SECURE DATA TRANSFER USING KEY EXCHANGE PROTOCOL BASED ON CLOUD

Project report submitted in partial fulfilment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

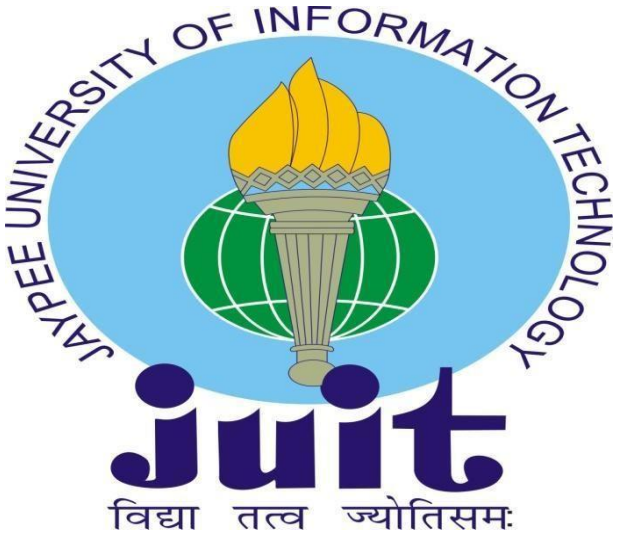
By

Mudit Mahajan (191318)

Under the supervision of

Dr. Pankaj Dhiman

to



Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

Candidate's Declaration

We hereby declare that the work presented in this report entitled **" Secure Data transfer Using Key Exchange protocol Based on Cloud"** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August 2015 to December 2015 under the supervision of Dr. Pankaj Dhiman (Assistant Professor (SG)).

We also authenticate that we have carried out the above-mentioned project work under the proficiency stream **Information Security.**
The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Mudit Mahajan (191318)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Pankaj Dhiman
Assistant Professor (SG)
Computer Science & Engineering Department
Dated:

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: …….

Type of Document (Tick): **PhD Thesis** **M.Tech Dissertation/ Report** **B.Tech Project Report** **Paper**

Name:_____ **Department:**_____ **Enrolment No** _____

Contact No._____ **E-mail.**_____

Name of the Supervisor: _____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _____

_____

_____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ..... (%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

<table>
<tr><td>Copy Received on</td><td rowspan="4">• All Preliminary Pages
• Bibliography/Images/Quotes
• 14 Words String</td><td rowspan="2"></td><td>Word Counts</td><td></td></tr>
<tr><td>Report Generated on</td><td>Character Counts</td><td></td></tr>
<tr><td rowspan="2"></td><td>Submission ID</td><td>Total Pages Scanned</td><td></td></tr>
<tr><td></td><td>File Size</td><td></td></tr>
</table>

Checked by
Name & Signature **Librarian**

……

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

ACKNOWLEDGEMENT

We owe our profound gratitude and indebtedness to our project supervisor **Dr. Pankaj Dhiman**, who took keen interest and guided us all along in our project work titled **—Secure Date Transfer Using Key Exchange Protocol Based on Cloud**, till the completion of our project by providing all the necessary information for developing the project. The project development helped us in research, and we got to know a lot of new things in our domain. We are really thankful to him.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

LIST OF FIGURES

LIST OF GRAPHS

LIST OF TABLES

ABSTRACT

The cloud is a very useful platform that is used to provide dynamic resources, which helps in virtualization and high availability. Since cloud computing rests on the internet, security issues like privacy, data security, confidentiality, and authentication are encountered. In order to get rid of the same, various encryption algorithms and mechanisms are used in different combinations.

In similar terms, we chose to use multilevel encryption with the help of hybrid cryptographic algorithms to enhance the security of data on the cloud.

We aimed at developing a web system which facilitates the text to be communicated to users securely by using Diffie-Hellman key exchange to exchange key to facilitate the exchange of the private key which is used to encrypt the data at the sender end and hence the same key will be used at the receiver end.

CHAPTER 1

INTRODUCTION

Information is currently one of the most valuable assets owned by businesses, organizations, and people. People have attempted to protect information stored on various types of storage since the dawn of time. Due to the availability of elastic, flexible, and on-demand storage and processing services for customers, cloud computing is quickly becoming a mainstream technology. A form of Internet-based computing known as "cloud computing" makes data and pooled processing resources available on demand to computers and other devices.

A particular IT environment known as a "cloud" was developed with the goal of remotely provisioning scalable and managed IT resources. The expression was initially used as a metaphor for the Internet, which is merely a networked infrastructure that enables remote access to a variety of dispersed IT resources.

This paradigm allows for ubiquitous, on-demand access to a pool of reconfigurable computing resources (including networks, servers, storage, applications, and services), which are easily deployed and released with little administrative labour. In an effort to reduce the cost of database upkeep, businesses are now converting from traditional databases to cloud computing. The transition from conventional methods to the cloud has its share of issues with data security and reliability.

Security is one of the key issues with cloud computing. In order to secure data transfer over the cloud, this project suggests using encryption algorithms like the Diffie-Hellman Key Exchange Algorithm. The project entails encrypting the cloud-stored file and utilizing Diffie-Hellman to authenticate the user before decrypting the necessary content in a secure and private manner.

Problem Statement

In this digital era, security of data has been a matter of concern for everyone as the amount of data is expanding day by day and hence the threat is also been increasing. Even the big tech giants are also concerned about the security of the data over the network as there is a huge chance of attacks where the data can be intercepted and sometimes can be modified.

Hence, to secure the data there must be some methods and hence cryptography plays a crucial role here which converts the data in a format which can only be interpreted when the system fulfils a certain criterion. This can be equally helpful for the security of the network, where the data is secured over a network to ensure secure communication of data to the sender and the receiver.

To ensure this we have implemented the Diffie-Hellman key exchange algorithm which ensures that the key which is used to securely encrypt the data and is securely transferred to the receiver using an asymmetric cryptographic technique.

Objectives

The main goal of this project is to investigate the significance of secure data transfer using key exchange protocol based on cloud as a component and requirement in today's world, as well as why it has become such a hot topic for discussion. Therefore, the study attempts to comprehend the associated risks and threats as well as why maintaining data security is of utmost importance.

The main aim or the goal of this project is to encrypt and decrypt the text using encryption techniques, a standalone programme will be created. It will be a GUI-based application that encrypts and decrypts text using the sender's key and the receiver's key.

The GUI makes it easier for users to use the application. The GUI for our programme will be created using the Python module: tkinter. The project includes the Local desktop application (GUI) and the Web Application that provides the facility of uploading, downloading of the files as well as the feature of encrypting the uploaded files. The web application hosted on AWS EC2 has been used for securely storing the data on the cloud. It will include user registration, generation of private keys, upload of files, download of other files stored on the cloud storage.

The goal is to fully understand and acquire accustomed to the technology being utilized to carry out the project so that we may utilize it to the fullest extent possible to accomplish the job. By using problem-solving techniques, collaborative research work, and collective learning, we hope to produce findings that can be applied to real-time initiatives in the future. Finding the real-world applications of the suggested discoveries is more important than just concentrating on the theoretical ideas, and that being our main goal.

Methodology

1. Diffie-Hellman Algorithm:

Assume two people, Alice (A) and Bob(B), want to use not secure method of communication to decide and come to an agreed “secret key”. This secret key then can be further used to do further encryption and decryption for a data and information transfer between them.

How to make that possible? The so-called Diffie-Hellman Algorithm method provides a way. This method is one of the ingredients of SSL and various other techniques, the encryption package that is part of most modern and secure web browsers. Diffie-Hellman creates a mechanism that makes it possible for two parties to communicate secretly while exchanging data over a public network.

By using the strength of cryptography rather than sending the plaintext, an analogy demonstrates the idea behind public key trade. With no prior communication between them other than Bob having trusted knowledge of Alice's public key, it is also possible to use Diffie-Hellman as a component of a public key structure, enabling Bob to encrypt a communication so that only Alice will be able to decipher it.

Only the knowledge of each other’s public key is enough to securely transfer data between the two participants with the help of the Diffie-Hellman Key Exchange Protocol Algorithm

Alice's public key is **(g^a % p, g, p)**.

To shoot her a communication, Bob chooses an arbitrary **'b'** and also sends Alice **(g^b % p)** together with the communication translated with a symmetric key **((g^a)^b % p)**.

Due to the fact that only Alice has “a,” she can only determine the symmetric key and subsequently decipher the communication.

Man-in-the-middle attacks are also avoided by using a pre-shared public key.

