**CLOUD BASED SECURE-TEXT TRANSFER**

A PROJECT REPORT

Submitted by -

Allada Girish - 20BCS6146

Kolichalam Vinay - 20BCS5780

Varini Malhothra - 20BCS4235

Damini - 19BCS4020

in partial fulfilment for the award of the degree of

# BACHELOR OF ENGINEERING

IN

CLOUD COMPUTING





# 

# BONAFIDE CERTIFICATE

Certified that this project report “Cloud Based Secure-Text Transfer” is the Bonafede work of “Allada Girish, Kolichalam Vinay, Varini Malhotra, Damini”, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science Engineering, who carried out the project under our supervision.

SIGNATURE SIGNATURE

Dr. Aman Kaushik Dhawan Singh

HEAD OF THE DEPARTMENT SUPERVISOR

AIT CSE

**ACKNOWLEDGEMENT**

We would like to express our deep gratitude to our project guide Dhawan Singh, Assistant Professor, Department of AIT CSE, Chandigarh University, for his guidance with unsurpassed knowledge and immense encouragement. We are grateful to Dr.Aman Kaushik, Head of the Department AIT CSE, for providing us with the required facilities for the completion of the project work.

We are very much thankful to the Management, for their encouragement and cooperation to carry out this work. We express our thanks to all the teaching faculty of Department of AIT CSE, whose suggestions during reviews helped us in accomplishment of our project. We would like to thank all nonteaching staff of the Department of AIT CSE, Chandigarh University for providing great assistance in accomplishment of our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. At last but not the least, we thank everyone for supporting us directly or

indirectly in completing this project successfully.

**ABSTRACT**

A remarkable time in the development of computer science has been the advent of cloud computing. It offered the ability to tackle a wide range of issues that were previously thought to be beyond the scope of machine computation. It relieved the strain on those accountable for producing better machinery to keep up with the problems' growing complexity that the machines are meant to address. Cloud computing gave rise to a platform for improved usage of the resource that is distributed worldwide. It is filled with because it is a young field engineers and scientists are devotedly tackling a wide range of issues remove. Security is one of the primary disadvantages of the cloud. Thus, this project puts forth a

Diffie Hellman and encryption are used in this safe cloud file storing method.

**Keywords**

Encryption, Decryption, private key, public key, ciphertext

**TABLE OF CONTENTS**

**TITLE** **PAGE NO**

**ABSTRACT** ……………………………………………………………………………….i

**LIST OF FIGURES** ……………………………………………………………………..ii

**CHAPTER 1 INTRODUCTION**

1.1 Cloud Security…………………………………………………………………..

1.1.1 What is cloud computing? …………………………………………….

1.1.2 Types of Cloud……………………………………………………………….

1.1.3 ………………………………………

1.1.4 ……………………………………………………………

1.2 …………………………………………………………………

1.3 ………………………………………………

1.3.1 ………………………………………………………

1.3.2 …………………………………………………….

1.4 …………………………………………………….

1.5 ……………………………………………………………

**CHAPTER 2 LITERATURE SURVEY**

2.1 …………………………………………………………………

2.2 ………………………………………………………………..

2.2.1 ………………………………………………………………….

2.2.2 ………………………………………………….

2.3 ………………………………………………………………………………

2.4 ……………………………………………………

2.4.1 …………………………………

2.5 ………………………

2.6 …………………………………………………

2.6.1 ……………………………………………………………………………………….

2.6.2 ……………………………..

2.7 ………………………………………………………

2.8 …………………………………………………….

2.9 ………………….

2.10 …………………………………………………

2.11 ………………………………….

**CHAPTER 3 METHODOLOGY**

3.1 ………………………………………..

3.2 ……………………………………………………………………..

3.3 ………………………………………

3.4 ……………………………………………………………….

**CHAPTER 4 EXPERIMENTAL RESULTS**

4.1 System Configuration ……………………………………………………….

4.1.1 Software Configuration …………………………………………………

4.1.2 Hardware Configuration ………………………………………………

4.2 …………………………………………………………………….

4.3 ………………………………………………….

**CHAPTER 5 CONCLUSION AND FUTURE WORKS**

5.1 Conclusion ………………………………………………………………………..

5.2 Future works …………………………………………………………………….

**REFERENCES** ……………………………………………………………………………

**APPENDIX** ……………………………………………………………………………….

**LIST OF FIGURES**

**Figure No Description Page No**

1. **INTRODUCTION**

One crucial area of computer science was cloud computing. It offered the tools necessary to resolve complicated issues that were thought to be beyond the capabilities of a machine. The exceptional platform that cloud computing offered allowed for the best possible use of resources that were dispersed throughout the globe. It lessened the demand on computer makers to create systems with more computing power. The security standard of cloud computing was one of its primary disadvantages. This project suggests utilising Diffie Hellman to store files in the cloud in a secure manner. The necessary file can be decrypted by the user thanks to the key exchange algorithm. Security is the primary issue with cloud computing. It is extremely risky to save sensitive and personal data since it could be taken advantage of or misused by someone or some group with bad intentions. The magnitude of the problem has deterred governments and several organisations from extending their online operations. When it comes to cloud scenarios, the traditional approach to file security is superfluous. To ensure that using cloud services is safe and secure, a great deal of research and development is needed.

