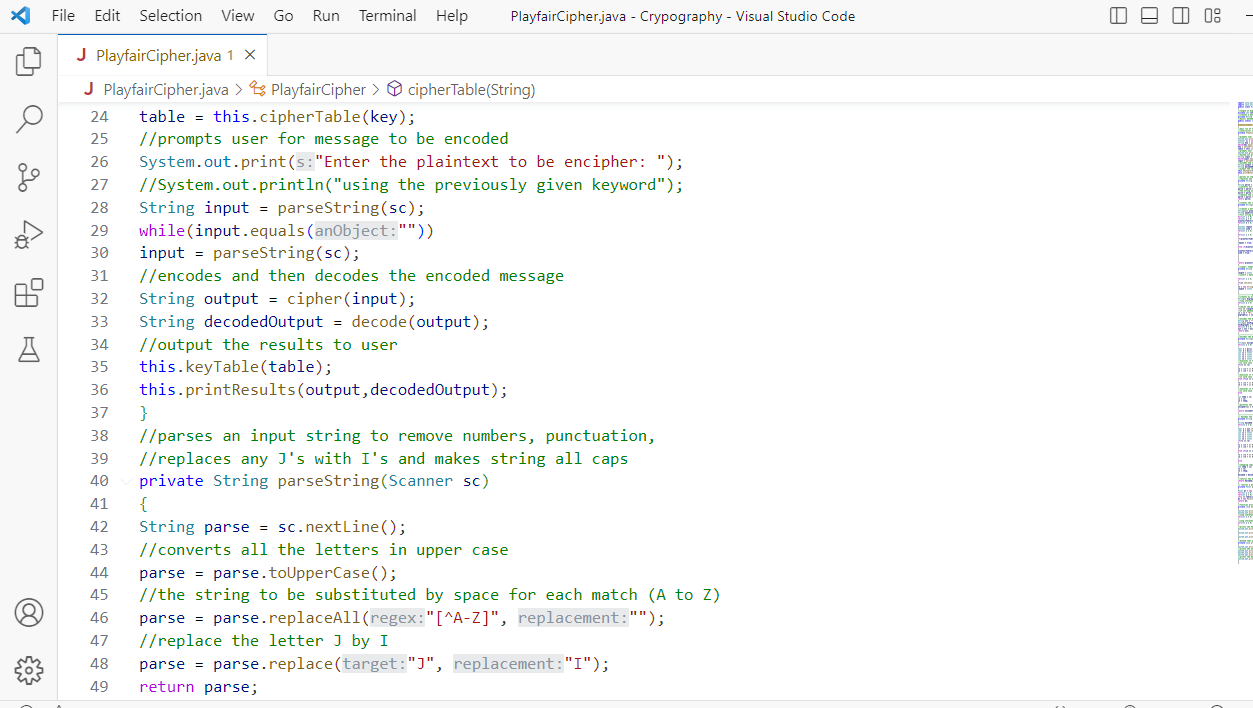
**🔹Playfair Cipher:**

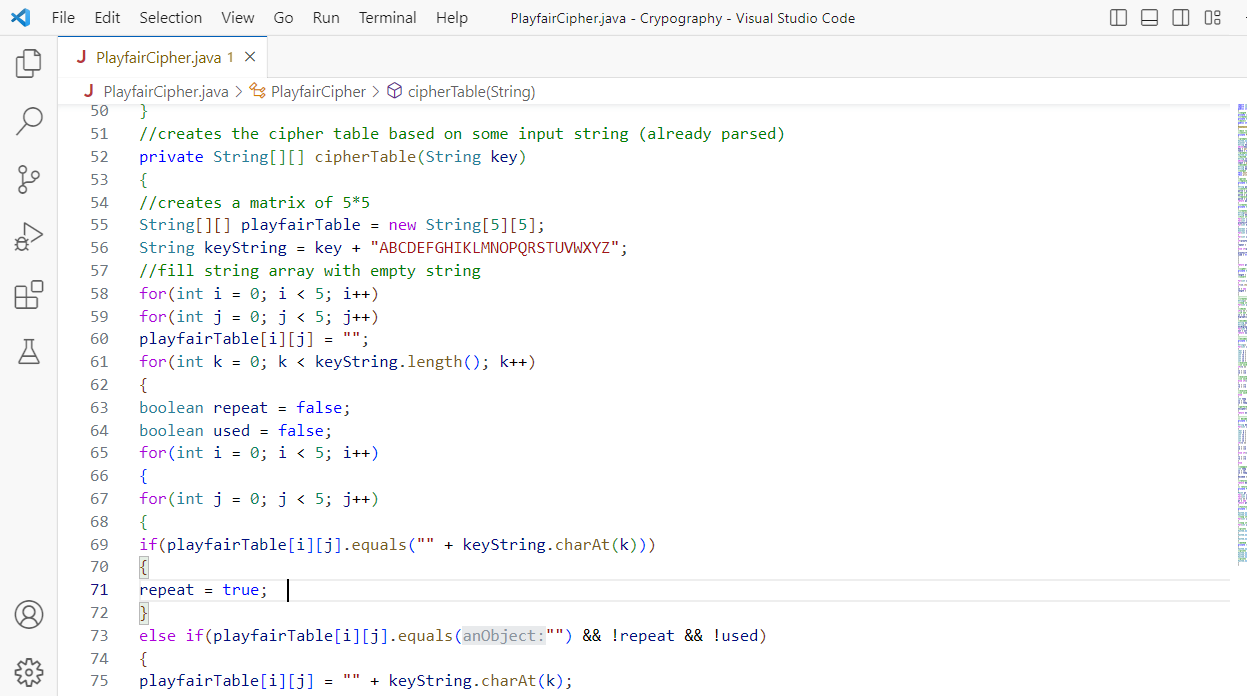
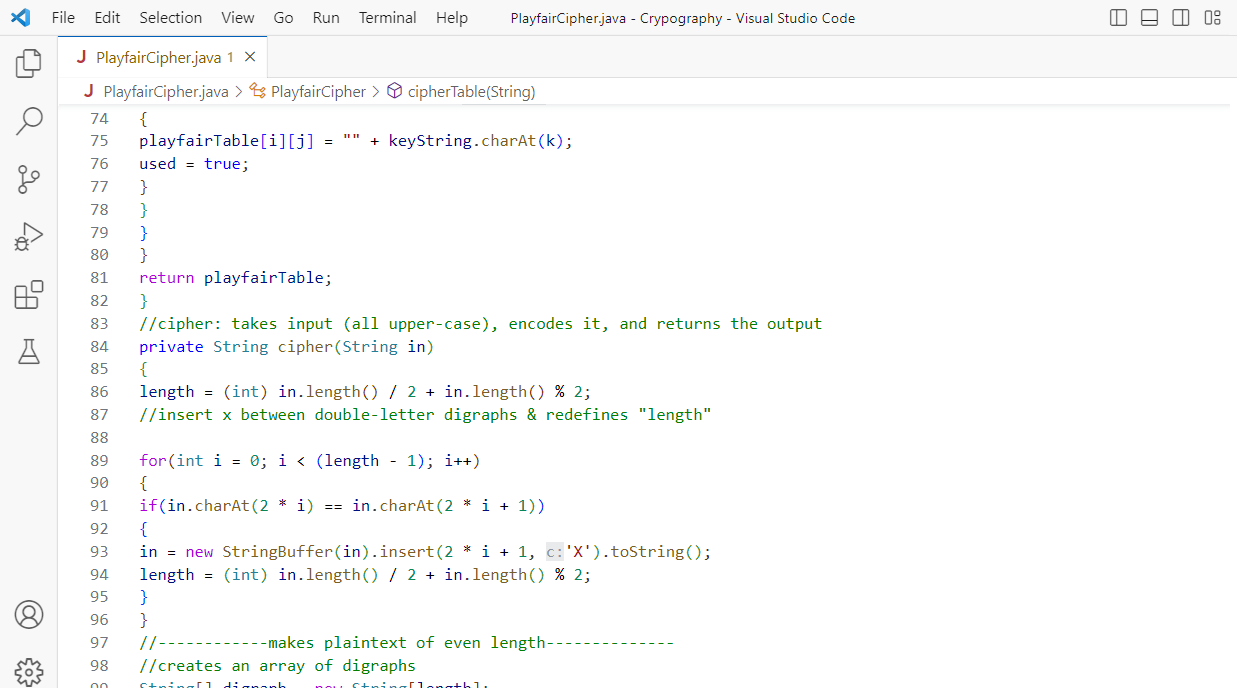
Playfair cipher is an encryption algorithm to encrypt or encode a message. It is the same as a traditional cipher. The only difference is that it encrypts a digraph (a pair of two letters) instead of a single letter.

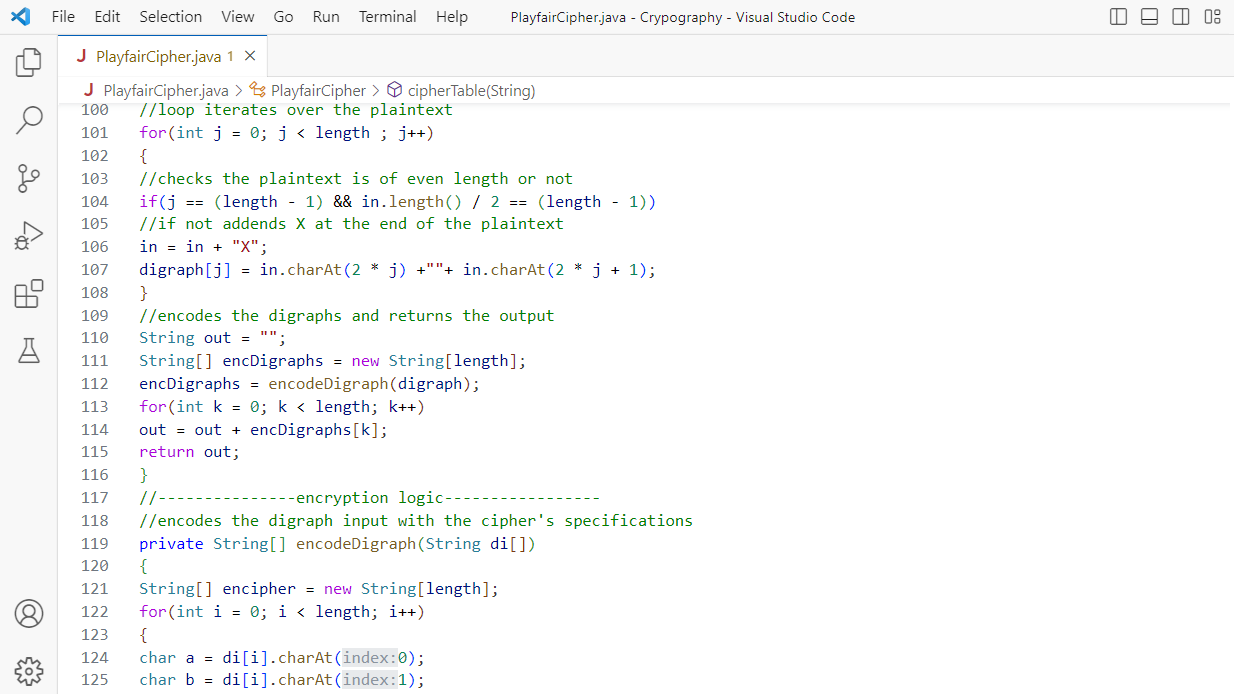
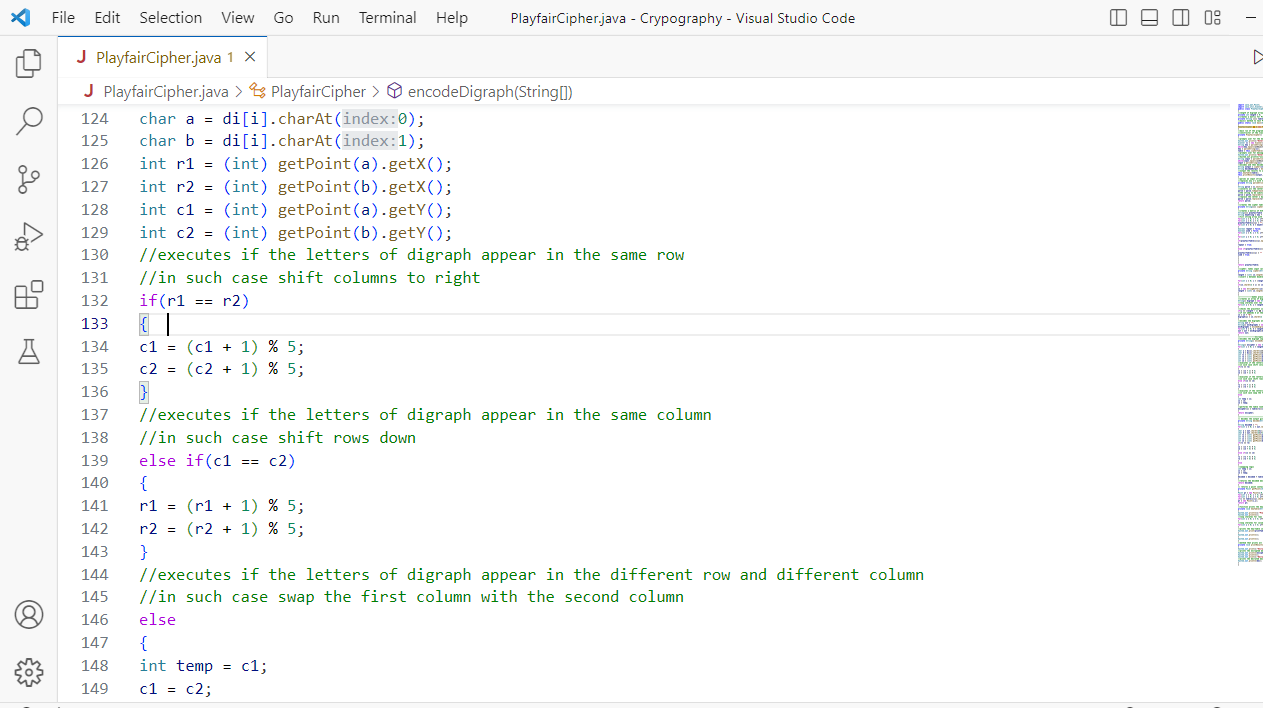
It initially creates a key-table of 5\*5 matrix. The matrix contains alphabets that act as the key for encryption of the plaintext. Note that any alphabet should not be repeated. Another point to note that there are 26 alphabets and we have only 25 blocks to put a letter inside it. Therefore, one letter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