What Alice did:

$$B^{a} \bmod p = (g^{b} \bmod p)^{a} \bmod p = g^{ba} \bmod p$$

What Bob did:

$$A^{b} \bmod p = (g^{a} \bmod p)^{b} \bmod p = g^{ab} \bmod p$$

Figure 1 Diffie-Hellman Keys

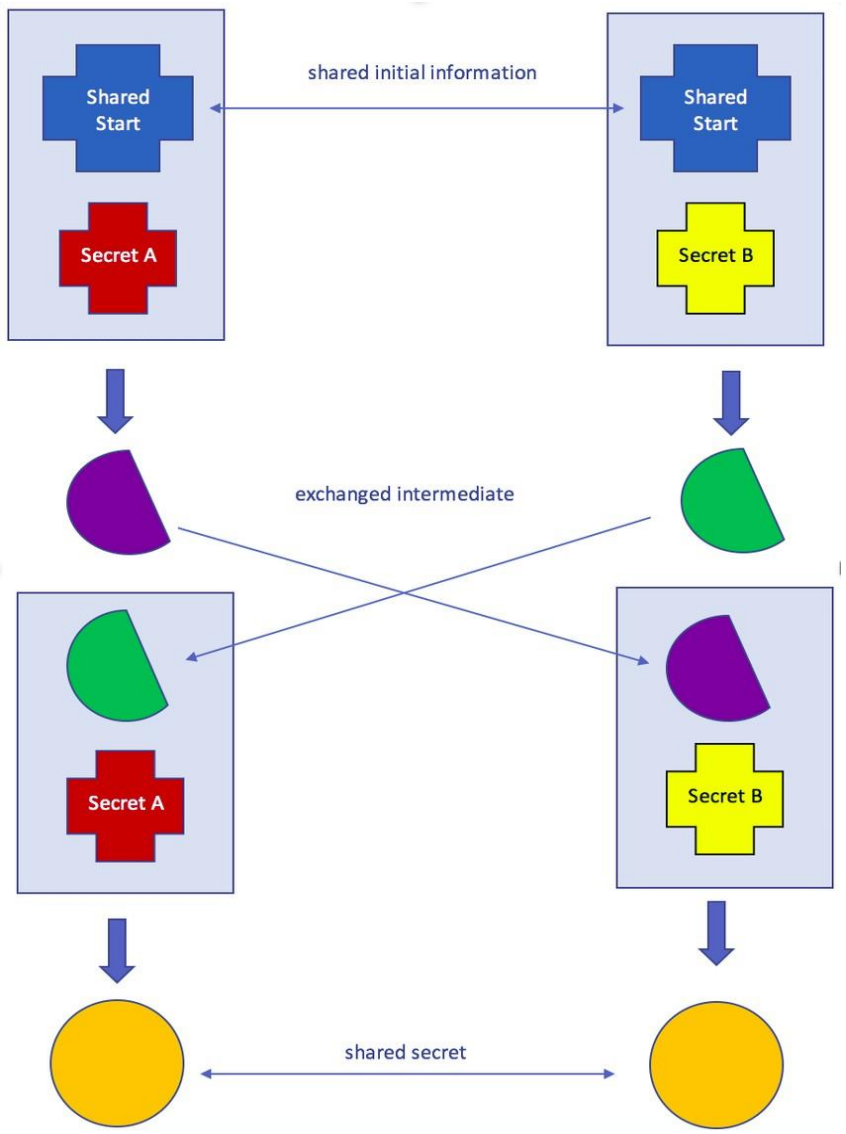


Figure 2 Diffie-Hellman Algorithm

2. Tkinter (Python Package):

The default Python interface for the **Tcl/Tk** GUI toolkit is the Tkinter package (sometimes known as **"Tk interface"**). Most Unix operating systems, including macOS, as well as Windows systems support Tk and Tkinter. When you type **"python - m tkinter"** on the command line, a window displaying a basic Tk interface should open, confirming that Tkinter has been correctly installed on your system and identifying the Tcl/ Tk interpretation that is installed, allowing you to read the Tcl/ Tk attestation for that interpretation.

A variety of Tcl/Tk performances are supported by Tkinter, pitched with or without thread support. The authorized Python double release packets support threaded Tcl and Tk8.6.
For more details about supporting performances, refer to the Tkinter module's source code. To make the experience more pythonic, Tkinter doesn't just add a thin wrapper; it also adds a considerable amount of its own sense. This attestation will focus on these changes and additions, and refer to the authorized Tcl/Tk attestation for information that hasn't changed.

The following notions make up the Tkinter's core fundamental implications:

Widgets

Individual widgets make up a Tkinter interface. Every widget is displayed as a Python object, made up of elements from the ttk.Frame, ttk.Label, and ttk.Button classes.

Hierarchy of widget

A scale is used to position widgets. The frame, which was housed inside the root window, contained the marker and button. When making each widget, including its children and parent widgets. It is supplied to the widget function Object() { [native code] } as the first argument.

Different options for configuration

Similar to the textbook's display of an appearance or button, widgets can be configured to change how they look and function. There are many sets of settings available for various classes of widgets.

Relative positioning or geometry management

Every time a new widget is created, it must be manually added to the user console. The defined objects' placement in relation to the reference to the interface is controlled by a grid resembling that of a figure director.

Loop for event

Tkinter responds to the user's manual input, changes made by your software, and actually updates the display only when an event loop is vigorously executed. However, if your software isn't executing the event circle as a requirement for Tkinter, your user interface won't upgrade. Tkinter attempts to bridge the gap between the very dissimilar threading models of Python and Tcl/Tk. You might need to worry about this though, as we'll need to understand this idea if we're going to use threads.

There may be several threads connected to a Python interpreter. Tcl allows for the creation of numerous threads, however each thread is coupled with a unique Tcl interpreter case. Although each interpreter case can only be used by the one thread that produced it, threads can produce more than one interpreter case. A Tcl interpreter is included in each Tk object that the Tkinter generates. Additionally, it records which thread developed that interpreter.

While one method of accessing Tkinter is internally from the Python thread, if a call originates from a thread other than the specific thread that created the Tk object, an event is logged to the Tkinter event log, the event line of the interpreter, and after execution, the calling Python thread receives the outcome. The majority of Tcl/Tk activities are event-driven, which means that following startup, the interpreter runs an event circle (i.e., Tk.mainloop()) and responds to events.

Event preceptors must react carefully because the system is single-threaded, or else they will prevent other events from being reused.

Any lengthy computations should not be performed in an event tutor; instead, they should be divided into smaller chunks using timekeepers or performed on a separate thread. Contrary to many GUI toolkits, this one runs all operation law, including event instructors, in a completely distinct thread. However, any Tkinter calls made from apparatus other than that which is executing the Tcl interpreter will be unsuccessful. If the Tcl expert isn't handling events and operating the event circle.

A number of special cases like:

It is possible to build Tcl/Tk libraries that aren't thread-aware. In this instance, Tkinter makes a call to the library from the newly generated Python thread, even though this thread is distinct from the one that created the Tcl interpreter. A global touch makes sure that only one call is active at once.

While Tkinter enables you to construct several instances of a Tk object (each with its own interpreter), all interpreters that are a part of the same thread share a common event line, which quickly becomes ugly.

Produce no more than one case of Tk at a time in actuality. Otherwise, it's fashionable to create them in separate threads and make sure you're using a Tcl/Tk figure that is thread-aware.

It's not the only technique to prevent the Tcl interpreter from reinitializing the event circle: you can also block event Instructors. It is possible to run numerous nested event rings or to stop using them altogether. However, if you're doing anything risky in regard to events or garb, be cautious of these possibilities.

Many specific Tkinter functions can only be used from the thread that produced the Tcl practitioner at this time.

2. AWS (Amazon Web Services):

AWS (Amazon Web Services) offers a wide range of on-demand, pay-as-you-go cloud-based goods and services to customers all over the world, including information security, data storage warehouses, databases, data analytics, data science, computer networking, mobile, inventor tools, operation tools, IoT, security, servers to host, and enterprise-grade applications.

From data storage and databases to deployment services or CI/CD tools, directories to content delivery, constant delivery networks (CDN), over 250 services are up and running and available for use by the public.

Without the overt fixed expense, new services can be provisioned easily. This makes it possible for businesses, start-ups, small and medium-sized businesses, and visitors from the public sector to get through the barriers that prevent them from quickly adapting to shifting market conditions.

One can swiftly spin up resources such as databases or computation power ranging from custom servers to machines used for machine learning, deep learning or web hosting. They inherently come up with proper security baked into to ensure the CIA triad of Information Security. AWS includes services and products to help in Internet of Things (IoT) and much more.

Technology has made it possible to deploy ideas and products swiftly, moving from the ideation stage to execution of an idea several orders of magnitude faster than in the past.
As a result, one is free to experiment, test novel ideas on various user bases, and quickly pivot or create a business. With cloud computing, one does not have to order more than necessary resources up-front to handle peak situations or heavy traffic of business activity in the future.

Instead, one can request the amount of resources and services that one actually needs at the moment. Then one can estimate the amount of resources required during peak hours and increase or decrease the amount of resources according to the research conducted.

The cloud gives one easy and quick access to a wide range of technologies so that one can develop products swiftly and make nearly everything that one can imagine without worrying about the availability of resources.

With the use of cloud computing technologies, you may exchange set fees (such those for data warehouses and physical waiters) for variable fees so that you only pay for the resources and services you really utilize. Additionally, due of scale, the variable costs are far less than what you would spend to do it yourself.

With the cloud computing, you can expand to new geographic regions and deploy all over the world in a matter of minutes. For example, Amazon Web Services (AWS) has data storage centres in 5 continents all over the world, so you can deploy your operation and business in multiple physical locations with just a matter of clicks in a limited span of minutes. Putting operations in near proximity to the user base helps in reducing the cost of doing the business and provide a better experience to the users as well.

Information as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service are the three main categories of cloud computing. Each type of cloud computing provides different situations and methods of technologies, control, flexibility, and operation so that you can select and choose the right and the best suited or required set of services and resources for your requirements and needs.

Information as a Service (IaaS)

The foundational elements of cloud information technology are contained in information as a services (IaaS). In general, it gives users access to networking capabilities, computers (virtual or on dedicated hardware), and servers that hold data.
IaaS offers you the greatest and best degree of operational flexibility and control over your computing resources. It's almost like being IT resources and services, which are known to many IT departments and inventors

Platform as a Service (PaaS)

PaaS frees you from having to manage the underlying framework and hardware (it typically deals with the issue of physical resources like hardware and software issues like operating systems and needed computation power), allowing you to focus on the creation, deployment, and management of your company. Being less concerned with resource acquisition, capacity planning, software conservation, platform doctoring, or any other first phase planning of acquisition of resources, services, or platforms allows you to be more successful.

Software as a Services (SaaS)

With SaaS, the service provider offers you a whole product or service that is run and controlled by them. In most cases, people pertaining to SaaS are pertaining to end-user operations (similar to web hosting applications). You do not need to be concerned about how the service is administered or how the underlying structure is maintained if a SaaS service is offered. You just need to consider how you'll apply to that specific programme.

AWS offers more services than any other cloud provider, including computing power, data storage facilities, databases, networking, data lakes and analytics, machine learning and artificial intelligence, Internet of Things (IoT), security, and many other features. With the help of the AWS platform, you can develop or build more quickly thanks to the extensive portfolio of services that AWS offers. Each AWS Solution includes a thorough architecture, deployment guidelines, and instructions for both automated and manual deployment.