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

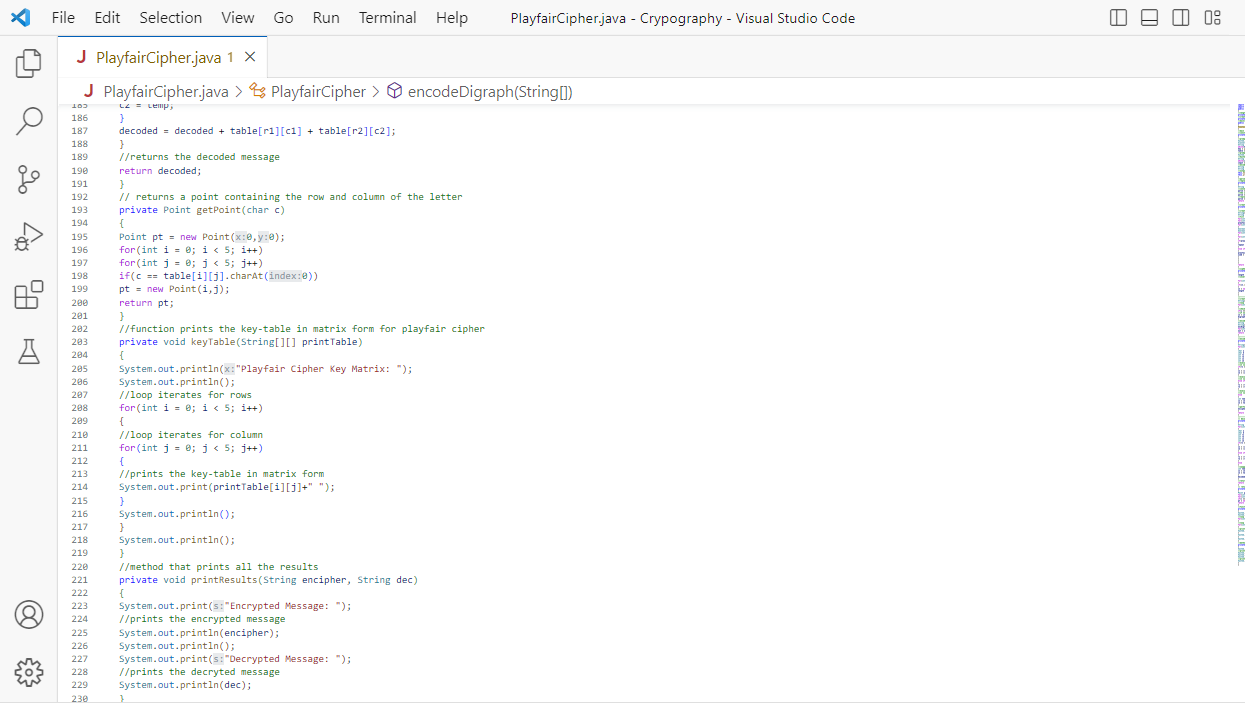
**Input:**

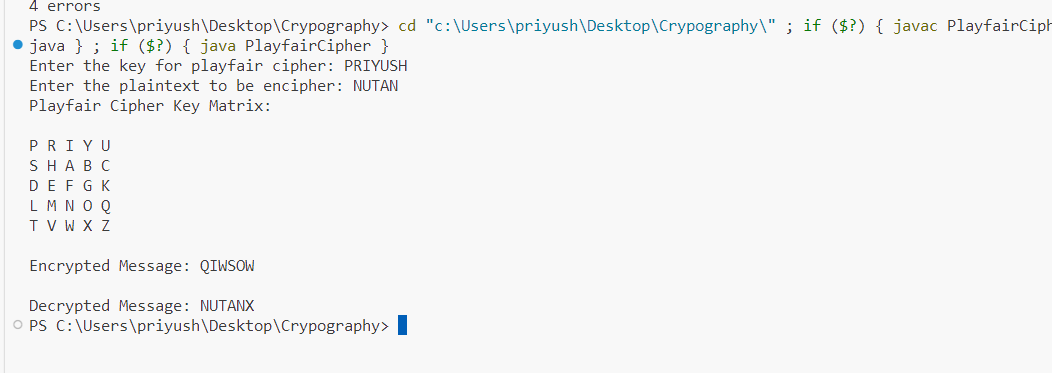
 





**Output:**

**Conclusion: -**

Thus, we have done Cryptanalysis or decoding Playfair, Vigenère cipher

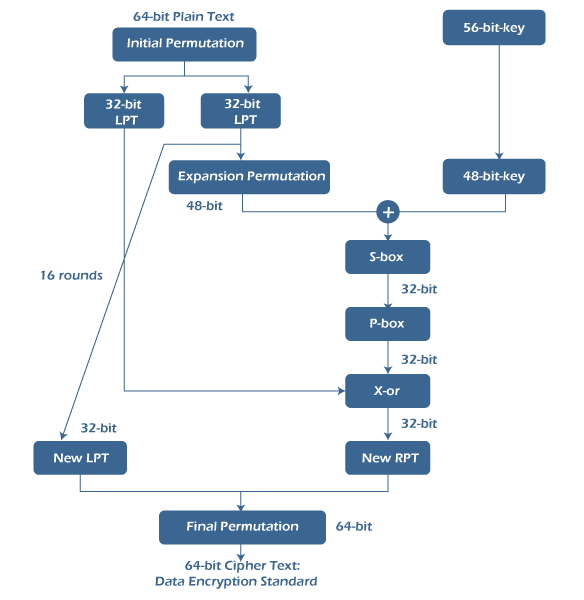
**DOP: / /2023 DOS: / /2023**

**Experiment No: 04**

**Aim**: Encrypt long messages using various modes of operation using AES or DES.

**Theory:**

**🔹What is DES:**

The Data Encryption Standard (DES) is a symmetric-key block cipher. In the year 1977, DES is published by the National Institute of Standards and Technology (NIST). It is based on the Feistel structure in which the plaintext is separated into two halves. It takes input as 64-bit plaintext and a 56-bit key to produce 64-bit ciphertext. Before processing, the entire plain text is separated into two pieces of 32 bits each, and the same operations are done on each portion. Each piece goes through 16 rounds of operations before the final permutation is used to obtain the 64-bit ciphertext.

Expansions, permutations, and substitutions are some of the functions used in the rounds, as well as an XOR operation with a round key. Decryption is done in the same way as encryption but in the opposite sequence. Although DES was regarded to be less safe for encrypting highly confidential data of government because it uses a smaller shared key, triples-DES was invented to counter this. Still, it was also not considered a good algorithm because it encrypts data very slowly. In DES, even a minor change in the input text results in a completely new ciphertext.

**🔹Advantage of DES:**

There are various advantage of DES which is as follows −

* DES has been around a long time (since 1977), even no actual weaknesses have been discovered and the most effective attack is still brute force.
* DES is an official United States Government standard. The Government is needed to re-certify, DES every five years and ask it be restored if essential.
* DES is also an ANSI and ISO standard. Because DES was designed to run on 1977 hardware, it is rapid in hardware and associatively quick in software.
* It supports functionality to save a file in an encrypted format which can only be accessed by supporting the correct password.
* It can change the system to create the directories password protected.

**🔹Disadvantage of DES:**

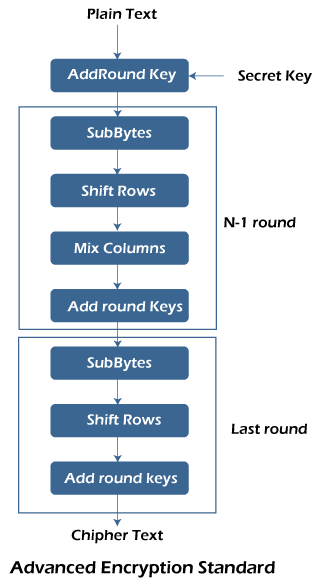
There are various disadvantage of DES which is as follows −

* The 56 bit key size is the largest defect of DES and the chips to implement one million of DES encrypt or decrypt operations a second are applicable (in 1993).
* Hardware implementations of DES are very quick.
* DES was not designed for application and therefore it runs relatively slowly.
* In a new technology, it is improving a several possibilities to divide the encrypted code, therefore AES is preferred than DES.

**🔹What is AES?**

Advanced Encryption Standard (AES) is also a symmetric key block cipher. The National Institute of Standard and Technology published AES in 2001. Because DES utilises a relatively short cipher key and the algorithm was quite slower, AES was introduced to replace it.

It is currently one of the most popular symmetric block cipher algorithms. It is at least six times faster than triple-DES encryption. Unlike DES, it is based on the "Substitution and Permutation'. It takes a step-by-step method. In AES, bytes are used instead of bits.



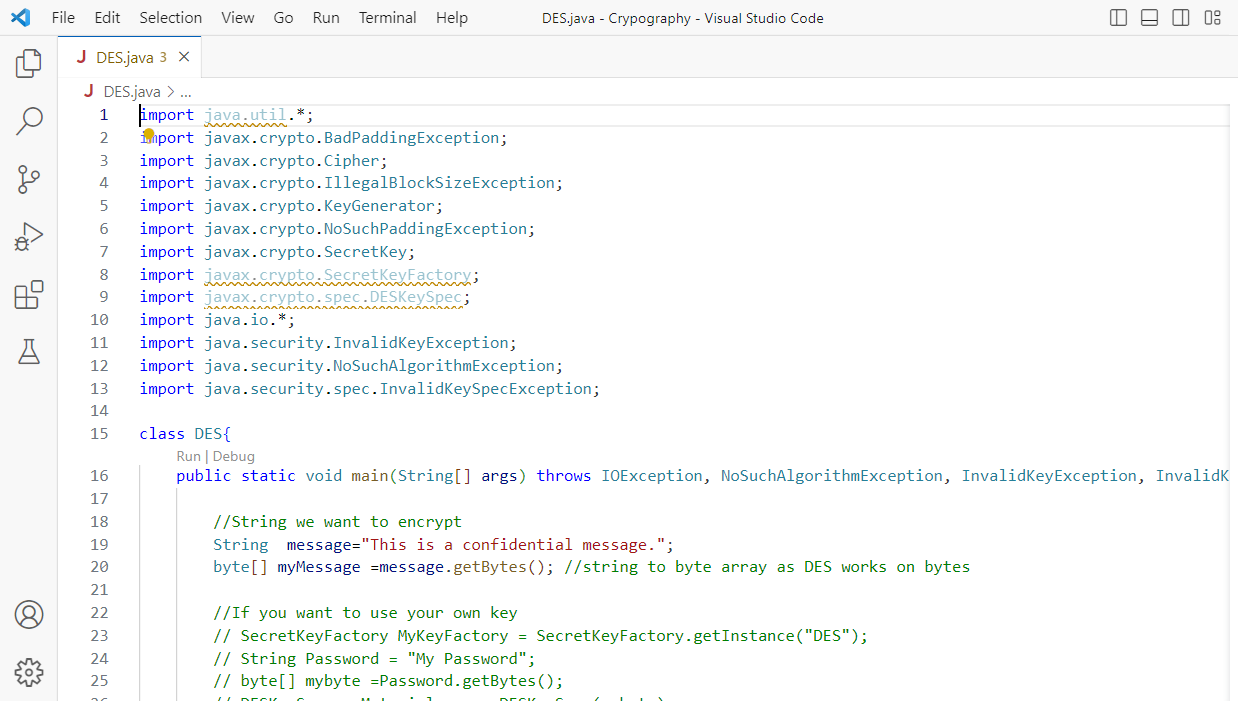
In AES, plain text is considered 126 bits equivalent to 16 bytes with a 128-bit secret key to generate a 44-bit matrix (having 4 rows and 4 columns). It then does 10 rounds after this step. Each round has its own subprocesses, with 9 rounds including Sub bytes, Shift Rows, Mix Columns and Add Round Keys. The 10th round includes all the above operations excluding 'Mix columns' in order to produce the 126-bit ciphertext.

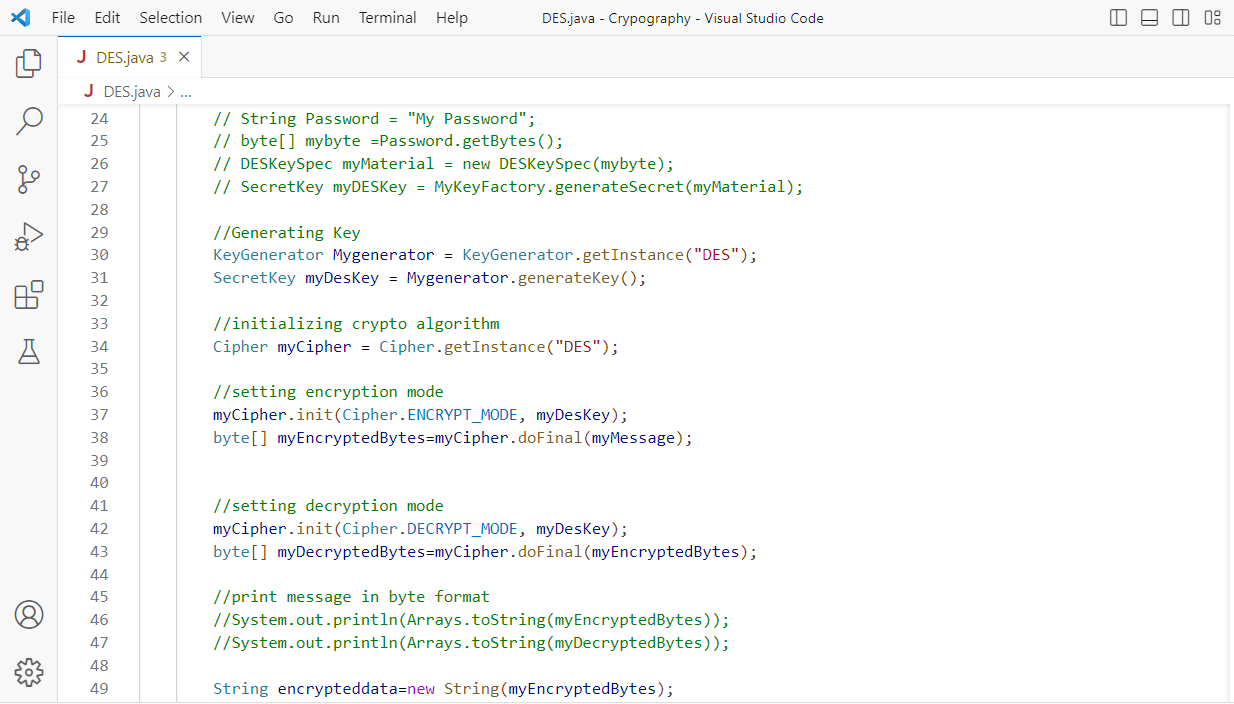
The number of rounds in AES is determined by the key size, which is 10 for 128-bit keys, 12 for 192-bit keys, and 14 for 256-bit keys. We can use it in several protocols such as TLS, SSL and numerous modern application which need high encryption security. We can also use AES for hardware which needs high throughput.

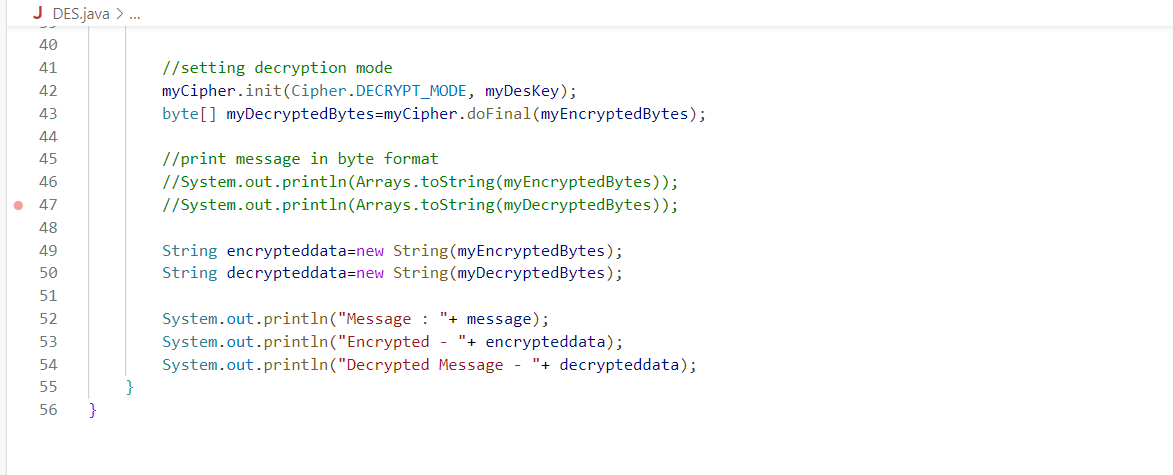
**🔹Advantages of AES over 3DES:**

* AES is more secure (it is less susceptible to cryptanalysis than 3DES).
* AES supports larger key sizes than 3DES's 112 or 168 bits.
* AES is faster in both hardware and software.
* AES's 128-bit block size makes it less open to attacks via the birthday problem than 3DES with its 64-bit block size.
* AES is required by the latest U.S. and international standards.

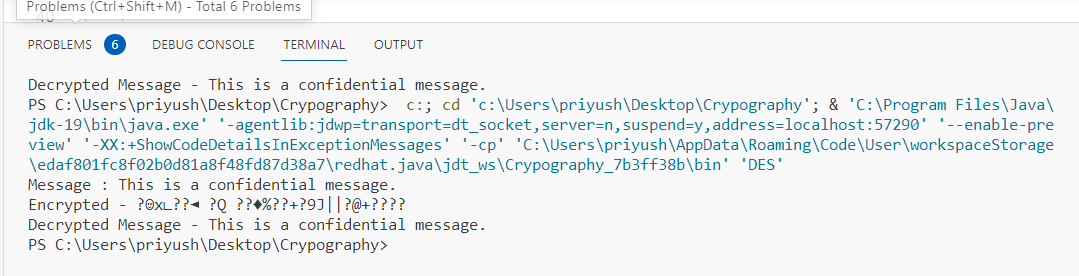
**Input:**







**Output:**



**Conclusion: - Thus** we have implemented Encrypt long messages using various modes of operation using AES or **DES.**

**DOP: / /2023 DOS: / /2023**

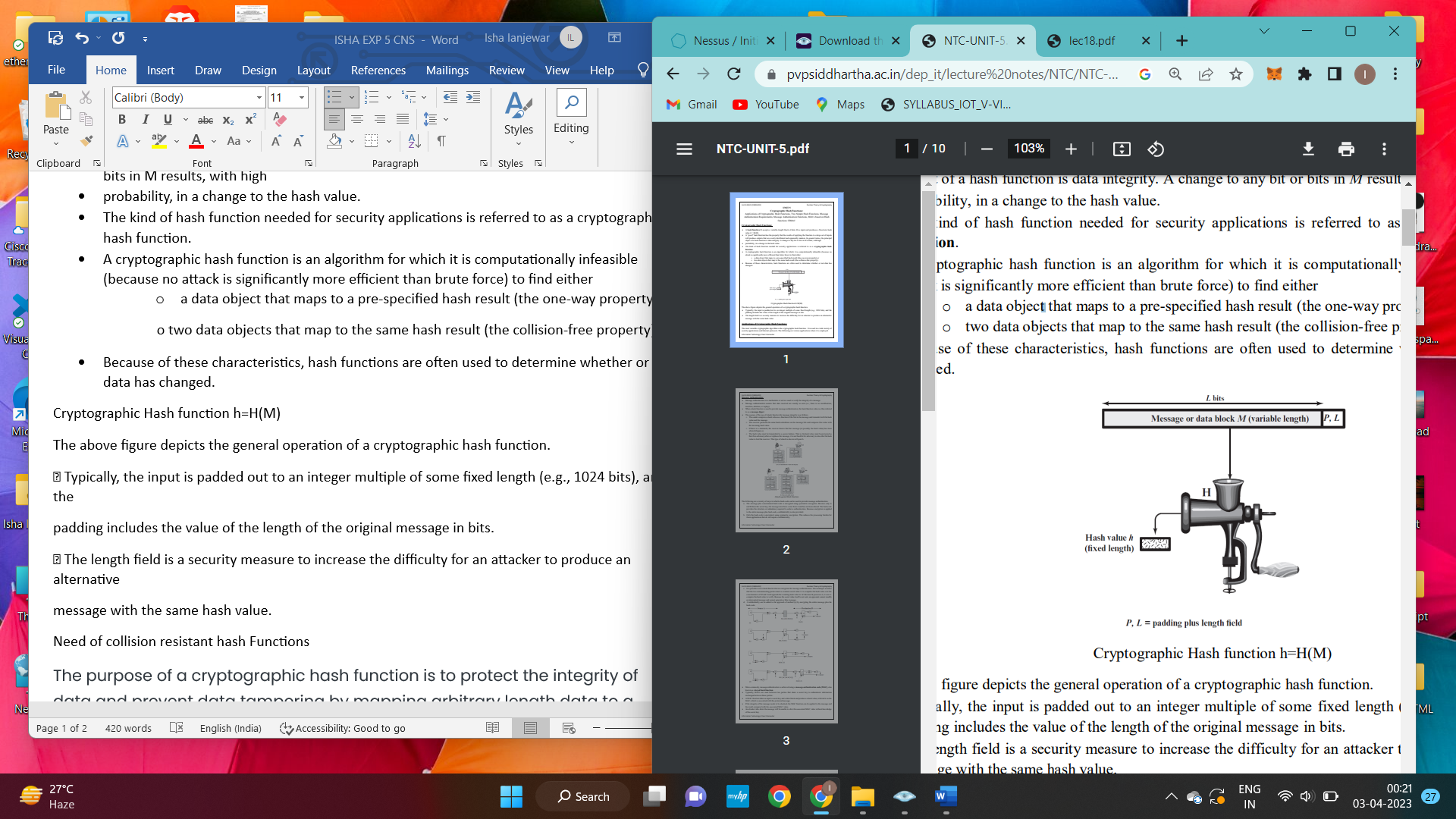
**Experiment No: 05**

**Aim**: Cryptographic Hash Functions and Applications (HMAC): to understand the need, design and applications of collision resistant 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.



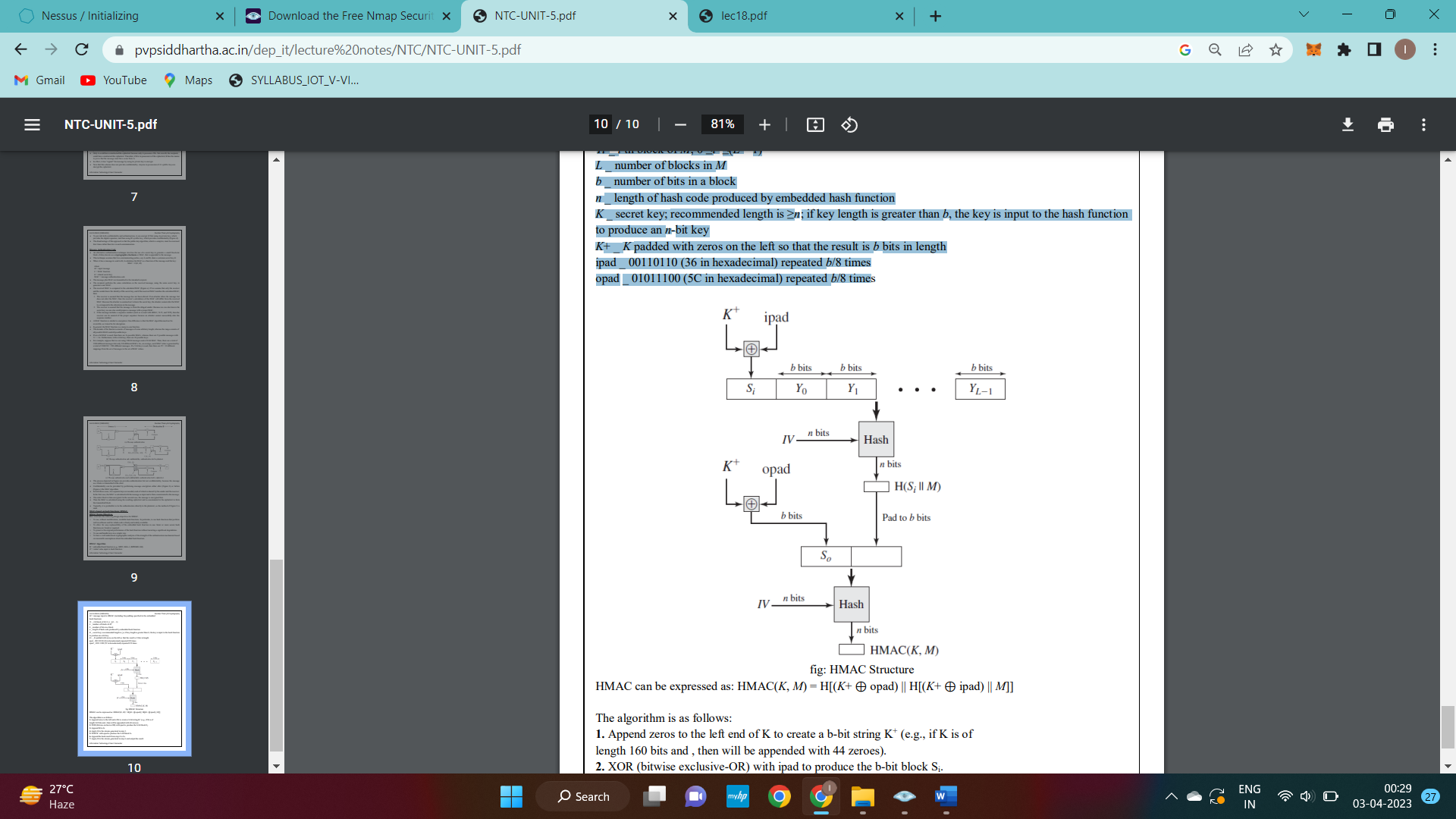
The above figure depicts the general operation of a cryptographic hash function.

Typically, the input is padded out to an integer multiple of some fixed length (e.g., 1024 bits), and the padding 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 alternative message with the same hash value.

**🔹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 embedded hash function)
* Yi \_ i th block of M, 0 ≤i ≤(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 ≥n; if key length is greater than b, the key is input to the hash function to produce an n-bit key
* K+ \_ K padded with zeros on the left so that the result is b bits in length
* ipad \_ 00110110 (36 in hexadecimal) repeated b/8 times
* opad \_ 01011100 (5C in hexadecimal) repeated b/8 time



**🔹Applications of Cryptographic Hash Functions:**

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

Message Authentication:

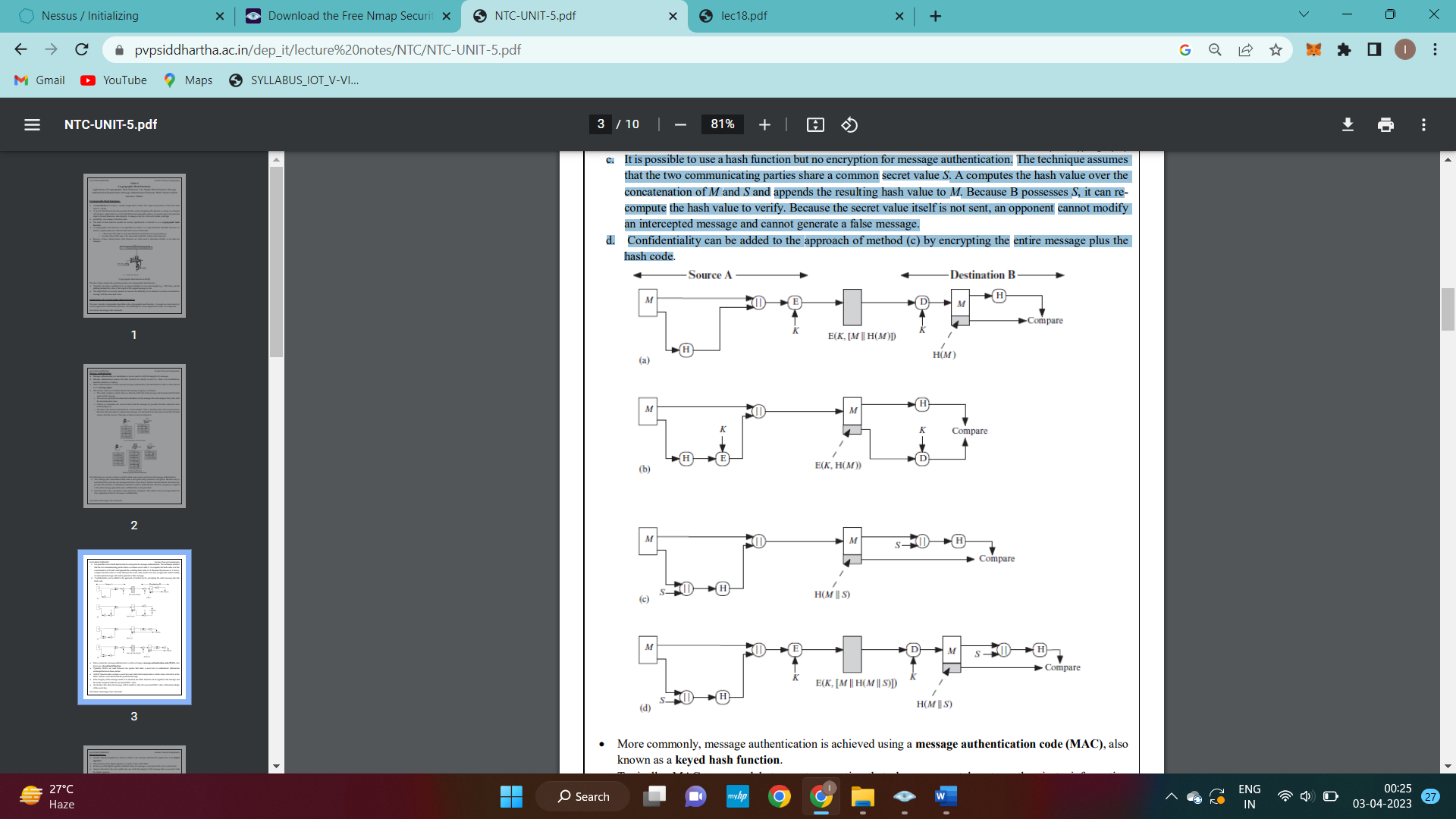
* Message authentication is a mechanism or service used to verify the integrity of a message.
* Message authentication assures that data received are exactly as sent (i.e., there is no modification, insertion, deletion, or replay)
* . When a hash function is used to provide message authentication, the hash function value is often referred to 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 hash value 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
  + The hash value must be transmitted in a secure fashion. That is, the hash value must be protected so that if an adversary alters or replaces the message, it is not feasible for adversary to also alter the hash value to fool the receiver.

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

1. 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 code provides the structure or redundancy required to achieve authentication. Because encryption is applied to the entire message plus hash code, confidentiality is also provided.
2. Only the hash code is encrypted, using symmetric encryption. This reduces the processing burden for those applications that do not require confidentiality.
3. It is possible to use a hash function but no encryption for message authentication. The technique assumes that the two communicating parties share a common secret value S. A computes the hash value over the concatenation of M and S and appends the resulting hash value to M. Because B possesses S, it can recompute the hash value to verify. Because the secret value itself is not sent, an opponent cannot modify an intercepted message and cannot generate a false message.
4. Confidentiality can be added to the approach of method (c) by encrypting the entire message plus the hash code.