3. Flask :

Micro web framework Flask has documentation written in Python. Due to the fact that it doesn't require any particular tools or libraries, it is referred to as a micro-framework. There is no database, abstraction layer, form confirmation, or any other factors where-existing third-party libraries give common functions.

Even said, Flask offers extensions that can provide functionality as if it were a built-in feature. Object-relational mappers, form validation and confirmation, upload execution,

colourful open authentication and validation technologies, and a number of widely used frame-related utilities all have extensions live.

A Python module called Flask acts as a web frame and enables fluid web operations development. It is a micro-framework without an ORM (Object Relational Manager) or other features, with a compact and simple-to-extend core. It does offer a lot of great features like a template engine and URL routing. It is also a WSGI web app frame. The Werkzeg WSGI toolkit and the Jinja2 templating engine serve as the foundation for Flask. They are both Pocco systems.

WSGI

For the creation of Python web applications, the Web Server Gateway Interface (WSGI) has often been used as a starting point or a standard.

By handling the lower level operations and leaving the higher level code to the developer, WSGI serves as a programming interface between the web servers and the web applications at the lowest levels and speeds up the development of web applications.

Werkzeug

A WSGI toolkit called Werkzeug implements web requests, the objects obtained from the server's response, and other crucial utility functions. This makes it possible to build a web framework on top of it. A root base for the Flask micro-framework is Werkzeg.

Jinja2

A well-liked Python template engine is jinja2. A web template framework renders a dynamic web application page by combining a template with a specific data point.

Microframework

The term “micro-framework” is commonly used to describe Flask. The goal is to maintain scalability and simplicity at the heart of the web application. Instead of a database

mounting abstraction layer, Flask enables enhancements that provide related functionality to the web application. Flask, in contrast to the Django framework, is mostly Pythonic. Due to Flask's low learning curve, getting started with it is very simple. Despite being a micro-framework, your entire programme need not be contained in a single Python file. For large-scale programs, you can and should use a lot of files to navigate the added complexity.

The Flask framework is basic yet expandable since it is “micro.” All crucial technical decisions, such as which database to use and if an ORM is required, are yours to decide; Flask does not restrict your options in any way. One of the most widely used web frameworks is Flask, which suggests that it is modern and cutting-edge. You may easily increase its functionality. It is simple to scale up for sophisticated web applications.

4. Python Crypto :

Data confidentiality and encryption methods are included in the Python Crypto package.

Three types of encryption algorithms exist:

Symmetric ciphers

The same key is used by all parties to encrypt and decrypt data. In general, symmetric cyphers are almost always quick and can compute a lot of data.

Asymmetric ciphers:

Receivers and senders use different keys. Data is encrypted by senders using public keys (which are not secret), and is decrypted by recipients using private keys. Asymmetric ciphers can only compute a very small amount of data and are typically much slower than symmetric ciphers.

Hybrid ciphers:

The two cipher types mentioned above can be merged to create a structure that combines their best qualities. A symmetric cipher encrypts the original data while using the short-lived symmetric key secured by an asymmetric cipher.

Symmetric ciphers

Two types of symmetric cyphers exist.

Stream ciphers:
The most logically simple cyphers, which can only encrypt one byte at a time.
Examples include ChaCha20, XChaCha20, and Salsa20.

Block ciphers:
The cyphers that can only be used with a certain volume of data. AES is the most widely used and well-known block cipher, and its blocks are 128 bits in size. In general, a block cipher is substantially applicable only together with a mode of operation, which allows one to encrypt a variable or dynamic amount of data. Some modes (like **CTR**) effectively convert a block cipher into a stream cipher.

It is widely agreed that cyphers that just offer secrecy and no kind of authentication should be avoided. Instead, asymmetric encryption and authentication have been combined using primitive encryption methods (MAC). For instance

1. Modern block cypher operating procedures (like GCM).
2. Stream cyphers like ChaCha20- Poly1305 and XChaCha20- Poly1305 that are combined with a MAC technique.

Legacy ciphers

A numerous amount of ciphers are developed in the Python Crypto package purely for backward compatibility use cases. They're disapproved or indeed completely broken and shouldn't be utilized in upcoming projects.

- Single DES and Triple DES

- RC2 (block cipher)
- ARC4 (sluice cipher)
- Blowfish (block cipher)
- CAST- 128 (block cipher)
- PKCS# 1 v1.5 encryption (RSA)

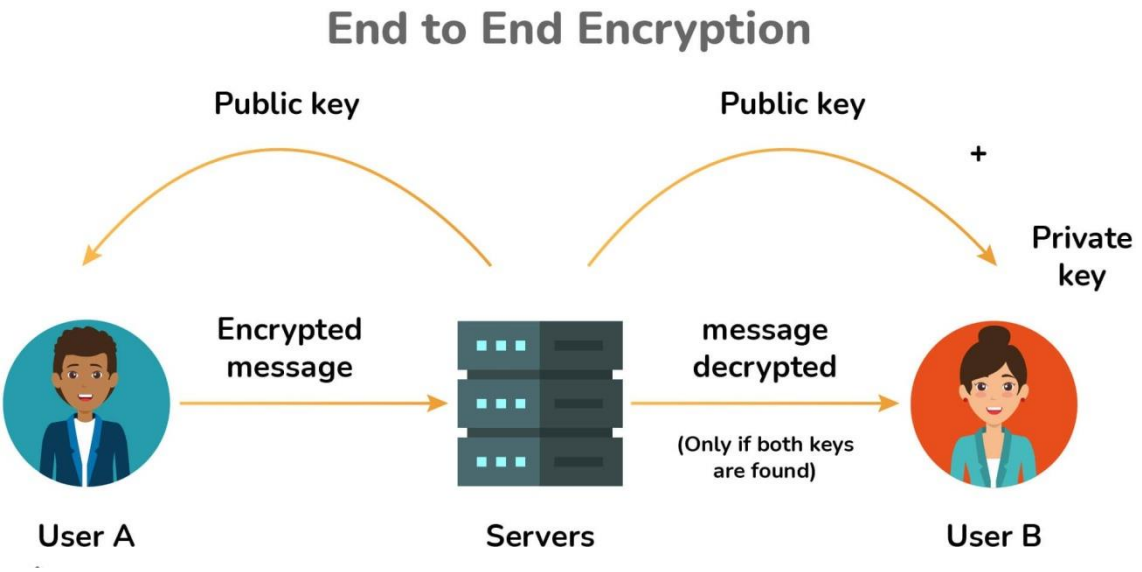


Figure 3 Encryption between two users

5. Prime Number :

Any natural number higher than 1 that cannot be created by multiplying two smaller natural numbers is known as a prime number (or a prime). The choice of a prime integer is the single user-defined pre-existing parameter in the Diffie-Hellman protocol. The prime number p ought to be big enough to fight off any known threats. NFS (attack on the network file system), which has been used against numbers on the order of 2768 is the most effective attack.

It would seem prudent to choose a p that is far larger than that; at the very least, it should be approximately 1024 bits, and ideally, it should be at least 1536 bits. Another characteristic of p is that its factorization should be known, and that p1 should have a big prime factor q. We're probably safe if we decide a random prime p and a random generator

g, but we won't be sure (and we might leak a few bits of the private exponent if the order of your random g happens to have some small factors).

CHAPTER 2

LITERATURE SURVEY

The easiest to use and fastest growing platform for resource sharing and pay-per-use services is the cloud. The articles that follow clarify the goals of cloud computing and outline its advantages and difficulties. Security has emerged as the main concern for the shift to the cloud in the analysis that follows, so we look for various cryptographic encryption algorithms to analyse the best single level encryption algorithms. Users have the option to store and reuse their data in third-party data centres and use it conveniently thanks to cloud computing and data storage over the cloud. Many different service models (with acronyms similar to SaaS, PaaS, and IaaS) and deployment models are used by businesses when using the cloud service.

Security concerns related to cloud computing are typically divided into two categories: security concerns experienced by cloud-based service providers (organisations providing software, platforms, or structures as a service utilising the cloud) and security concerns experienced by their clients (companies or associations who host operations or store data on the cloud). In a cloud provider's participated security responsibility model participated responsibility model, the responsibility is still participated and regularly explained.

While the user of the service must take steps to fortify their operation and use strong watchwords and authentication measures, the provider must ensure that their structure is secure and that the data and operations of their guests are protected. An association loses the capacity to have direct physical access to the servers holding its information when it decides to store data or host processes on the public cloud storage.
As a result, bigwig attacks pose a risk to potentially sensitive data. One of the top seven cloud computing hazards is bigwig assaults, according to a 2010 research by the Cloud Security Alliance.

As a result, cloud service providers must make sure that rigorous background checks are carried out on any employees who have direct access to the data centre's servers.
Additionally, it is advised that data centres be continuously watched for suspicious activity. Cloud service providers frequently store more than one client's data on the same server to

preserve user data, reduce costs, and maintain effectiveness. As a result, there's a possibility that other potential users could see a user's private information.

Cloud service providers should guarantee sufficient data insulation and logical storage separation to manage comparable delicate scenarios.

For visitors or various users of a public cloud service, the dependent utilisation of virtualization in enforcing cloud structure presents distinctive security enterprises. The interaction between the operating system and supporting hardware, such as computing, storage, or networking, is changed by virtualization. Virtualization is a new subcaste introduced as a result, and it needs to be properly setup, managed, and secured. The possibility of compromising the virtualization software, or "hypervisor," is present in some businesses.

Although these businesses are primarily hypothetical, they actually exist. A breach in the director workstation's operating system for the virtualization software, for instance, could cause the entire data centre to go offline or to be reconfigured to a bushwhacker's delight.

Security:

The risk of a security breach in the form of data mishandling and mismanagement cannot be ignored in the cloud computing paradigm, where the same resource or data is used by a variety of users, whether individual or organizations. Therefore, data repositories must be safeguarded in order to eliminate this danger. Additionally, channel security may guarantee the security of data throughout operations, storage, and transmission. To increase the security of the data rendered on the cloud, it is necessary to implement security concepts of authorization, authentication, and access control.