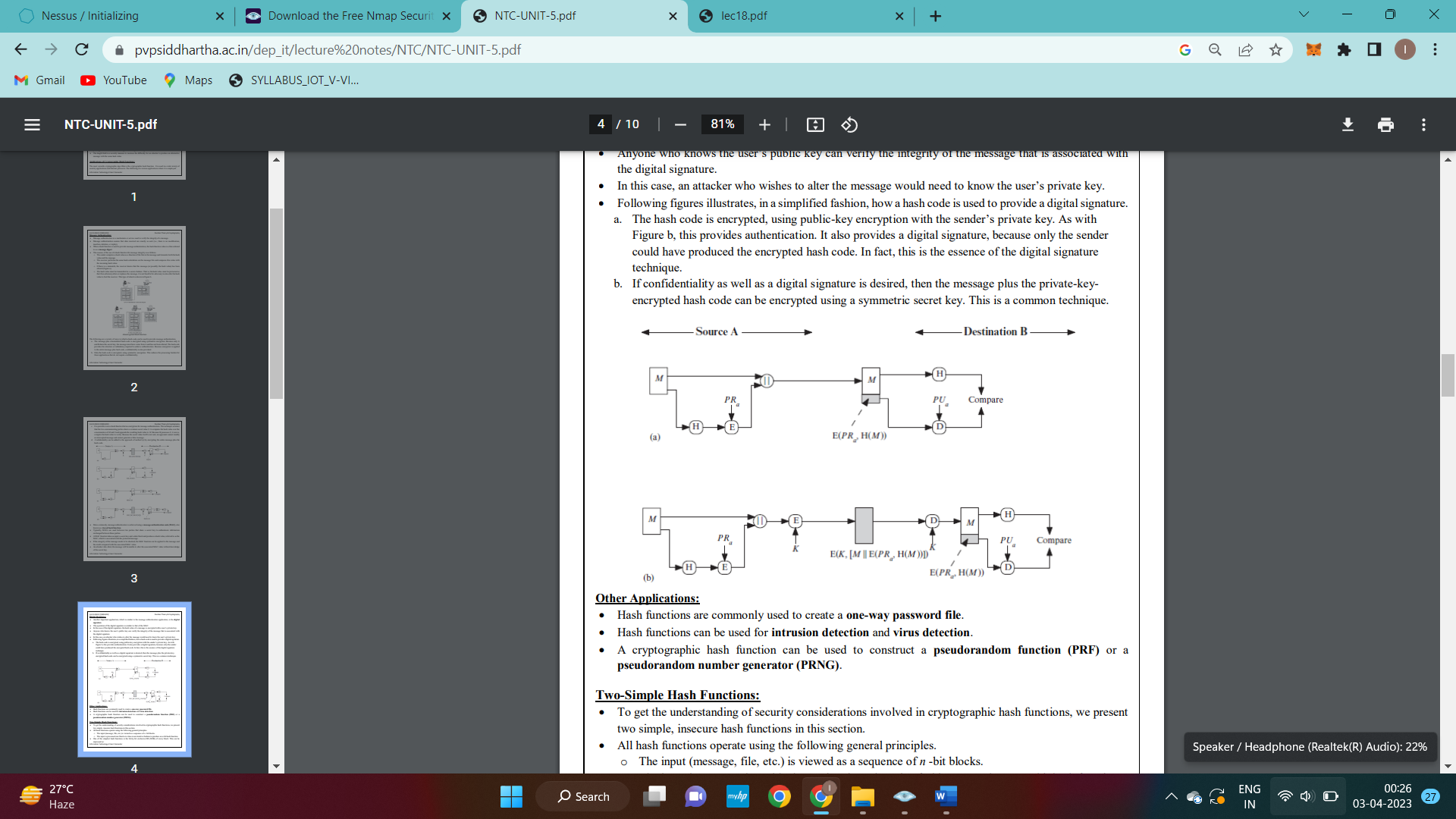


**🔹Digital Signatures:**

* Another important application, which is similar to the message authentication application, is the digital signature
* 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 with the 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.

1. 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 sender could have produced the encrypted hash code. In fact, this is the essence of the digital signature technique.
2. 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 present two simple, insecure hash functions in this section.

All hash functions operate using the following general principles.

* 1. The input (message, file, etc.) is viewed as a sequence of n -bit blocks.
  2. 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 be expressed as:

Ci = bi1 ⊕bi2 ⊕ … ⊕bim

Where

Ci = i th bit of the hash code, 1 … i … n

m = number of n-bit blocks in the input

bij = i th bit in j th bloc

k ⊕ = XOR operation

* This operation produces a simple parity bit for each bit position and is known as a longitudinal redundancy check.
* It is reasonably effective for random data as a data integrity check. Each n-bit hash value is equally likely.
* 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 of data 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 value after 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 that appear in the input.
* Although the second procedure provides a good measure of data integrity, it is virtually useless for data security 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, you may still feel that such a simple function could be useful when the message together with the hash code is encrypted.

**Conclusion:** Hence successfully studied the need, design and application of collusion resistant hash function of Cryptographic Hash Functions and Applications (HMAC).

**DOP: / /2023 DOS: / /2023**

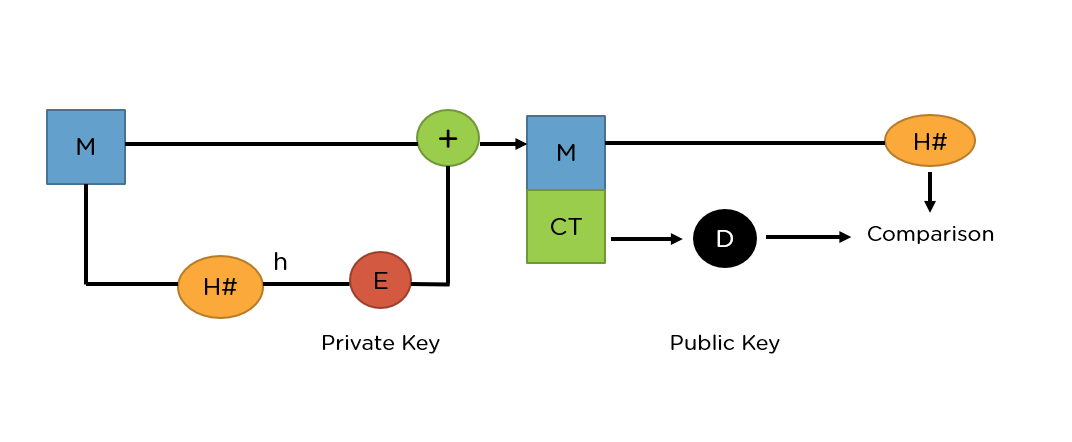
**Experiment No: 06**

**Aim**: Implementation and analysis of RSA cryptosystem and Digital signature scheme using RSA.

**Theory:**

**🔹RSA algorithm:**

The RSA algorithm is a public-key signature algorithm developed by Ron Rivest, Adi Shamir, and Leonard Adleman. Their paper was first published in 1977, and the algorithm uses logarithmic functions to keep the working complex enough to withstand brute force and streamlined enough to be fast post-deployment. The image below shows it verifies the digital signatures using RSA methodology.



**🔹RSA in Data Encryption**

When using RSA for encryption and m decryption of general data, it reverses the key set usage. Unlike signature verification, it uses the receiver’s public key to encrypt the data, and it uses the receiver’s private key in decrypting the data. Thus, there is no need to exchange any keys in this scenario.

There are two broad components when it comes to RSA cryptography, they are:

* Key Generation: Generating the keys to be used for encrypting and decrypting the data to be exchanged.
* Encryption/Decryption Function: The steps that need to be run when scrambling and recovering the data.

You will now understand each of these steps in our next sub-topic.

**🔹Steps in RSA Algorithm:**

Keeping the image above in mind, go ahead and see how the entire process works, starting from creating the key pair, to encrypting and decrypting the information.

**Key Generation**

You need to generate public and private keys before running the functions to generate your ciphertext and plaintext. They use certain variables and parameters, all of which are explained below:

* Choose two large prime numbers (p and q)
* Calculate n = p\*q and z = (p-1)(q-1)
* Choose a number e where 1 < e < z
* Calculate d = e-1mod(p-1)(q-1)
* You can bundle private key pair as (n,d)
* You can bundle public key pair as (n,e)

**Encryption/Decryption Function**

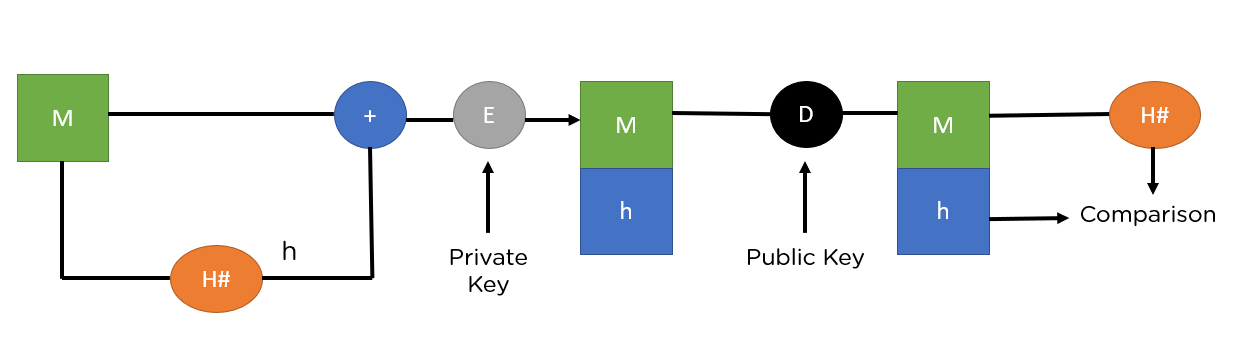
Once you generate the keys, you pass the parameters to the functions that calculate your ciphertext and plaintext using the respective key.

* If the plaintext is m, ciphertext = me mod n.
* If the ciphertext is c, plaintext = cd mod n

**🔹What Are Digital Signatures?**

Digital signatures serve the purpose of authentication and verification of documents and files. This is crucial to prevent tampering during official papers’ transmission and prevent digital manipulation or forgery.

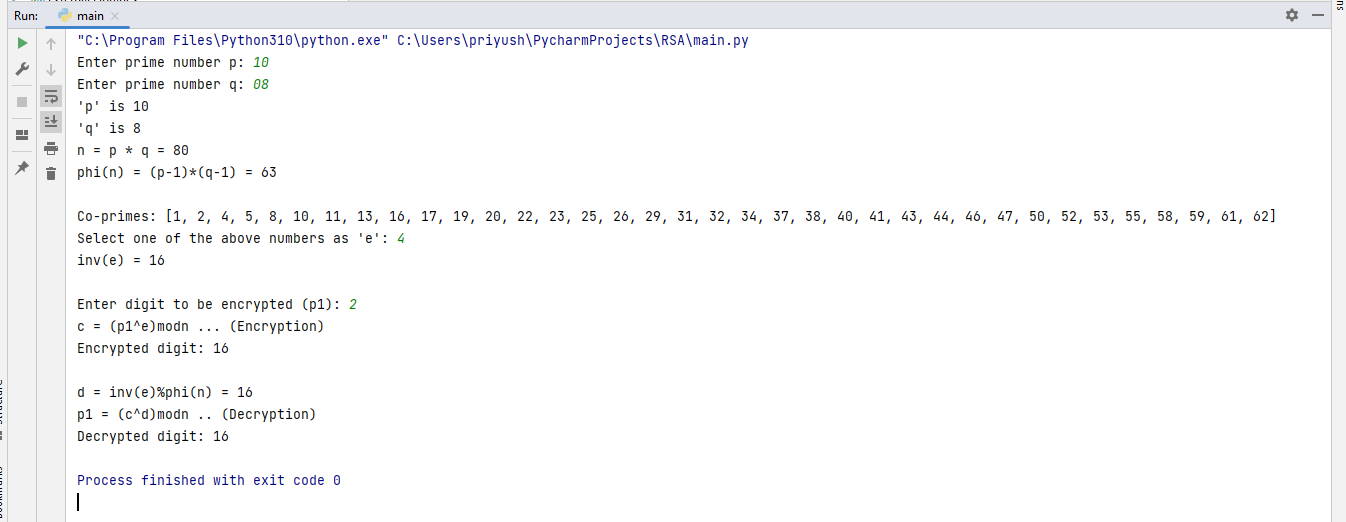
They work on the public key cryptography architecture, barring one small caveat. Typically, the asymmetric key system uses a public key for encryption and a private key for decryption. However, when dealing with digital signatures, it’s the opposite. The private key is used to encrypt the signature, and the public key is used to decrypt it. Since the keys work in tandem with each other, decrypting it with the public key signifies it used the correct private key to sign the document, hence authenticating the origin of the signature.



**Input:**

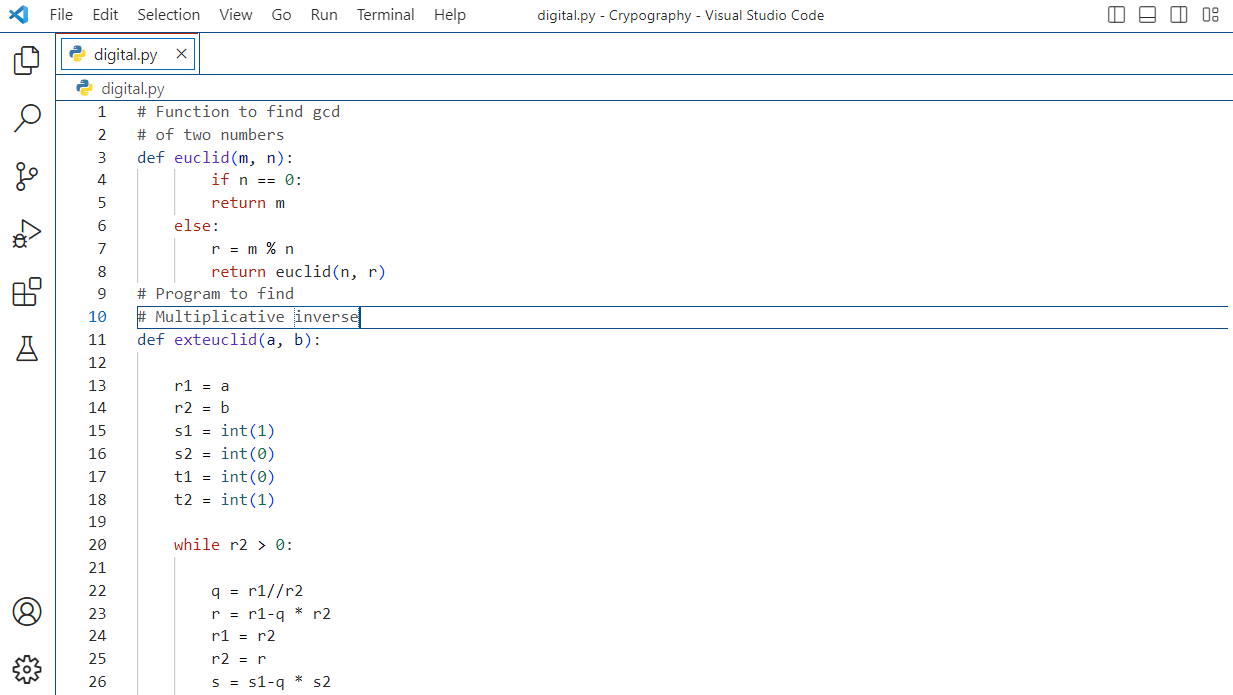
****

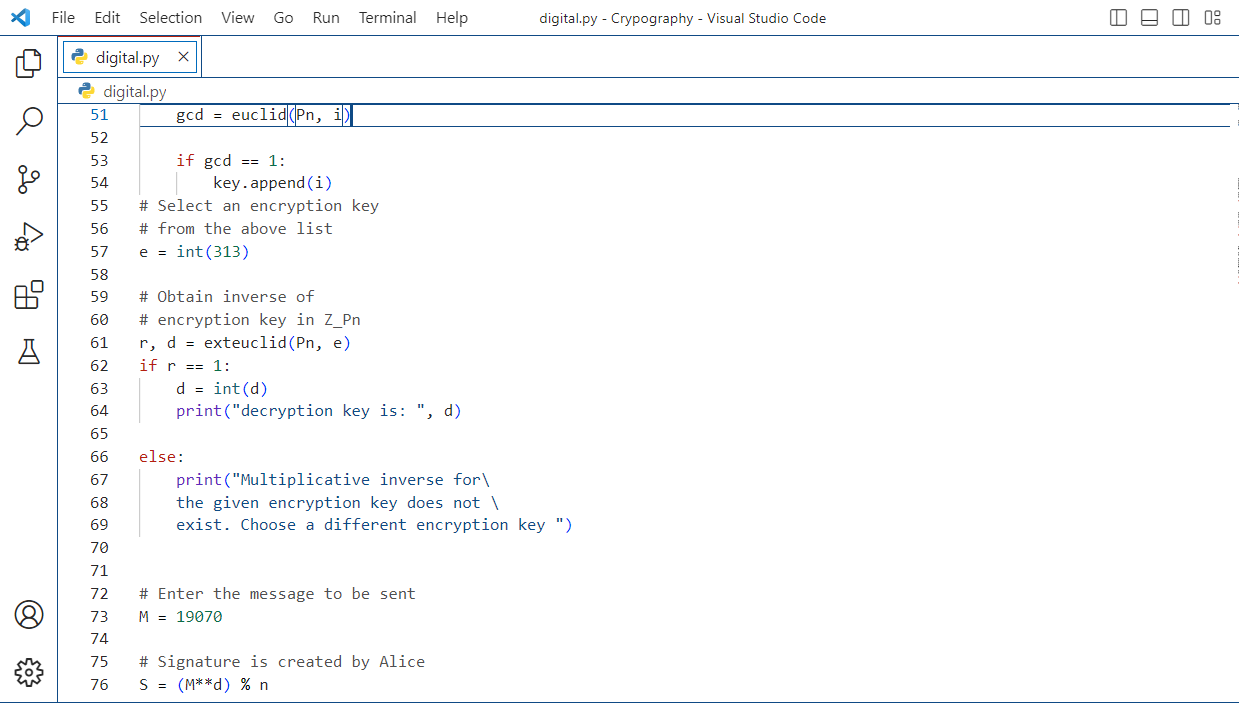
**Output:**



**Digital signature scheme using RSA.**

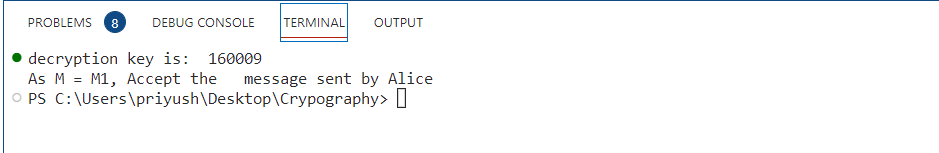
**Input:**





**Output:**



**🔹Advantages of RSA:**

* **No Key Sharing**: RSA encryption depends on using the receiver’s public key, so you don’t have to share any secret key to receive messages from others.
* **Proof of Authenticity**: Since the key pairs are related to each other, a receiver can’t intercept the message since they won’t have the correct private key to decrypt the information.
* **Faster Encryption**: The encryption process is faster than that of the DSA algorithm.
* **Data Can’t Be Modified**: Data will be tamper-proof in transit since meddling with the data will alter the usage of the keys. And the private key won’t be able to decrypt the information, hence alerting the receiver of manipulation.

**Conclusion: - Thus** weImplementation and analysis of RSA cryptosystem and Digital signature scheme using RSA.

**DOP: / /2023 DOS: / /2023**

**Experiment No: 07**

**Aim**: Study the use of network reconnaissance tools like WHOIS, dig, traceroute, nslookup to gather information about networks and domain registrars.

**Theory:**

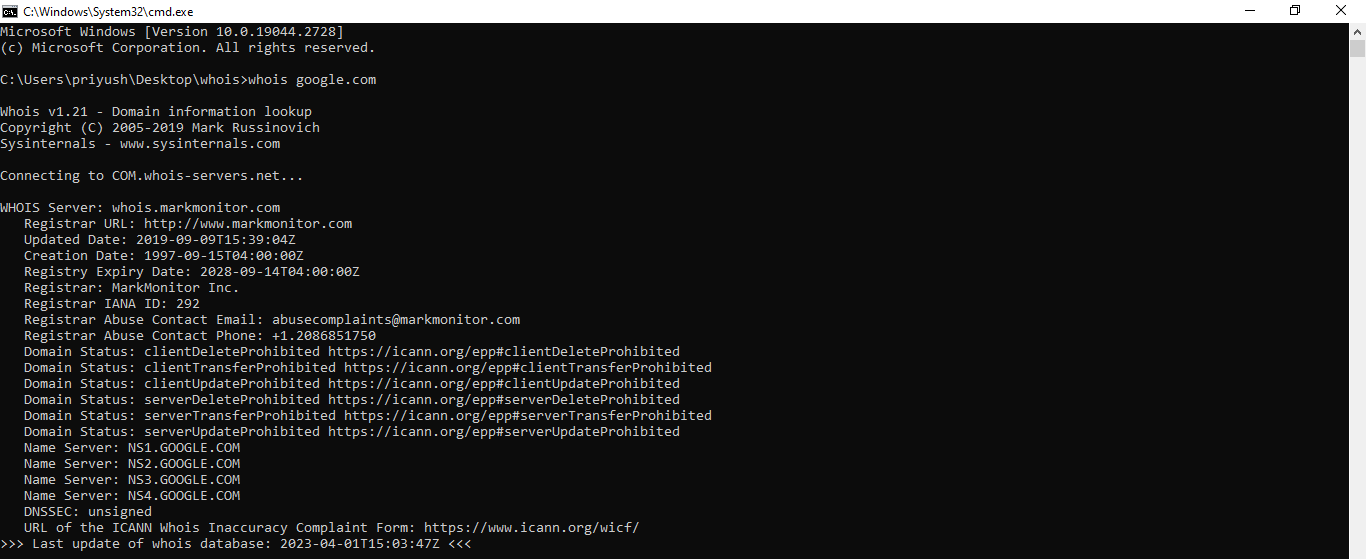
**🔹Reconnaissance :**

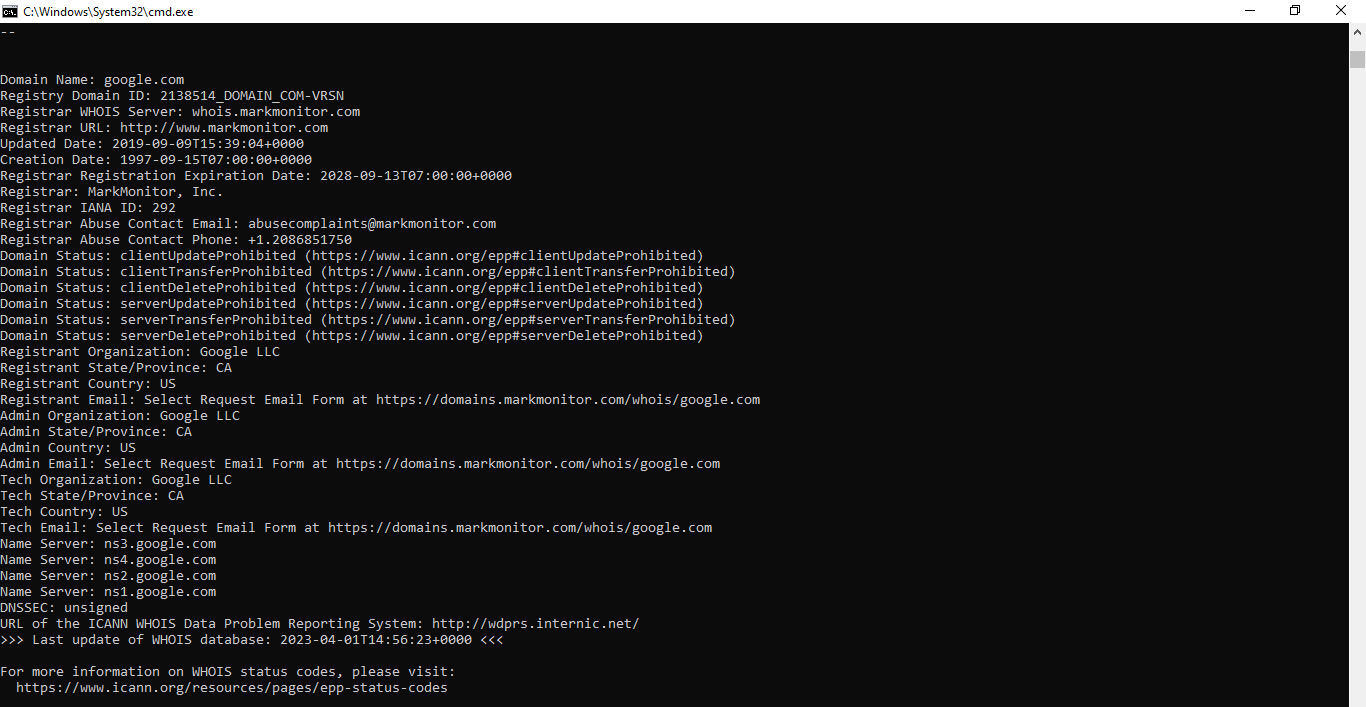
Reconnaissance is the information-gathering stage of ethical hacking, where you collect data about the target system. This data can include anything from network infrastructure to employee contact details. The goal of reconnaissance is to identify as many potential attack vectors as possible.

**🔹WHOIS:**

The whois command displays information about a website's record. You may get all the information about a website regarding its registration and owner's information.

whois **<websiteName>**

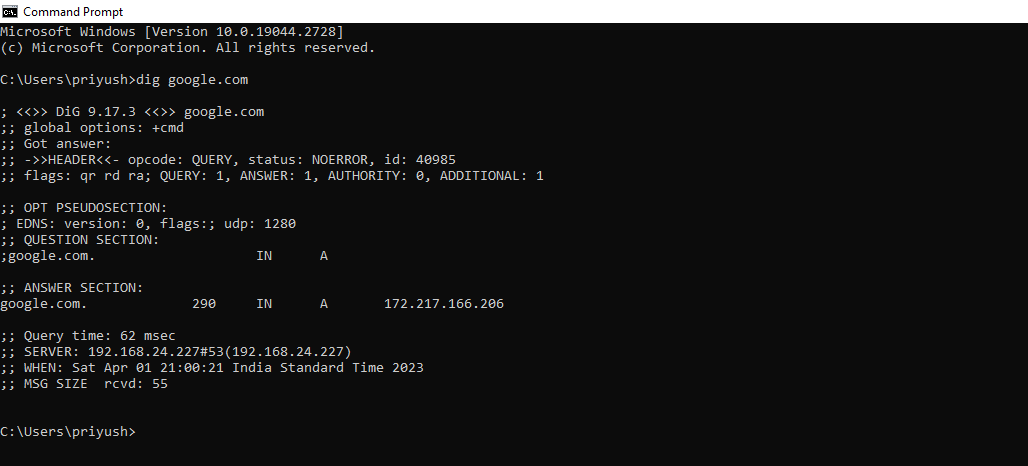
****

****

**🔹dig:**

Linux dig command stands for **Domain Information Groper**. This command is used for tasks related to DNS lookup to query DNS name servers. It mainly deals with troubleshooting DNS related problems. It is a flexible utility for examining the DNS (Domain Name Servers). It is used to perform the DNS lookups and returns the queried answers from the name server. Usually, it is used by most DNS administrators to troubleshoot the DNS problems. It is a straightforward tool and provides a clear output. It is more functional than other lookups tools.

dig @server name type

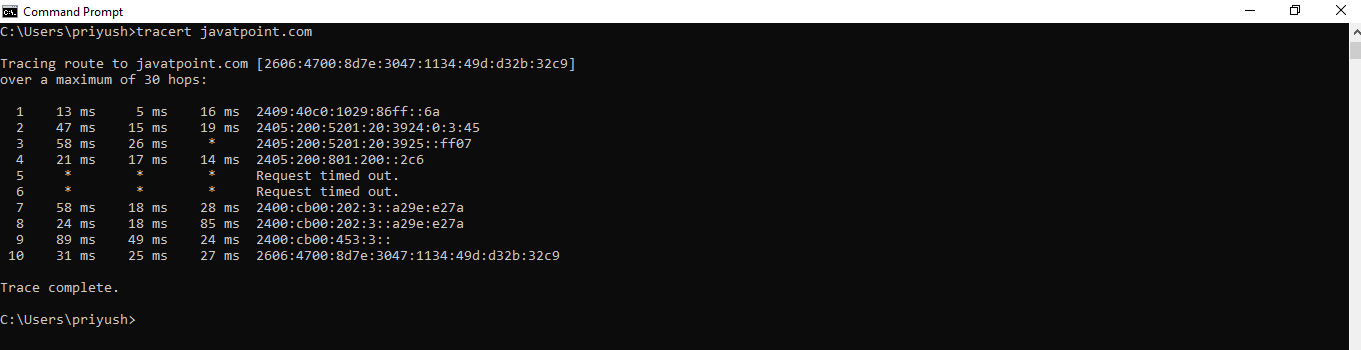


**🔹traceroute:**

traceroute is a network troubleshooting utility that helps us determine the number of hops and packets traveling path required to reach a destination. It is used to display how the data transmitted from a local machine to a remote machine. Loading a web page is one of the common examples of the traceroute.

Traceroute prints the route that packets take to a network host. Traceroute utility uses the TTL field in the IP header to achieve its operation. For users who are new to TTL field, this field describes how much hops a particular packet will take while traveling on network.

traceroute [OPTION...] HOST



**🔹nslookup:**

nslookup is a great utility for diagnosing DNS name resolution problems. Just type the nslookup command, and Windows will display the name and IP address of the device’s default DNS server. From there, you can type host names in an effort to see if the DNS server is able to resolve the specified host name.

The nslookup command is used to query internet name servers interactively for information. nslookup, which stands for "name server lookup", is a useful tool for finding out information about a named domain. By default, nslookup will translate a domain name to an IP address (or vice versa).

nslookup **<domainName>**



**Conclusion: -**

In this experiment you learned how to take the first steps toward ethical hacking. Information gathering, in the form of reconnaissance, foot printing, and social engineering, is necessary to learn as much about the target as possible. By following the information-gathering methodology, ethical hackers can ensure they are not missing any steps and valuable information. Time spent in the information gathering phase is well worth it toss peed up and produce successful hacking exploits.

**DOP: / /2023 DOS: / /2023**

**Experiment No: 08**

**Aim**: Study of packet sniffer tools wireshark: -

1. Observer performance in promiscuous as well as non-promiscuous mode.
2. Show the packets can be traced based on different filters.

**Theory:**

**🔹Wireshark:**

Wireshark is an open-source packet analyzer, which is used for education, analysis, software development, communication protocol development, and network troubleshooting.

It is used to track the packets so that each one is filtered to meet our specific needs. It is commonly called as a sniffer, network protocol analyzer, and network analyzer. It is also used by network security engineers to examine security problems.

**🔹Uses of Wireshark:**

Wireshark can be used in the following ways:

* It is used by network security engineers to examine security problems.
* It allows the users to watch all the traffic being passed over the network.
* It is used by network engineers to troubleshoot network issues.
* It also helps to troubleshoot latency issues and malicious activities on your network.
* It can also analyze dropped packets.