Title: "Secure User Data in Cloud Computing Using Encryption Algorithms" (2013)

In addition to providing an illustrative presentation of several cryptographic security methods including DES, RSA, Blowfish, and AES, the authors of this paper also provide

us a quick overview of the many cloud computing concerns and challenges. They contrast these algorithms' scalability, security, data encryption on capacity, and execution speed.

Security Issues and Challenges of Cloud Computing:

Security continues to be one of the most important factors in the world of computing, and with the development of cloud technologies, it has gained ground because of the importance of the data provided on the cloud. In order to ensure user trust, it is essential to analyse the platform on privacy and security criteria because cloud computing, in its initial form, uses a range of novel and insecure technologies.

Title: "Analysis of Security Algorithm in Cloud Computing" (2014)

The writers of this study provide us a quick overview of cloud computing's definition and features as well as some of its benefits. It also presents us with a number of cloud computing problems. The paper also discusses various cloud computing data security concerns. AES, Blowfish, DES, RC5, and 3-DES are just a few of the encryption algorithms that are compared in the paper.

Title: "Advantages and challenges of adopting cloud computing from an enterprise perspective" (2014)

Cloud computing has emerged as a cutting-edge, quick, and simple technology with the development of technology, the expansion of internet penetration, and its greater adoption.

It is becoming a good option for businesses because it doesn't require significant capital investment in the form of resource purchase and provisioning, but this process is still in its infancy. Computing as we know it today exhibits a paradox: on the one hand, computers continue to grow exponentially more powerful, and on the other, the cost of computing per unit is falling quickly, to the point where computing power per second is now largely regarded as a commodity. Contrarily, as computing becomes the industry's backbone, the

complexity of resource, protocol, standard, and infrastructure management has increased, making the organization's management of computing more expensive.

Title: "A Review of Cryptographic Algorithms in Network Security"

Computers that are connected to a network are subject to numerous threats from hackers. They should be given the information that travels through the network since they have an impact on data transmission over the mechanisms. This system is known as cryptography. Data may be delivered via cryptography in an undetectable manner. This is unreadable by an outsider. Only the sender of the message and the intended recipient, using the sender's provided key, can comprehend the message.

Encryption and decryption are the two ideas of cryptography. Using cypher text, encryption converts plain text into an unintelligible format. The recipient will receive both the key and the cypher text. The key is applied to the cypher text at the receiving end, where the actual information is obtained.

Types of Cryptography:

- **Symmetric Key Cryptography:** if a single key is employed for both encryption and decryption. This key is a secret one. This category includes the algorithms DES, TDES, AES, and RC4.

- **Asymmetric Key Cryptography:** If two keys—one for encryption and the other for decryption—are used. This category includes the algorithms RSA, MD5, and ECC.

- **Existing Algorithms:**

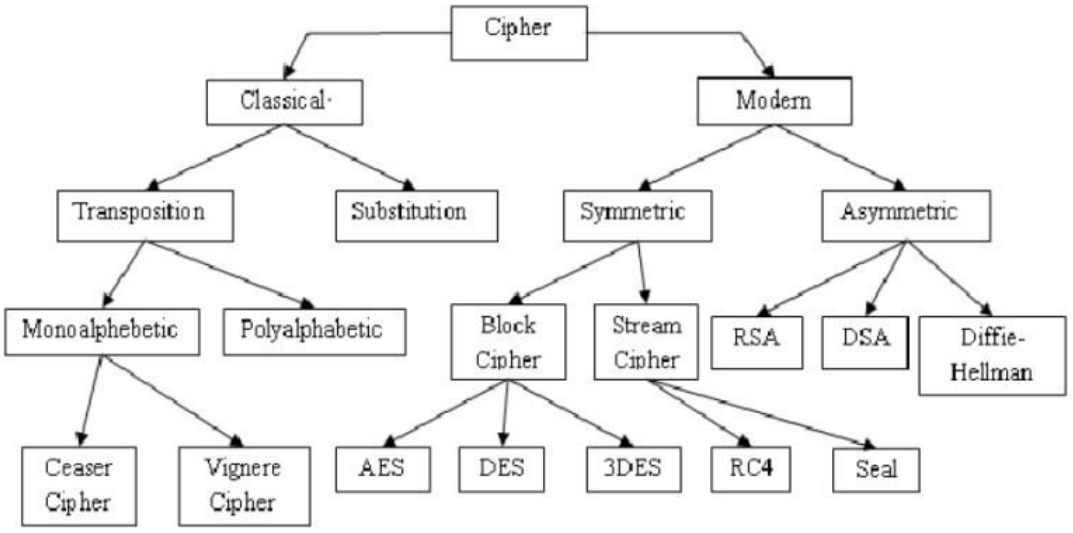


Figure 4 : Classification of Cipher

- **DES:** DES was created in the 1970s and makes use of the Feistel Structure. The DES block cypher technique is symmetric and use a single key for both encryption and decryption. Therefore, the private key must be known by both the sender and the recipient. Eight bits of the key's 64-bit length are reserved for parity checks. To encrypt a message, 16 rounds of permutation are used. The only difference between the encryption and decryption processes is that the latter is carried out in the opposite order. A brute-force attack is one method of possible DES attack. Additionally, the DES algorithm is vulnerable to three quick attacks. Which are:
    - Multiple Cryptanalysis
    - Using Linear Cryptanalysis
    - Attack by Davies

DES is considered as less secure, and this algorithm is not used much since this has been broken very easily.

- **Triple DES**: The next level of DES, known as Triple DES, was created to counteract the attacks that DES could withstand. It performs three DES processing steps to increase security. The TDES method requires 48 rounds and uses a key that is 168 bits long. The first text is encrypted by applying this longer key to each block. The TDES is another name for it (Triple DES). There are three different ways to key. The strongest keying choice is 1, and each of the three keys—K1, K2,

and K3—is independent. The two keys K1 and K2 are separate in keying option 2, however the three keys K1, K2, and K3 are the same in keying option 3.

- **AES**: AES is a symmetric and block encryption algorithm, which helps it overcome the flaw in the DES method. AES was created in 1977 and is officially known as Rijindeal. The key sizes are 128 (10 rounds), 192 (12 rounds), and 256 bits, with a 128 bit block size (14 rounds). [10] Alternate bytes, redirect rows, rearrange columns, and add round keys are the four stages of the AES permutation process..
    - Substitution bytes - In this stage, the Rijindeal S-Box is used to replace each of the matrix's bytes (ai, j). The sub-bytes are reversed at the back of decryption to obtain the initial state.
    - Shift Rows: Using a specific constraint, this operation shifts each row. The first row of the matrix is kept the same, while the second, third, and fourth rows are shifted to the left by one row.
    - Columns are multiplied by a fixed polynomial in this step, and the new value of each column is then entered. d) Add Round Key: The round key is added in this step by applying XOR to the matrix and deriving it from the main key.

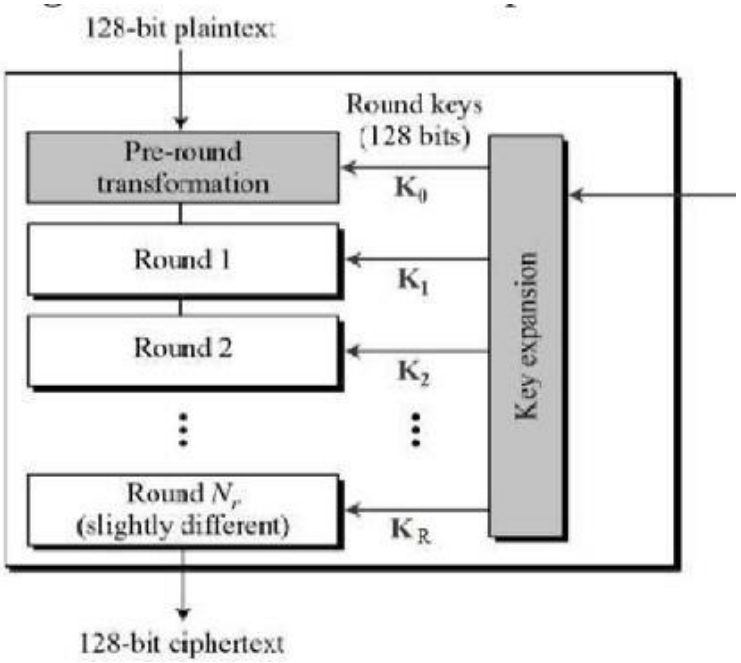
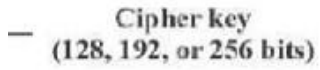


Figure 5 : Rounds of AES



R	Key size
10	128
12	192
14	256

Relationship between number of rounds(R) and cipher key size

Table No 1 : Relation between no of rounds and cipher key

- **RSA** :A public-key encryption technique using two keys—the private and public keys—is the RSA algorithm. Block cypher encryption is used, and the key length is 25 bits out of 1024. Public and private keys are generated via RSA using two prime integers. You should pick these two prime integers at random. The following are possible RSA attacks:
    - It is simple to break the exponent of a small number.
    - The chosen-cipher text can also be decrypted if more receivers receive encrypted messages using the same exponential.

- **RC4** : Ron Rivest created RC4, also known as Rivest Cipher 4. Here, the plain text is encrypted using the stream cypher. The RC4 technique generates a pseudorandom stream of bits (key stream), and bit-wise encryption and decryption has been carried out. The two steps of the creation key system are the permutation of all 256 bytes and the use of two 8-bit index-pointers. This RC4's key length ranges from 40 to 128 bits. Bit-flapping attacks are possible if the common block cyphers are not used MAC strongly, and stream cypher attacks are also susceptible if they are not implemented properly.