**🔹features:**

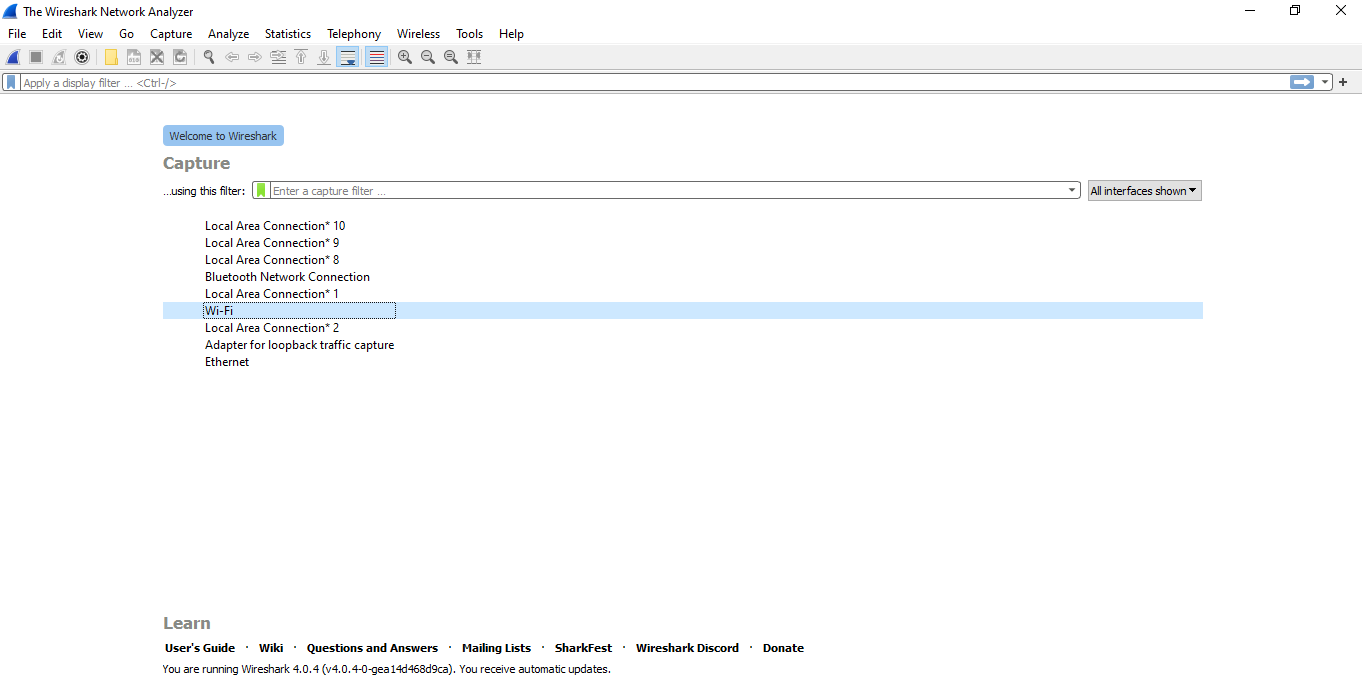
The following are some of the many features wireshark provides:

* Available for UNIX and Windows.
* Capture live packet data from a network interface.
* Open files containing packet data captured with tcpdump/WinDump, Wireshark,
* and a number of other packet capture programs.
* Import packets from text files containing hex dumps of packet data.
* Display packets with very detailed protocol information.

Wireshark can also monitor the unicast traffic which is not sent to the network's MAC address interface. But the switch does not pass all the traffic to the port. Hence, the **promiscuous mode** is not sufficient to see all the traffic. The various network taps or port mirroring is used to extend capture at any point.

**🔹Capturing Packets:**

After downloading and installing wireshark, you can launch it and click the name of an interface under Interface List to start capturing packets on that interface. For example, if you want to capture traffic on the wireless network, click your wireless interface. You can configure advanced features by clicking Capture Options.

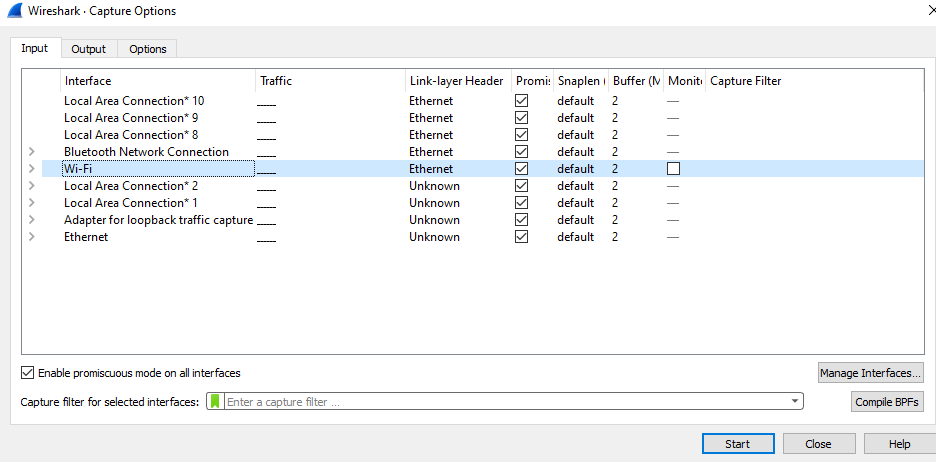


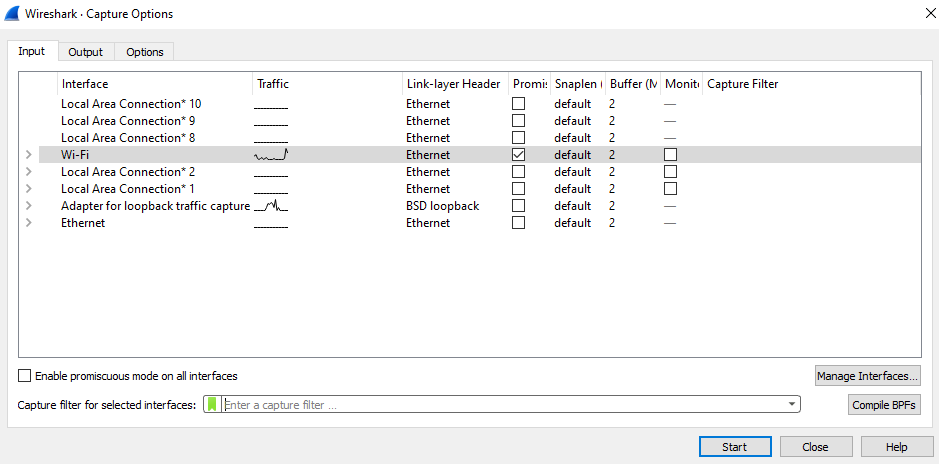
As soon as you click the interface‘s name, you‘ll see the packets start to appear in real time. Wireshark captures each packet sent to or from your system. If you‘re capturing on a wireless interface and have promiscuous mode enabled in your capture options, you‘ll also see other the other packets on the network.

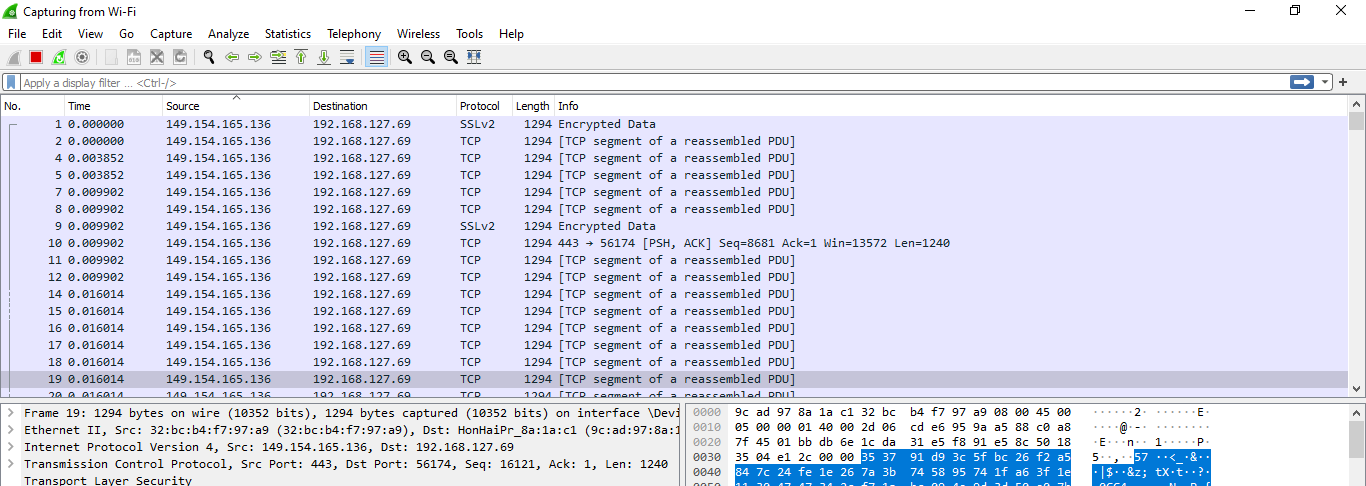
**🔹performance in promiscuous as well as non-promiscuous mode:**

To turn on promiscuous mode, click on the CAPTURE OPTIONS dialog box and select it from the options. If everything goes according to plan, you'll now see all the network traffic in your network. However, many network interfaces aren't receptive to promiscuous mode, so don't be alarmed if it doesn't work for you.

Click on the network and make sure the promiscuous mode settings are set to ALLOW ALL. Promiscuous mode enables lots of Wireshark's functions, so you should do all you can to make sure your interface can use it, if possible.





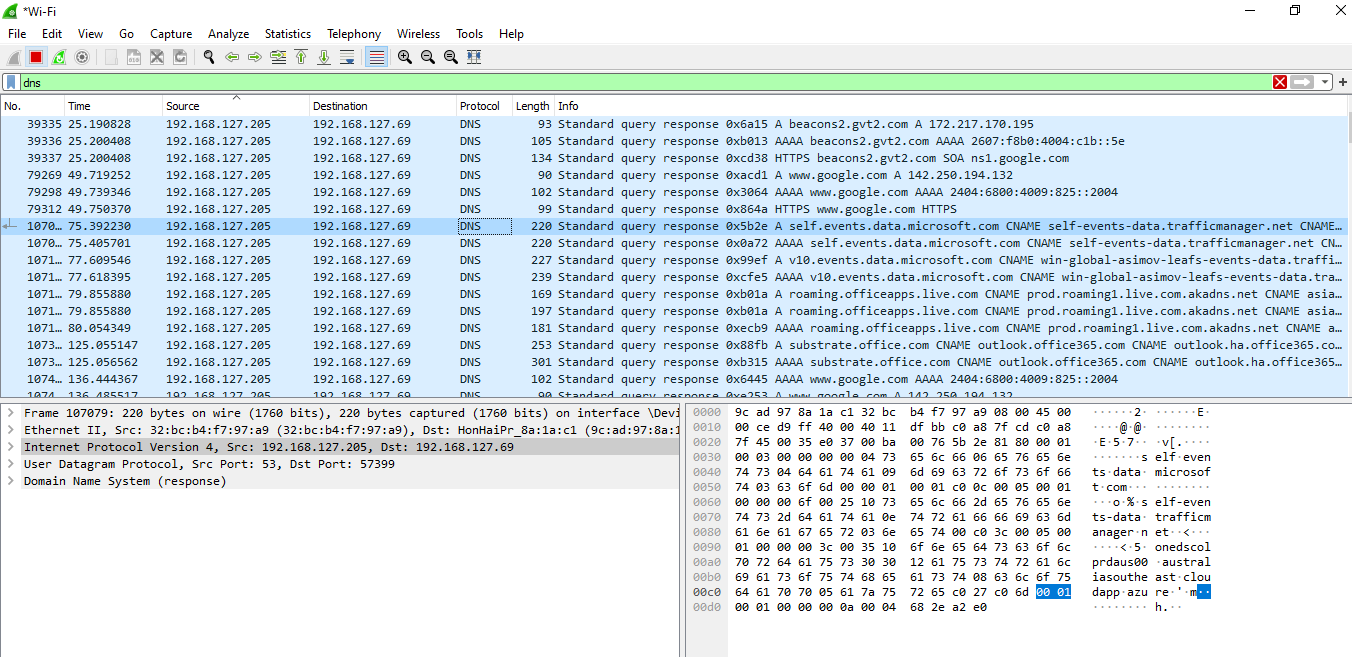


Wireshark uses colors to help you identify the types of traffic at a glance. By default, green is TCP traffic, dark blue is DNS traffic, light blue is UDP traffic, and black identifies TCP packets with problems — for example, they could have been delivered out-of-order.

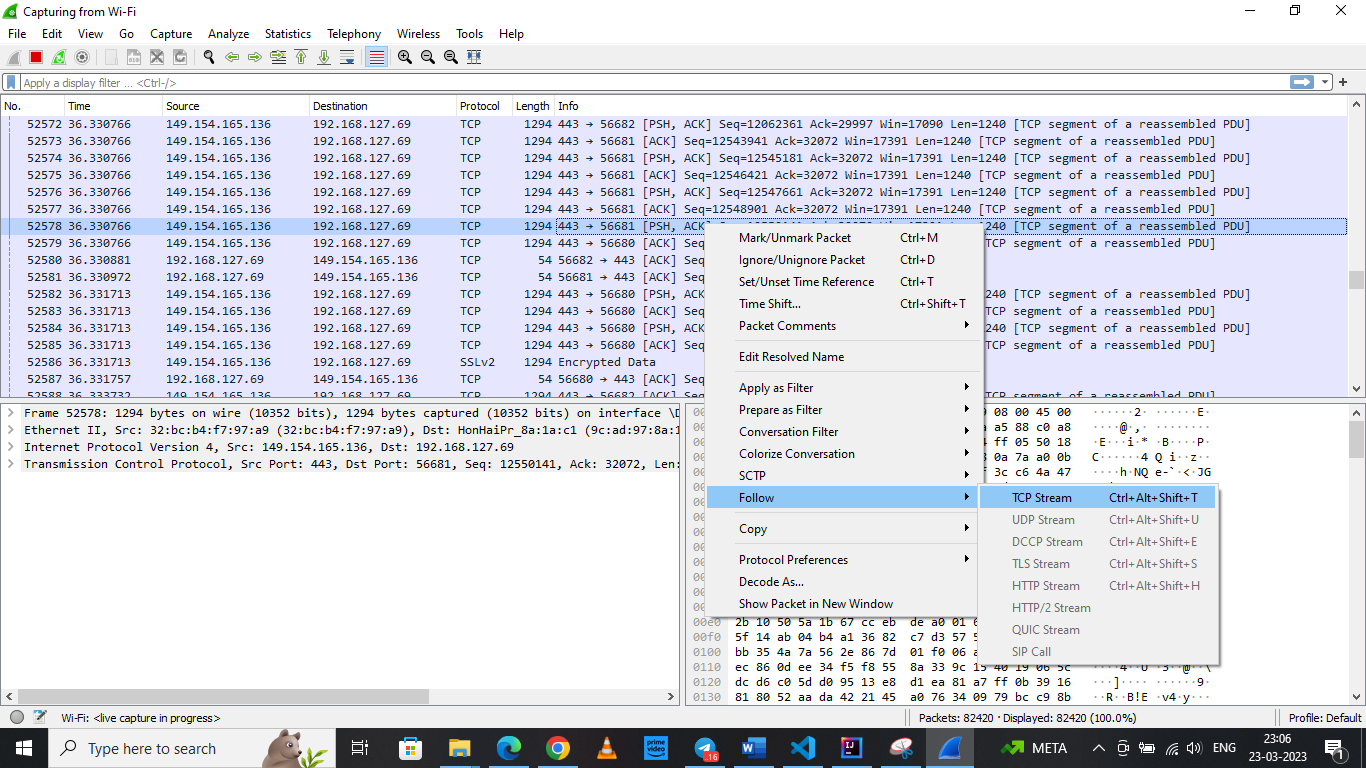
**🔹Filtering Packets:**

If you‘re trying to inspect something specific, such as the traffic a program sends when phoning home, it helps to close down all other applications using the network so you can narrow down the traffic. Still, you‘ll likely have a large amount of packets to sift through. That‘s where Wireshark‘s filters come in.