- **MD5** : A common cryptographic hash function is the MD5 Message Digest method. The MD2 and MD4 predecessors of the MD5 standard are followed by the MD6 standard. For encryption and decryption, MD5 uses a hash function of 128 bits, or 16 bytes. In the software industry, MD5 is used to ensure that no intruders are encountered when downloading data. In order to verify the security of the downloaded file, the user can compare the MD5 checksum provided by the file servers with the checksum of the downloaded file.

- **SHA** : The cryptographic hash functions known as SHA include SHA-0, SHA-1, SHA-2, and SHA-3. The hashing algorithms are the most extensively used and trusted for security in many applications. The hash function is used to provide the hash table an index. NIST developed it, and it was published in 1993. Additionally, the block size (160 bits) and rounds for SHA-0 and SHA-1 are same (80). Different block sizes for the SHA-2 algorithm are designated as SHA-224, SHA-256, SHA-384, and SHA-512. The SHA-3 also has various block sizes, which are denoted as SHA3-224, SHA3-256, SHA3-984, and SHA3-512 respectively.

- **ECC**: A pair of keys—one a public key and the other a private key—make up this public-key cryptography scheme. The private-key is a user-selected random integer, while the public-key is a specific coordinate (x,y) in the curve. Shorter key lengths, minimal CPU and memory utilisation, and shorter key lengths are all benefits of the ECC algorithm.

CHAPTER 3

SYSTEM DEVELOPMENT

Software Requirements

- Python
- Python Crypto Package
- Flask Web Development Micro-framework
- VS Code
- PyCharm Community Edition
- AWS (Amazon Web Services)
- Git / GitHub

Hardware Requirements

- Random Access Memory (RAM): 4 GB or above
- Central Processing Unit (CPU): 2.4 GHz Processor and above
- Operating System (OS): GNU/Linux (Ubuntu & Manjaro)

Design

The main goal of this project was to offer a cloud-based file storage solution that was as secure as possible. So, a number of issues needed to be resolved, including:

- Where is it necessary to encrypt the file?
- In what way must the user be verified?
- What is the password for AES encryption?

What connection does this key have to the Diffie-Hellman key exchange protocol?

The assault man in the middle forced us to abandon our first plan of online text encryption. About an attack known as NFS, we also learned that. NFS in an attack that may compute a key of order 2768. This included around 232 digits. As a result, we deduced that a larger prime number is required for the process. As a result, we employed a prime number with 600 digits.

We arrived at this course of action after analysing each of these questions. We used an application to encrypt the file directly on the owner's computer to give them more control. Only users with access to the final same key generated by the Diffie-Hellman algorithm would be able to decrypt the file. The final Diffie-Hellman key, which was the same for both intended participants, served as the foundation for the AES encryption key.

There were three main ways to execute this module. The tasks were

- Create a new user's private key with the specified length.
- Create a user's public key using his private key.
- Using the public and private keys that are provided, create a secret key.

A prime number was first required. We already discussed the NFS attack in detail previously, as well as the need for a strong prime number. Thus, we hard coded the prime number :

FFFFFFFFFFFFFFFFFFD3DDE89DD089C0617E69B088E65F18CFD4965672E
0503E9AA9E7765744E855974E620CF6AD0919CB283EE5DE200A9887DC6C18
4B834900429F978BE4F640953862AFCE13CB63FAF37D47EE7DB015D1B1AE7
DED59184BA9FD15864805CC13F461CD5520A998E795463A6BA470EE013FD
D28F44F38F31A2D66A872A3B5FC3B260A3DDB5DE303386978E780C2AF7B4
D9AFDD5838F2688CBCB4B603AEC646CF37437C0F32A18A75BE3170C7304A
40E222229DA848CFEB8BDF3ED256D93EBC61F832879C1FB2474B3243865F4
EA5C936BAA366424AB6396F6752DE74CFA8FB7AF147003DD9D86B238CB3A
B139B370D4CEB8DC8FD239A6BC777834E5EEF21F6293A479511EBD6E19F4
96E6D4747B0B5D2FB21EE0684F366F8E3408640C3E67D8EF78323692FE2CE
9C65ADC55D313A1BCC91BE0FFBF1A8D3AA055F67AD5C7BB85D1A60823EB
92FEE51797A0E7308F02DE5CC418D1E84FBE7D6F8CCAA4514B22B704DB45
CA585FF7B54D1F0EB88CC63E7A79A323F8E7CEBF79C7E95E9C1B00913048
A5B28C08964F207E55D6FEF47E6EC7D284F30D1271D51AE230FCE29B46C1
0AC3895FB2DC873104F8BF624FA7DB215150233B452CC6E7441F7BDB1BB4
DECABC6EC44DE41B9FCBEBDE3AD8198A5680B727971138D7209DE35A60B
EA100314F283BD5EE035E9DA6049FF8FDB73ABD367E1579CED6462062BCD
4D1C6207C72D4BAF3C63294820439C534FD4F80C6A09CD7BFF681CDDF8D8

89AE4B399AA50B5D7219600DC184071267DDB18BFA7D2EEC079216F1DE7E
B515681AB3322A3C090AF4692B99D50CB6E47495C7826AACEBF141CFE8E2
9FE8ED40BD2CBBBD8EC44DA2AC9E3852ADB0516B438649A1C32E4FA6812
3C9962B5ABDB01A917887D6E787A21C327A11080129A021D28B4E801DF0EC
FB5BD3413BA5E470AF42E802E649DAB0C889077C6D516A775711EBBC002B
77181B2F12546A68CE33720678D4680A89D60AFF21FB6EE2DA1622D3EECD9
1652A40E49C8E17D3390BDC8EA5FBA7C4E1E6A58F0793BD7C060D57517AE
A8A0FEBD854058BFCE46ABC1FDBA12558A33A70540D07133DAD24CAAA8A
5E827510150AF8981622D515EA659AEC7945993817185596FBCB2ED9C25C4
F60FD55C5BF82A70CE2A3872B93068E081C277E93EB3EC63E264E50923C71
281AC80C6471F4089CBA4E453C076D669690770925DE9BB255802653F26C16
9DA3ACD32D55638F5FC42DF8AF36196A93D55C16384AD8950FB361A8BC70
02CD3B54ECE156682946EF1B4C71142F9EA5AF998A5BFB683EEDE7B604F6
BC5FFB0B6DE736A9E24C44F6CE7E526675B584E542C15D6D6531EF47341F5
2FD6A0B203B134A3DC3B9159FEDD4043E89780A41522B931B36AEBB02047C
C76A880E420921DC1CD08B8266C4C432C86122AADF09CFFFFFFFFFFFFFFF
Fx00AC3895FB2DC873104F8BF624FA7DB215150233B452CC6E7441F7BDB1B
B4DECABC6EC44DE41B9FCBEBDE3AD8198A5680B727971138D7209DE35A6
0BEA100314F283BD5EE035E9DA6049FF8FDB73ABD367E1579CED6462062B
CD4D1C6207C72D4BAF3C63294820439C534FD4F80C6A09CD7BFF681CDDF
8D889AE4B399AA50B5D7219600DC184071267DDB18BFA7D2EEC079216F1D
E7EB515681AB3322A3C090AF4692B99D50CB6E47495C7826AACEBF141CFE
8E29FE8ED40BD2CBBBD8EC44DA2AC9E3852ADB0516B438649A1C32E4FA6
8123C9962B5ABDB01A917887D6E787A21C327A11080129A021D28B4E801DF
0ECFB5BD3413BA5E470AF42E802E649DAB0C889077C6D516A775711EBBC0
02B77181B2F12546A68CE33720678D4680A89D60AFF21FB6EE2DA1622D3EE
CD91652A40E49C8E17D3390BDC8EA5FBA7C4E1E6A58F0793BD7C060D5751
7AEA8A0FEBD854058BFCE46ABC1FDBA12558A33A70540D07133DAD24CAA
A8A5E827510150AF8981622D515EA659AEC7945993817185596FBCB2ED9C2
5C4F60FD55C5BF82A70CE2A3872B93068E081C277E93EB3EC63E264E50923
C71281AC80C6471F4089CBA4E453C076D669690770925DE9BB255802653F26
C169DA3ACD32D55638F5FC4A3872B93068E081C277E93EB3EC63E264E5092
r3C71281AC80C6471F4089CBA4E453C076D669690770925DE9BB255802653F
26C169DA3ACD32D55638F5FC4

Development

3.3.1 Desktop GUI using Tkinter

Tkinter is Python's most common and the industry standard GUI (Graphical User Interface) package. It is built upon as a thin layer over the object-oriented layer on top of Tcl/Tk.

Tkinter is not the only package present for GUI development in the Python ecosystem. There are other packages for GUI development involved in the Python ecosystem, such as the **Qt-Framework** or the **Kiwi** Framework. Python- Tkinter is however the most commonly used one.

This report and this project was written for **Python 3.10** and **Tkinter 8.6** running in the **X-11** or **Xorg** Windowing system under GNU/Linux.

Tkinter helps users to build a cross-platform application and is easy to get started with, thus, we used it to build the GUI of our local desktop application. The below figure shows the GUI developed with the help of Python Tkinter for our project

`

Figure 6 : GUI of the Project using Tkinter

3.4.2 Web Application developed using Flask

As was already mentioned, the Python micro-framework Flask was created for web development and is based on Werkzeug and Jinja 2. The “micro” in its definition means that Flask follows the principle of keeping things simple and easy to use instead of filling it up with bloatware. Flask does not make the core choices of the technologies you use, such as to what database to use, or make you bound into using only specific technologies.

The decisions that it does make initially, for example which templating engine to use, can be easily customized or changed by the developer. Flask

follows the ideology of being everything as well as nothing to a developer's point of view at the same time. In the default installation, there are no abstraction layers of a database provided in the Flask micro-framework, form validation, form handling for the frontend and backend or any other abstraction layers that ease the web development.

Rather, Flask micro-framework has support for extensions or other libraries / packages to add these functionalities or features to the application as if they were baked right into the Flask framework from the get go and not added later.