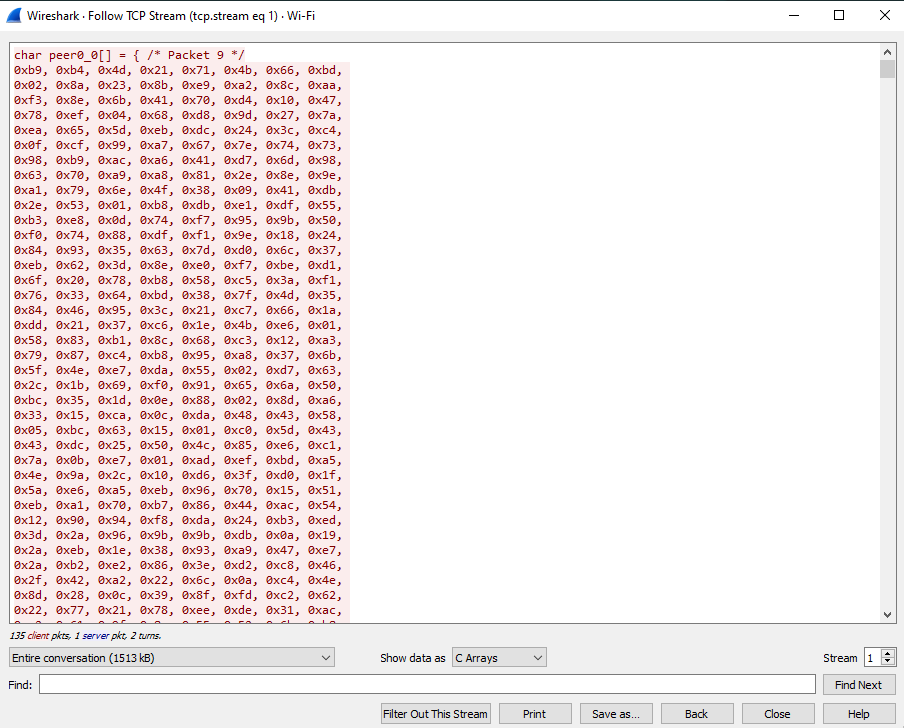
The most basic way to apply a filter is by typing it into the filter box at the top of the window and clicking Apply (or pressing Enter). For example, type ―dns‖ and you‘ll see only DNS packets. When you start typing, Wireshark will help you autocomplete your filter.



Another interesting thing you can do is right-click a packet and select Follow TCP Stream



You‘ll see the full conversation between the client and the server.

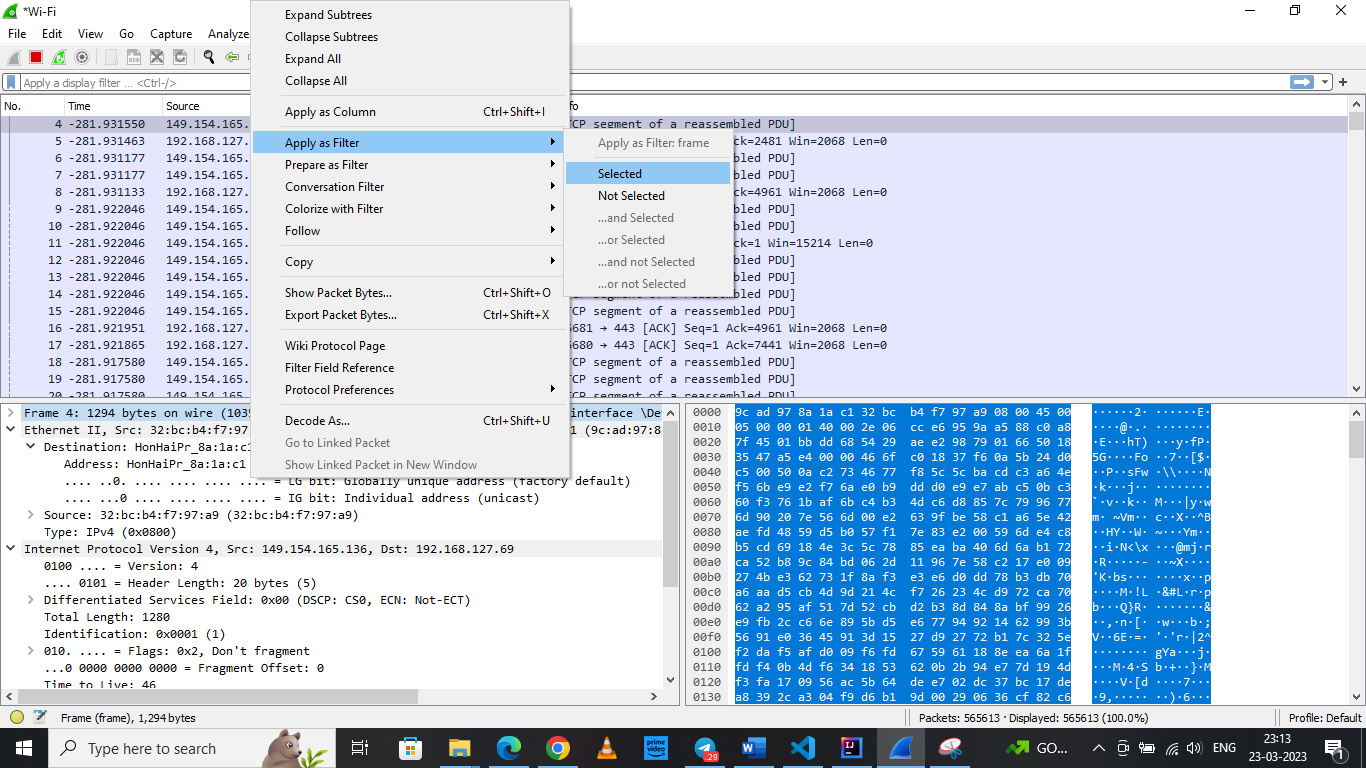


Close the window and you‘ll find a filter has been applied automatically — Wireshark is showing you the packets that make up the conversation.

**🔹Inspecting Packets**

Click a packet to select it and you can dig down to view its details.

You can also create filters from here — just right-click one of the details and use the Apply as Filter submenu to create a filter based on it.



Wireshark is an extremely powerful tool, and this tutorial is just scratching the surface of what you can do with it. Professionals use it to debug network protocol implementations, examine security problems and inspect network protocol internals.

**Conclusion: -**

In this experiment we analyze various packet sniffing tools that monitor network traffic transmitted between legitimate users or in the network. The packet sniffer is network monitoring tool. It is opted for network monitoring, traffic analysis, troubleshooting, Packet grapping, message, protocol analysis, penetration testing and many other purposes

**DOP: / /2023 DOS: / /2023**

**Experiment No: 09**

**Aim**: Download, install nmap and use it with different options to scan open ports, perform OS fingerprinting, ping scan, tcp port scan, udp port scan, etc.

**Theory:**

**🔹Nmap:**

Nmap, short for Network Mapper, is a free and open-source tool used for vulnerability checking, port scanning and, of course, network mapping. Despite being created back in 1997, Nmap remains the gold standard against which all other similar tools, either commercial or open source, are judged.

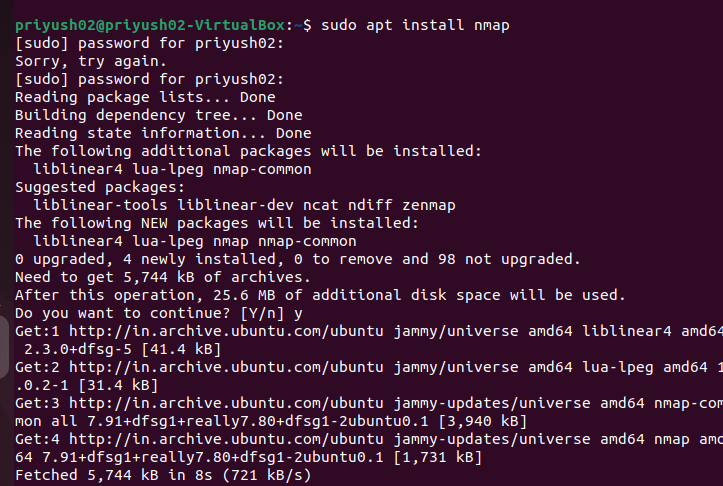
Nmap has maintained its pre-eminence because of the large community of developers and coders who help to maintain and update it. The Nmap community reports that the tool, which anyone can get for free, is downloaded several thousand times every week.

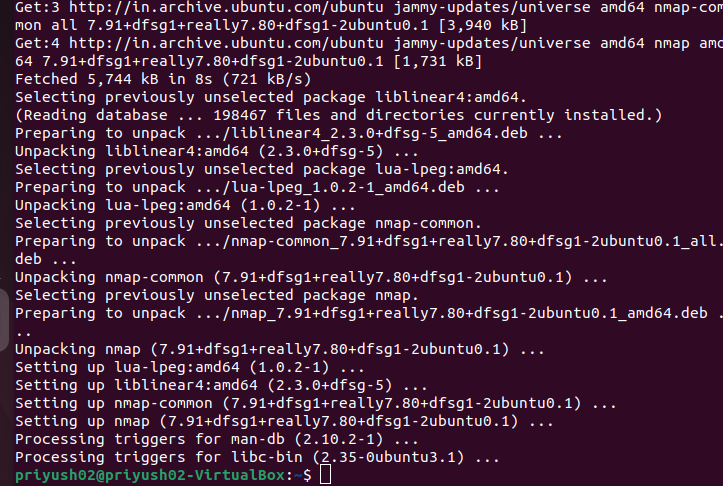
Because of its flexible, open-source code base, it can be modified to work within most customized or heavily specialized environments. There are distributions of Nmap specific to Windows, Mac and Linux environments, but Nmap also supports less popular or older operating systems like Solaris, AIX or AmigaOS. The source code is available in C, C++, Perl and Python.

**🔹Nmap features include:**

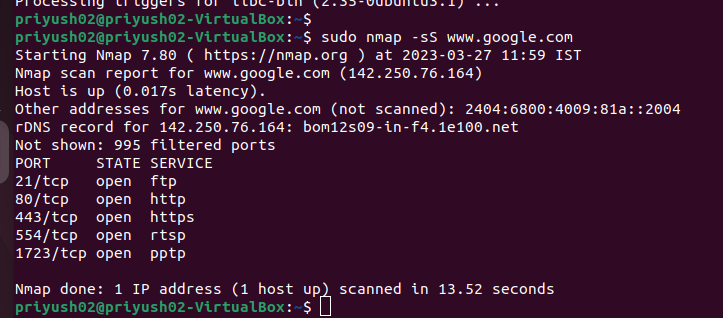
* **Host Discovery** – Identifying hosts on a network. For example, listing the hosts which respond to pings or have a particular port open.
* **Port Scanning** – Enumerating the open ports on one or more target hosts.
* **Version Detection** – Interrogating listening network services listening on remote devices to determine the application name and version number.
* **OS Detection** – Remotely determining the operating system and some hardware characteristics of network devices.

The installation of **nmap:> sudo apt-get install nmap**

****

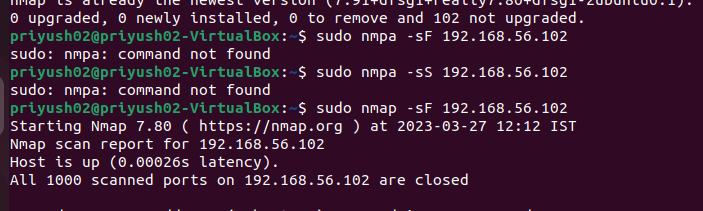
**🔹nmap -sS for TCP SYN Scan:**

It is required privilege access and identifies TCP ports. TCP SYN Scan is a standard method for detecting open ports without going through the Three-way Handshake process. When an open port is spotted, the TCP handshake is reset before accomplishment. Hence this scanning is also called Half Open scanning**.**



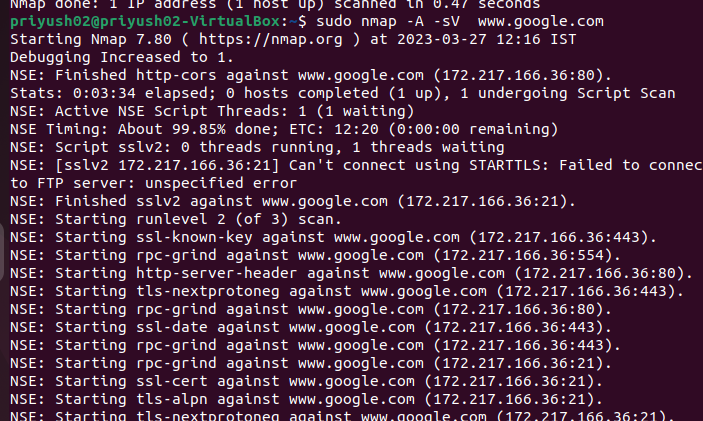
**🔹nmap -sF for FIN Scan:**

FIN scan transmits packets with a FIN flag to the target machine; therefore, these frames are abnormal as they are sent to the destination before the Three-way handshaking process can be completed. If there is no active TCP session, then the port is formally closed. If the destination machine's port is closed then the RST packet in the FIN Scan response is reversed.

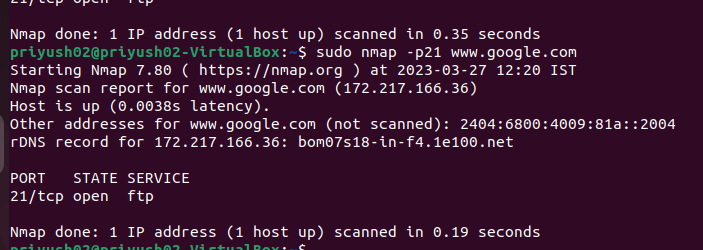


**🔹sV (Version detection):**

Enables version detection, as discussed above. Alternatively, we can use -A, which enables version detection among other things,



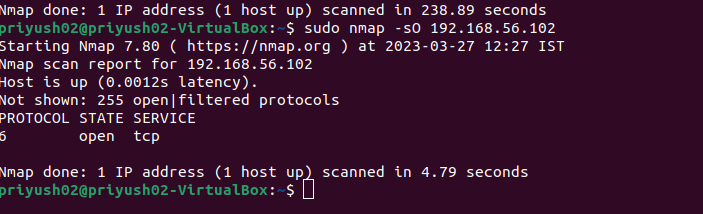
Multiple scan ports:



**🔹sO (IP protocol scan)**.

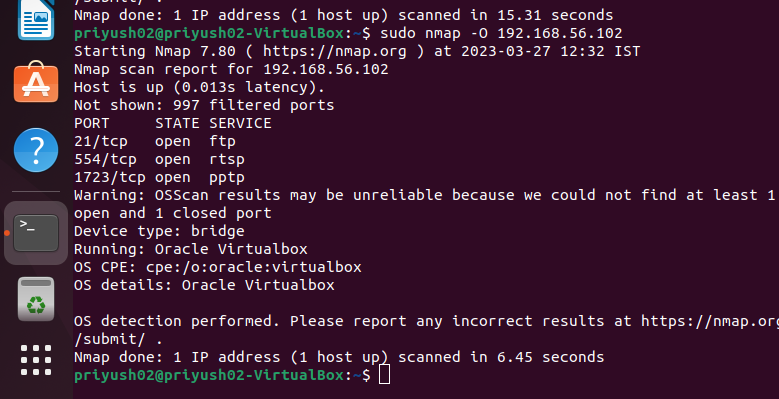
IP protocol scan allows you to determine which IP protocols (TCP, ICMP, IGMP, etc.) are supported by target machines. This isn´t technically a port scan, since it cycles through IP protocol numbers rather than TCP or UDP port numbers





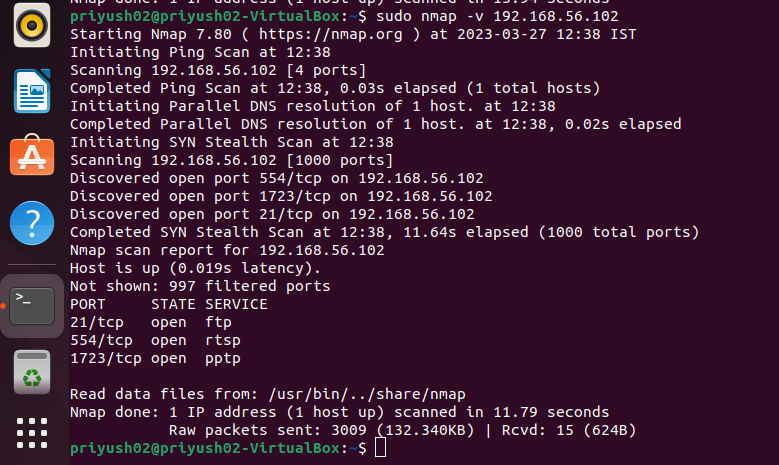
**🔹What is OS Fingerprinting?**

Operating System (OS) Fingerprinting is the process of analysing data packets which originate from a network in an attempt to glean intelligence to be used in later attacks. By detecting which operating system, a network operates on, hackers have an easier time targeting known vulnerabilities.



**🔹Ping:**

The Ping Scanner Tool sends ICMP ping packets to every IP address in any range of IPv4 addresses you specify. It looks for ICMP responses from active devices. This tool operates across any range of IPv4 addresses whether on your subnet or across the internet. It can also ping a list of IPv4 addresses that you need to ping. That list need not be contiguous, it can be random.



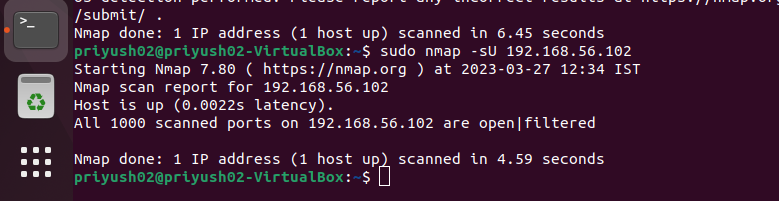
**🔹What is this TCP Port Scan**?

The TCP Port Scan will test an IP Address for common open ports. This technique of testing for listening services is known as a port scan. Try our advanced online port scanner that is able to scan any IP Address or IP range and all 65535 ports.



**🔹UDP scanning:**

UDP scanning is a process in which we scan for the UDP services that are being deployed on the target system or are currently in a running state. UDP is a connectionless protocol, hence it is hard to probe as compared to TCP.



**Conclusion:**

Network scanning provides a wealth of information about the target network, which is valuable regardless of whether you're trying to attack the network or protect it from attack. While performing a basic scan is a simple matter, the network scanners covered in this experiment provide a wide array of options to tweak your scan to achieve the best results. Nmap is used to detect IP spoofing and port scanning.

**DOP: / /2023 DOS: / /2023**

**Experiment No: 10**

**Aim**: Study of malicious software using different tools:

* 1. Keylogger attack using a keylogger tool.
  2. Simulate DOS attack using Hping or other tools
  3. Use the NESSUS/ISO Kali Linux tool to scan the network for vulnerabilities

**Theory:**

**🔹Keylogger:**

Keylogger attack using a keylogger tool. A keylogger is a tool that can record and report on a computer user's activity as they interact with a computer. The name is a short version of keystroke logger, and one of the main ways keyloggers keep track of you is by recording what you type as you type it. Some keyloggers go beyond just logging keystrokes and recording text and snoop in a number of other ways as well.

* It's possible for advanced keyloggers to:
* Log clipboard text, recording information that you cut and paste from other documents
* Track activity like opening folders, documents, and applications
* Take and record randomly timed screenshots
* Request the text value of certain on-screen controls, which can be useful for grabbing passwords.

🔹 **Keylogger Types:**

1**. API-Based Keyloggers** API-based keyloggers are by far the most common. These pieces of keylogging software use the keyboard API (short for application programming interface) to record your keystrokes. Each time you press a key, a notification is sent to the application you are typing in so that the typed character would appear on the screen. API-based keyloggers intercept these notifications and capture each of them as a separate event. The logs are then kept in a file on the system hard drive for easy retrieval by the hacker.

2. **Form Grabbing-Based Keyloggers** Rather than logging each keystroke separately, form grabbing-based keyloggers log the data from your web forms upon submission. Similar to API-based keyloggers, they intercept the submission notification to log all the information you have entered in the form. This can include your full name, address, email phone number, login credentials, or credit card info. The whole process takes place as soon as you hit the “Submit” or “Enter” button and is completed before your form data is submitted to the website.

3. **Kernel-Based Keyloggers** As the name suggests, kernel-based keyloggers inhibit the core of your computer’s operating system (also known as the kernel), which makes them very difficult to detect and remove. They hide inside your operating system and record your keystrokes as they pass through the kernel. Because they are more difficult to write, these keyloggers are rarer than other software -based varieties. They are distributed via rootkits, malicious software bundles that can bypass your computer’s kernel and target the hardware.

**4. Hardware Keyloggers-**Hardware keyloggers are devices that use the circuitry inside a keyboard to log keystrokes. They are most often built into the keyboard, although they are also available as either a USB connector (for personal computers) or a Mini-PCI card (for laptop computers). Rather than relying on software to store the logged keystrokes, all records are kept in the internal memory of the device. However, this also means that hackers must have physical access to the keyboard in order to retrieve this information.

**5. Acoustic Keyloggers-**Acoustic keyloggers are very complex and are therefore rarely used. They utilize the principles of acoustic cryptanalysis to record your keystrokes on the hardware level. No matter what keyboard you’re using, each key on it has a unique acoustic signature. The differences are subtle, but individual signatures can be determined by Analyzing a sample through a variety of statistical methods. However, not only is this very time-consuming but the results might not be as accurate as with other types of keyloggers.

**● How to Remove a Keylogger:**

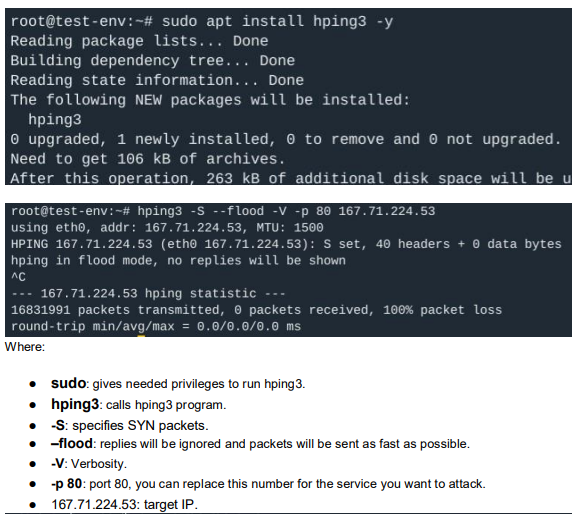
If you suspect that someone may have installed a keylogger on your computer but your antimalware software isn’t detecting anything, you may be able to find it in Windows Task Manager. Simply launch Task Manager and take a close look at the list of active processes to see if there’s anything out of the ordinary. You can also check your system’s firewall for any suspicious activity, such as unusual amounts of incoming and/or outgoing data.

**B) Simulate DOS attack using hping3 tool**

A denial of Service (DOS) attack is a very simple technique to deny accessibility to services (that’s why it is called a “denial of service” attack). This attack consists of overloading the target with oversized packets, or a big quantity of them. While this attack is very easy to execute, it does not compromise the information or privacy of the target. It is not a penetrative attack and only aims to prevent access to the target. By sending a quantity of packets, the target can’t handle attackers preventing the server from serving legitimate users.

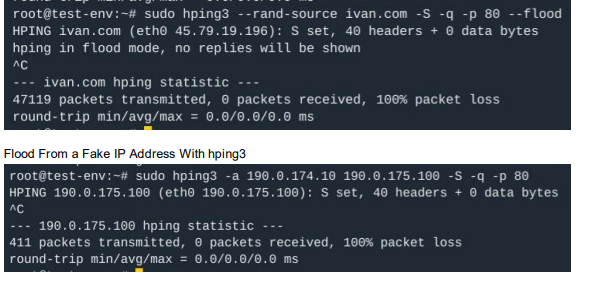
**hping3**

The hping3 tool allows you to send manipulated packets including size, quantity, and fragmentation of packets in order to overload the target and bypass or attack firewalls. Hping3 can be useful for security or capability testing purposes. By using it, you can test firewalls effectiveness and if a server can handle a big amount of connections. Below you will find instructions on how to use hping3 for security testing purposes.



The following example shows another possible SYN flood test for port 80.

Flood From a Fake IP Address With hping3



**C) Use NESSUS to scan the network for vulnerabilities.**

Nessus is a proprietary vulnerability scanner developed by Tenable, Inc. Nessus works by testing each port on a computer, determining what service it is running, and then testing this service to make sure there are no vulnerabilities in it that could be used by a hacker to carry out a malicious attack.

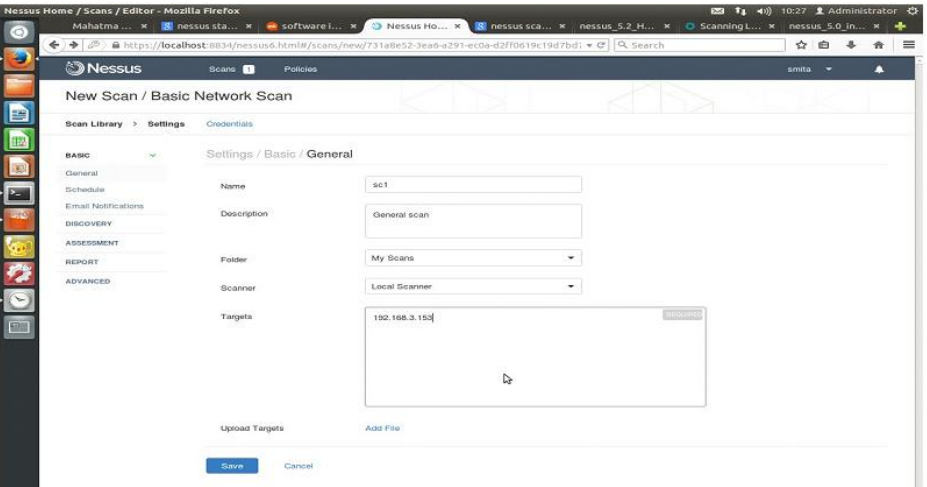
Nessus can scan these vulnerabilities and exposures:

* Vulnerabilities that could allow unauthorized control or access to sensitive data on a system
* Misconfiguration (e.g., open mail relay)
* Denials of service (Dos) vulnerabilities
* Default passwords, a few common passwords, and blank/absent passwords on some system accounts.
* Use NESSUS for scanning IP targets and checking network vulnerabilities

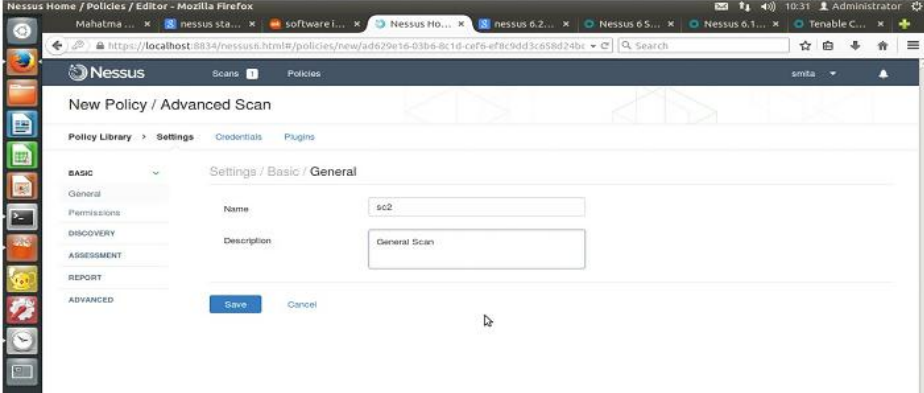
As we can see there are zero vulnerabilities on these two targets.

**Preparation for PCI DSS audits:**

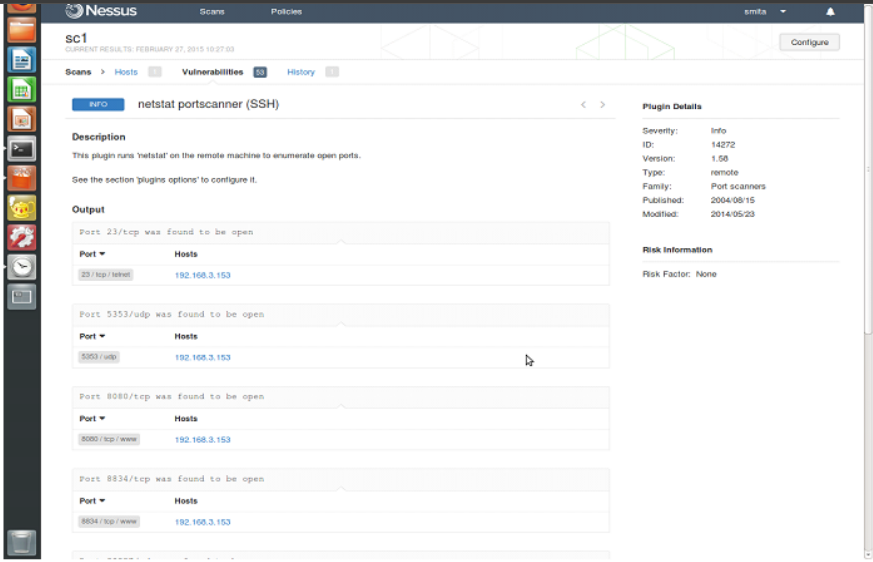
On UNIX (including Mac OS X), it consists of nessusd, the Nessus daemon, which does the scanning, and nessus, the client, which controls scans and presents the vulnerability results to the user. In typical operation, Nessus begins by doing a port scan with one of its four internal port scanners (or it can optionally use AmapM or Nmap) to determine which ports are open on the target and then tries various exploits on the open ports. The vulnerability tests, available as subscriptions, are written in NASL (Nessus Attack Scripting Language), a scripting language optimized for custom network interaction. Tenable Network Security produces several dozen new vulnerability checks (called plugins) each week, usually on a daily basis. files, compliance tests, additional vulnerability detection plugins).

**Basic Network scanning: **

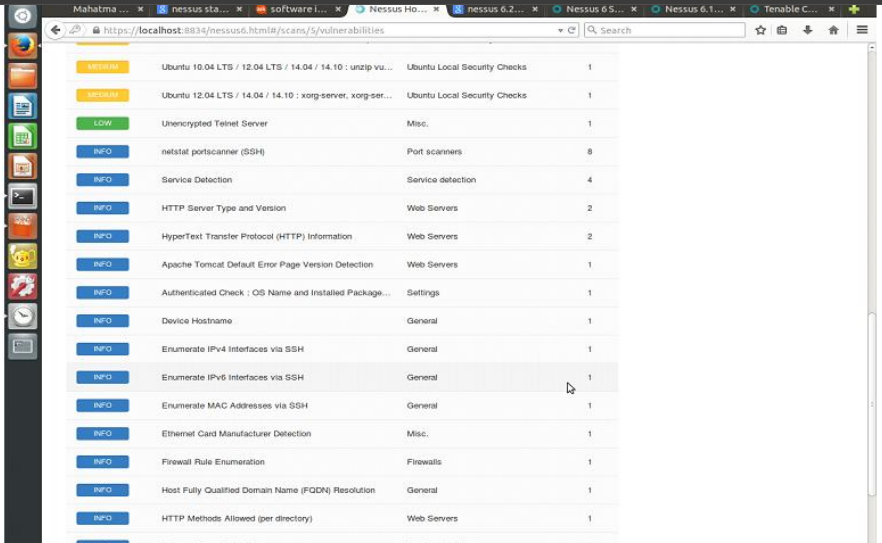
**Advanced scanning in general search:**



**Ntstat port scanning:**



**Vulnerability Mapping:**



**Conclusion:**

We understood what is keylogger, DOS attack & NESSUS.