A number of extensions provide features such as integration of desired database inside the framework, form handling and validation, handling of files in both server and client ends, support for various authentication / registration technologies, and much more. The Flask framework may be marked as “micro” but it is nonetheless very much production ready on its own as well without relying on the other extensions or other packages.

Flask may be a micro-framework, but it still does provide some defaults and basic configuration files and defaults to make it easy for the user to get started with the development of the actual application quickly. These defaults can be changed at the will of the developer, but it is mostly not required to. One of the convention of the framework is that the templates and the static files are stored inside the respective sub folders or subdirectories with the name template and static respectively.

This is a good practice as it uses the already constructed conventions followed by the large mass of users already using the framework and also helps anyone new to the project understand the code and the project structure quickly. This convention does not need to be followed per se, but is a good advice and often not changed. The project's web application was utilised for safe cloud storage. The following are the key steps in the web application:

1. The user's initial step was to sign up for an account on our site. The user would receive an automatically created private key upon registration, which they would use to conduct transactions. The user's private key is not kept in the database for security reasons.

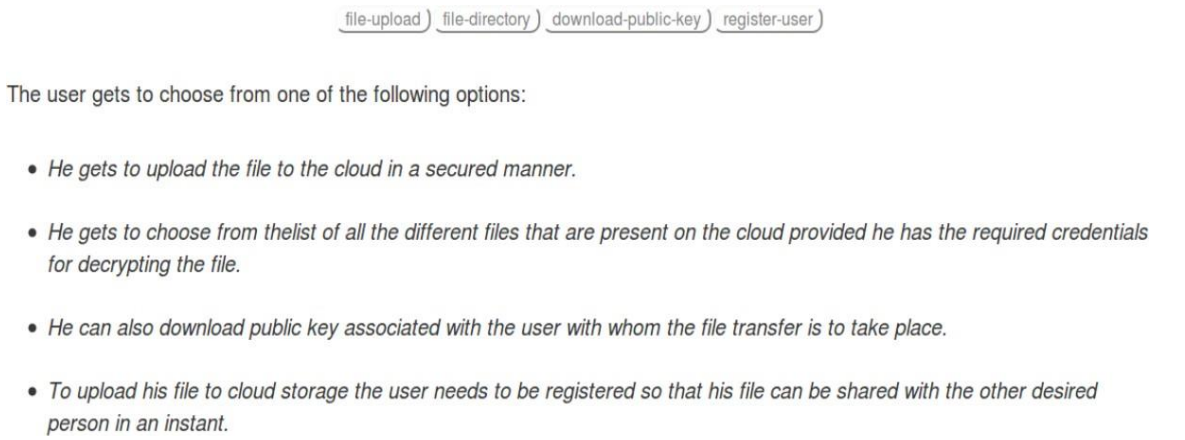


Figure 7 : Logging into the website

2. When the user chooses the upload file option, they are directed to a different page where they can upload the encrypted file.

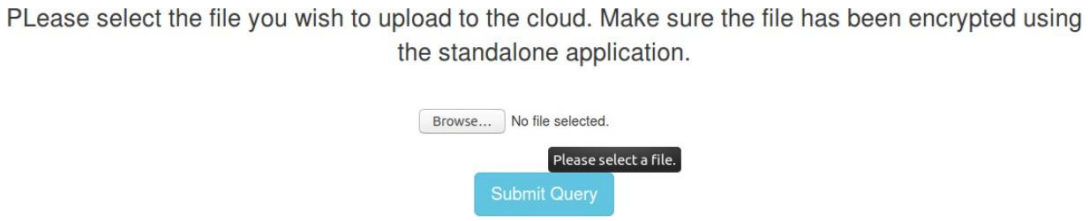


Figure 8 : Selection and uploading of the file

3. The user is directed to the above page, which lists all the various files stored in the cloud, by clicking the file directory option on the page.

Figure 9 : List of uploaded files

4. The user can download the public keys of the registered users and use them to authenticate themselves when someone tries to open a file they have uploaded by choosing options like download-public key.

5. The user is directed to a page where he or she registers as a user by clicking the "Register User" tab.

3.3.2 Hosting on Amazon Web Services - Elastic Compute 2

The steps to host the web application on Amazon Web Services - EC2 module are as follows:

1. To choose a machine to deploy, first sign in to your AWS account and head over to the EC2 console.

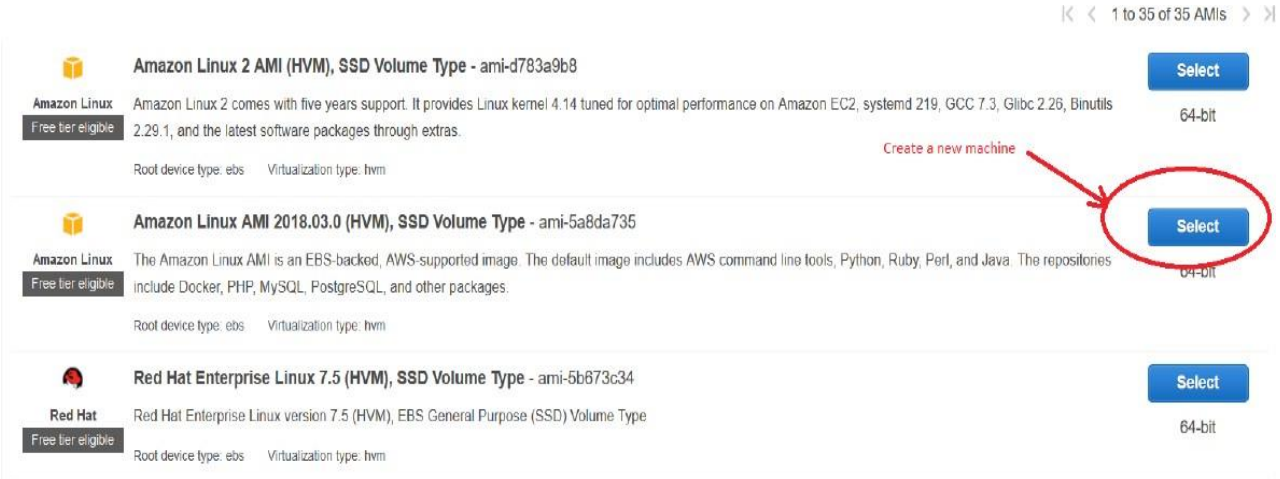


Figure 10 : Selection of an AWS EC2 instance

2. The second step is to configure the machine ports.

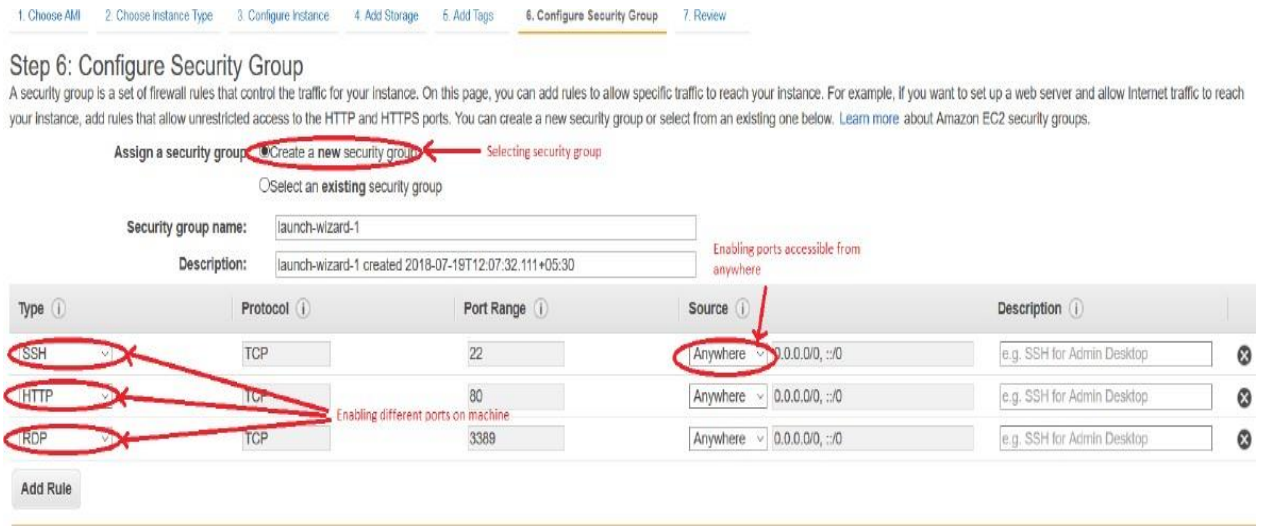


Figure 11 : Configuration of IP address

3. The public IP address of the machine should be copied once it is operational.

Figure 12 : Details of EC2 Instance created

Figure 13 : Coping of the public IP address

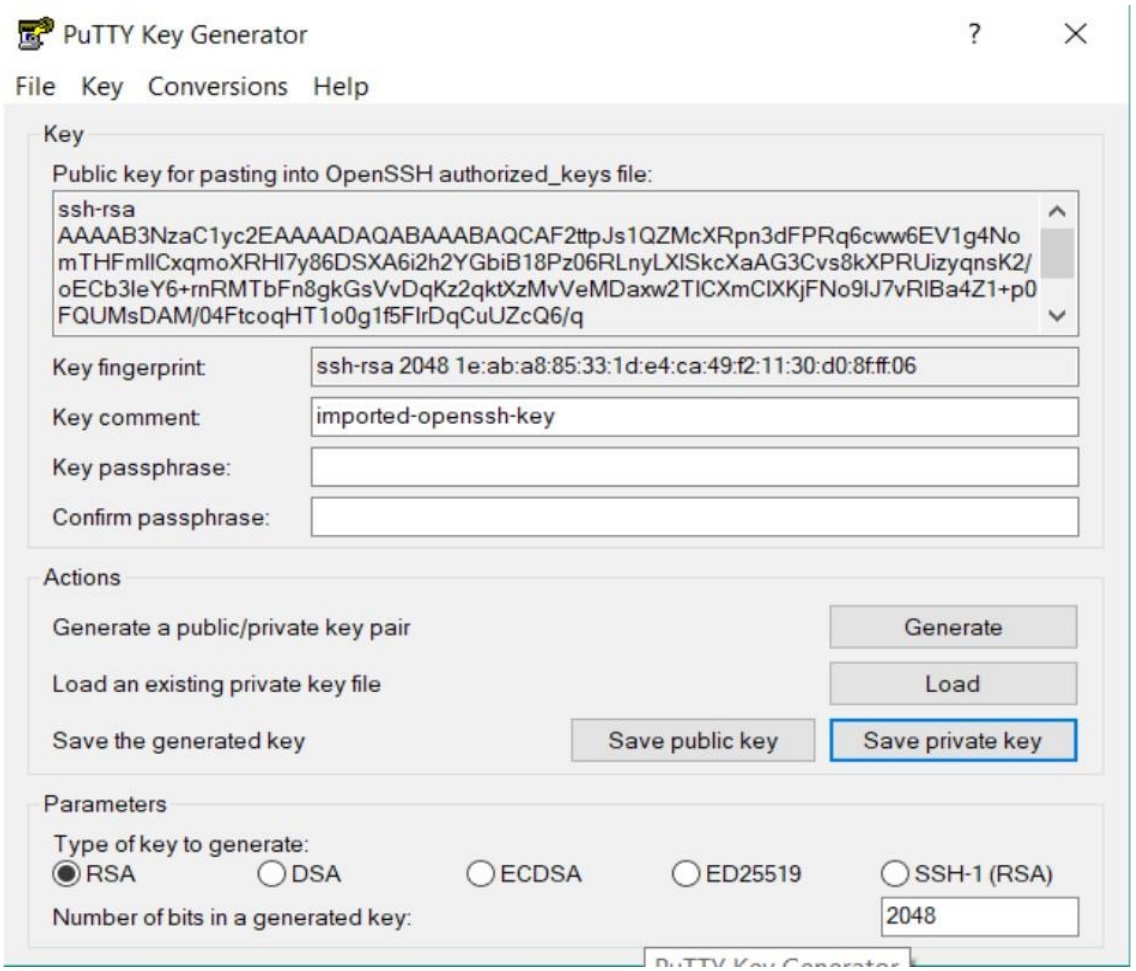


Figure 14 : Creation of private key from puTTY generator

4. Open the Putty Key Generator now, and create a private key for connection using the downloaded file. A private key file should be saved.

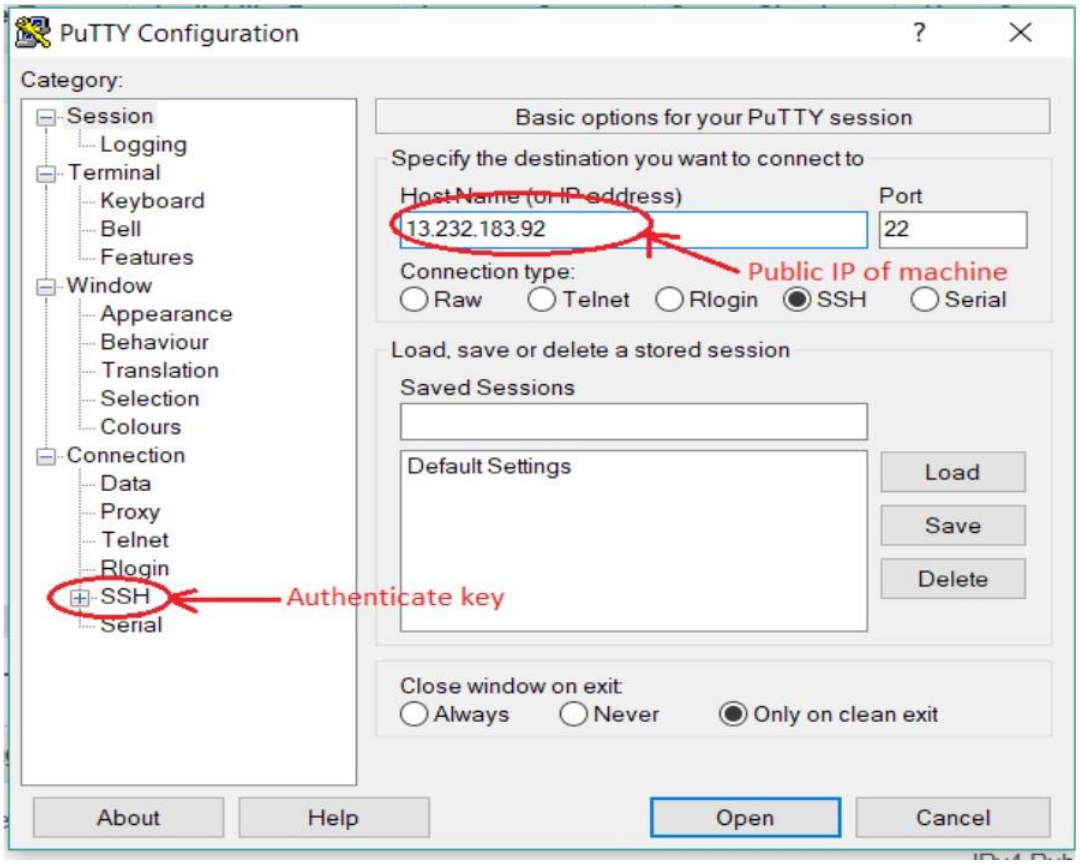


Figure 15 : Configuration of Putty

5. The public IP address should be copied and pasted into the Putty Configuration programme. Next, choose SSH, provide the location of the private key file, and then click Open. This will launch the remote computer's terminal. Open the terminal and enter your credentials.

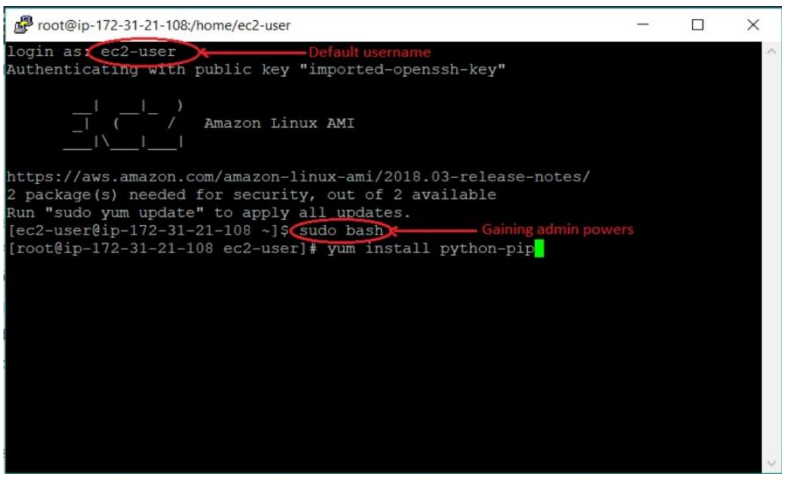


Figure 16 : Logging into the remote EC2 instance

6. After logging in, install any necessary dependencies, such as pip and flask, and then host the application on the distant machine. Use the machine's public IP address to access the application from anywhere once it has started running.

CHAPTER 4

PERFORMANCE ANALYSIS

Time Taken:

The time taken for encryption as well as decryption of a given plain text is calculated by using system clock time: The system clock is recorded twice i.e. before and after the execution of the encryption module and their difference yields the time taken for encryption. The same procedure is followed to calculate decryption time, just that decryption module is invoked instead.

Throughput:

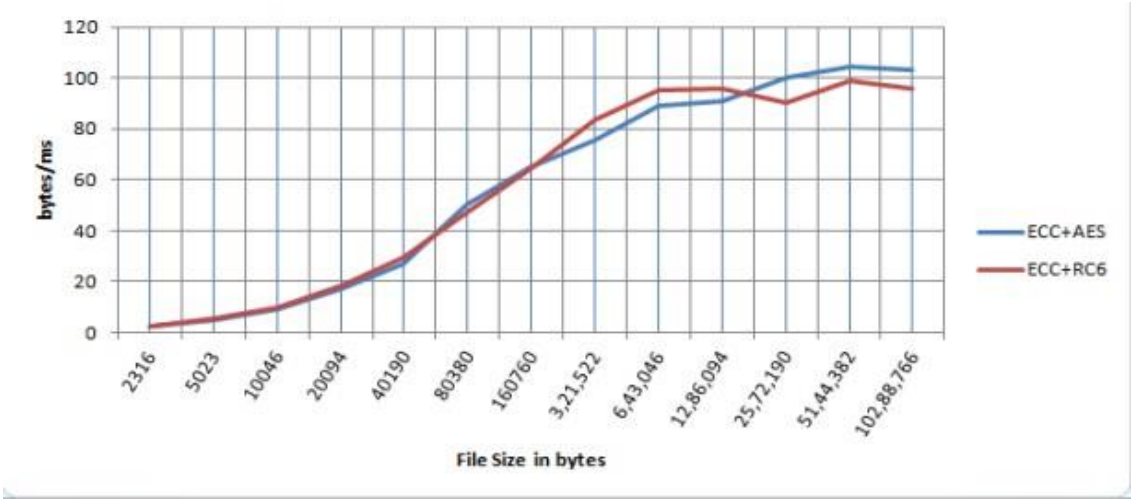
The throughput of a computer in technology refers to the quantity of work it can complete in a specific length of time. One of the most important metrics for evaluating an algorithm's performance is throughput. In case of AES, throughput depends on size of block as well as time taken for encryption/decryption given by:

$$T = \text{block size} / t$$

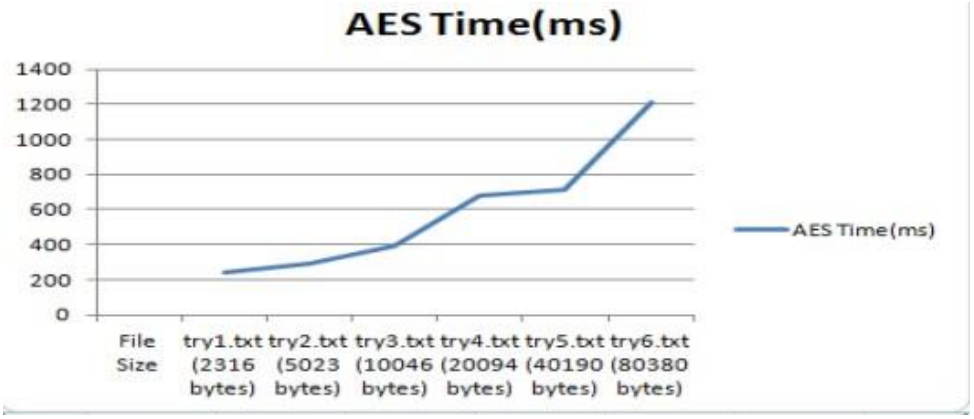
Where, T - Throughput, t - Time taken to encrypt/decrypt

FILE SIZE(bytes)	ECC+AES (bytes/ms)	ECC+RC6 (bytes/ms)
2316	2.351269	2.511931
5023	4.867248	5.270724
10046	9.058611	9.743938
20094	16.91414	18.38426
40190	26.88294	29.57322
80380	50.4266	47.19906
160760	65.11138	64.61415
3,21,522	75.3685	83.64256
6,43,046	89.2252	95.02675
12,86,094	90.83226	95.89844
25,72,190	99.9724	90.0217
51,44,382	104.2639	98.62506
102,88,766	103.0175	95.67741

Table 2 : Testing throughput of multilevel encryption algorithms



Graph 1 : Testing throughput of multilevel encryption algorithms



Graph 2 : Aes runtime with variable file size

Increasing the Block Size:

Using an affine transform after first finding the corresponding reciprocal or just inverting that byte in GF (28). - ShiftRows Transformation - In this manipulation and transformation, no changes or manipulations are made to the bytes in the state's initial row. mat. For the encryption of a sizable amount of data, symmetric cryptography—which employs the same key for both encryption and decryption—is preferable.

As the new symmetric encryption algorithm where the same key is used to encrypt and decrypt the data, the AES 27 algorithm, which has been defined by the National Institute of

Standards and Technology (NIST) of the United States of America, has been widely adopted and accepted to replace with the more effective DES algorithm. A symmetric block cypher, the AES algorithm operates on data blocks of 128 bits utilising cypher keys of 128, 192, or 256 bits. The state, a 4 4 array of bytes that makes up each data block, is where the fundamental operations of the AES algorithm are carried out and operated.

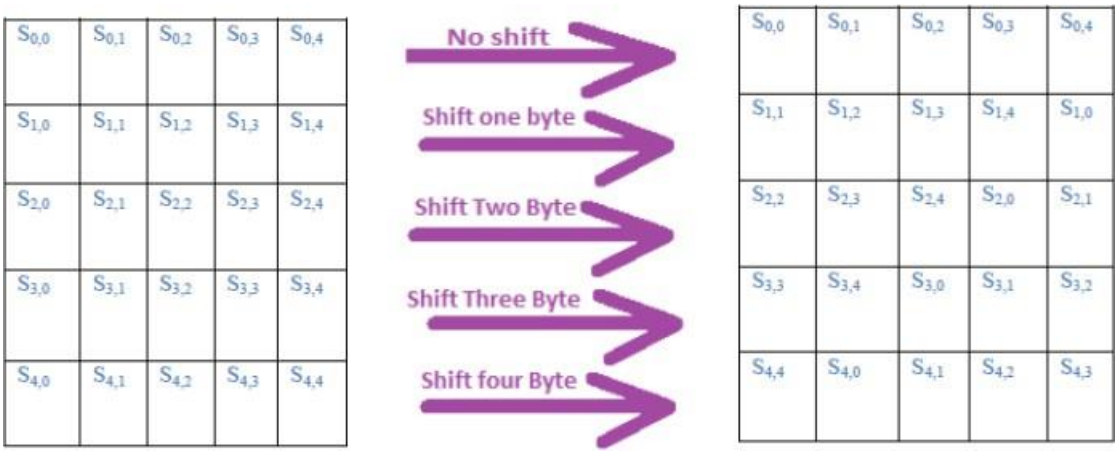


Figure 17 : Shift row transformation

200-bit input is copied to the State array of the 5*5 matrix at the beginning of the encryption process. The data bytes are inserted first into the column and then into the rows. After the first round of necessary addition, there are 10 rounds of encryption. With the exception of a minor variation in the tenth round, the first nine rounds are identical. SubBytes, ShiftRows, MixColumns, and AddRoundKey are the four transformations that make up each of the first nine rounds. nevertheless, in the last round.

SubBytes Transformation - Every byte in the state matrix will be changed in this transformation, and each new byte will match the S-box specifications. The S-box is created by first computing the corresponding reciprocal or simply inverting that byte in GF (28), followed by the application of the affine transform. ChangeRows Transformation The initial row of bytes in the State won't be changed or altered during this operation. Cyclically, the second, third, fourth, and fifth rows move one byte, two bytes, three bytes, and four bytes to the left, respectively.

S.NO.	FACTOR	DES	AES	RSA
1	Developed	1977	2000	1978
2	Key Length Value	56 bit	128, 192 and 256 bits	>1024 bits
3	Type of Algorithm	Symmetric	Symmetric	Asymmetric
4	Encryption Ratio	Low	High	High
5	Security Attacks	Inadequate	Highly Secured	Timing attack
6	Simulation Speed	Fast	Fast	Fast
7	Scalability	Scalable algorithm	No scalability occurs	No scalability occurs
8	Power Consumption	Low	Low	High
9	Hardware and Software Implementations	Better in hardware than in software	Faster and efficient	Not very efficient

Table 3 : Comparison between various encryption algorithm

CHAPTER 5

CONCLUSION

Safe cloud file storage is a current issue that the proposed solution seeks to address and overcome. This strategy is a simple implementation of the recommended technique that may be altered and modified to meet specific needs. It advises adding a second layer of protection to cloud-stored information by utilizing encryption or transforming the data into an unreadable format that can only be converted back to a readable format by using the same key.

FUTURE SCOPE

The future prospect of this project is that the implementation is limited for text files to only using symmetric key encryption, which can be converted to more secure asymmetric key encryption that uses the public and private key concept for enhanced security at the cost of reduced performance. We can also extend the support of the files format to different multimedia also.

REFERENCES

1. Diffie, W.; Hellman, M. (1976). "New directions in cryptography" (PDF). IEEE Transactions on Information Theory. 22 (6): 644–654. doi:10.1109/TIT.1976.1055638. Archived (PDF) from the original on 2014-11-29.

2. A review of data security challenges and their solutions in cloud computing . Available at: https://mecs-press.org/ijieeb/ijieeb-v13-n3/IJIEEB-V13-N3-4.pdf

3. W. Stalling, "William Stallings - Cryptography and Network Security 5th edition.pdf,"Docs,https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxha2hsYWdoZWF8Z3g6MTRmYTdkZDQ4Y2Q2MmFhMQ(accessed Jan. 1, 1970).

4. Chang, Yan-Cheng, et Michael Mitzenmacher. «Privacy Preserving Keyword Searches on Remote Encrypted Data.» Applied Cryptography and Network Security. Springer, 2005. 442-455.

5. Diffie–Hellman key exchange. (2022, November 11). In Wikipedia. https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange

6. "Cloud computing security - Wikipedia,"En, https://en.wikipedia.org/wiki/Cloud_computing_security#Security_issues_associated_with_the_cloud(accessed Jan. 1, 1970).

7. "Quickstart — Flask Documentation (2.2.x)," Flask, https://flask.palletsprojects.com/en/2.2.x/quickstart/#a-minimal-application(accessed Jan. 1, 1970).

8. AL.Jeeva, Dr.V.Palanisamy, K.Kanagaram “Comparative Analysis Of Performance Efficiency And Security Measures Of Some Encryption Algorithms” International Journal Of Engineering Research And Applications (IJERA), Vol. 2, Issue 3, pp.3033-3037, May-Jun 2012.

9. Shakeeba S. Khan, R.R. Tuteja, “Security in Cloud Computing Using Cryptographic Algorithms” International Journal of Innovative Research in Computer and Communication Engineering, Volume 3, Issue 1, January 2015, pp. 148-154.

10. Randeep Kaur ,Supriya Kinger, “Analysis of Security Algorithms in Cloud Computing” International Journal of Application or Innovation in Engineering & Management (IJAIEM), Volume 3, Issue 3, March 2014, pp. 171-176.

11. Mrs. Mamatha, Mr. Pradeep Kanchan, “Use of Digital Signature with Diffie Hellman Key Exchange and Hybrid Cryptographic algorithm to Enhance Data Security in Cloud Computing”, International Journal of Scientific and Research Publications, Volume 5, Issue 6, June 2015, pp 1-4.

12. B. Nithya, P.Sripriya, “A Review of Cryptographic Algorithms in Network Security” International Journal of Engineering and Technology (IJET), Volume 8, Issue 1, Febauray-March 2016, pp. 324-331

mudit

ORIGINALITY REPORT

3%	%	3%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	M. Arunkumar, K. Ashok Kumar. "Malicious attack detection approach in cloud computing using machine learning techniques", Soft Computing, 2022 Publication	1%
2	Mohammad Zakir Hossain Sarker, Md. Shafiul Parvez. "A Cost Effective Symmetric Key Cryptographic Algorithm for Small Amount of Data", 2005 Pakistan Section Multitopic Conference, 2005 Publication	<1%
3	Todorov, . "User Identification and Authentication Concepts", Mechanics of User Identification and Authentication Fundamentals of Identity Management, 2007. Publication	<1%
4	S K Geetha, R Naveenkumaran, Kaushik Selvaraju, C Kishore, A Nagha Rathish. "Blockchain based Mechanism for Cloud Security", 2023 International Conference on	<1%