* 1. **CLOUD SECURITY**

One of the primary issues in the field of cloud computing is cloud security. If private and sensitive data is stored on a third-party storage device, there is a significant chance that it will be stolen or misused by someone with bad intentions. The danger is so great that it has discouraged governments and numerous other large entities from transferring their activities to an online database. The conventional approaches to information and file security are unnecessary under cloud-related conditions. In this subject, a great deal of research and study is being done to make cloud more dependable and safe. In this enormous body of information, some of the two particularly noteworthy techniques are Diffie Hellman Key Exchange and AES encryption.

Even today's most powerful computers might take millions of years to decipher the encryption and read the file using the latter method due to its extreme power. Using Diffie Hellman for user authentication and any conventional encryption algorithm to encrypt the file is how our approach suggests doing it. The files can be safely preserved in a public domain in this manner without running the risk of being accessed by unauthorised parties.

* + 1. **What is cloud computing?**

The term "cloud computing" refers to Internet-based computing where a shared pool of resources is accessible over a wide range of network connections and can be provided or released with the least amount of service provider and administrative work. A more recent development in the distributed computing paradigm is cloud computing. However, the concept is not new; it has been around for a while. According to L. Kleinrock's prediction in 1969 [16], computer networks are still in their infancy. But as they mature and become more advanced, "computer utilities" will likely proliferate and, like current telephone and electric utilities, they will likely serve individual homes and workplaces all around the nation. The paradigm of utility-based computing that exists now was truly indicated by his vision. When grid computing—which enables users to access computer power on demand—was initially introduced in the mid-1990s, it was one of the most significant steps towards this world. One could consider the emergence of cloud computing as a progression of grid computing.

In late 2006, Eric Schmidt, the CEO of Google, may have created the term "cloud computing," since he brought it to prominence [6]. Thus, although the origins of cloud computing can be traced back to some ancient concepts, it is a relatively recent phenomenon with fresh business, technological, and social viewpoints. The cloud is organically built upon an existing grid-based architecture from an architectural perspective, utilising the grid's services while including new technologies such as virtualization and business models.

Briefly described, a cloud is simply a collection of commodity computers networked together in the same or different geographic locations, working together with the aid of virtualization to service a number of clients with varying needs and workloads on an as-needed basis. Cloud users can pay for cloud services using a pay-per-use basis, just like they would for utilities like phone, energy, and water. XaaS, or "X as a Service," is the term commonly used to characterise these utility services. X can be infrastructure, software, or platform, among other things. Cloud users develop their programmes online and then distribute them to their end customers by using these services offered by cloud providers. The installation and upkeep of necessary hardware and software is therefore not a concern for cloud users. Additionally, because they are required to pay for the services they utilise, they are able to afford them. As a result, by employing cloud services rather than building their own IT infrastructure, cloud consumers can save money and effort in the IT industry.

Big, dispersed data centres are basically what supply cloud services. The cloud is constructed on top of the grid services that these data centres are frequently arranged as. Virtual representations of the actual computers in the data centres are made available to cloud users. Since virtualization essentially creates an abstraction over the real system, it is one of the fundamental ideas of cloud computing. A lot of cloud applications are becoming more and more well-known every day due to its utility model, scalability, availability, and dependability. These apps made distributed computing simple because the cloud provider took care of the important parts.

* + 1. **Types of Cloud**

Cloud computing is the practise of accessing and storing data and programmes on distant servers hosted over the internet as opposed to local servers or the computer's hard drive. Cloud computing, sometimes called Internet-based computing, is a technique in which the user receives a resource as a service via the Internet. Files, photos, documents, and other storable documents can all be considered types of data that are stored.

The following are a few tasks that cloud computing can accomplish:

Data storage, backup, and recovery

on-demand software delivery

creation of fresh services and apps

streaming audio and video

The way the Cloud Deployment Model is deployed changes based on the amount of data to be stored and who has access to the infrastructure. It operates similarly to a virtual computing environment. The cloud deployment model determines the particular kind of cloud environment depending on the nature and purpose of the cloud, as well as ownership, scale, and access. A cloud deployment model specifies who owns and where the servers are located that you use. It outlines the appearance of your cloud architecture, what may be changed, and whether services will be provided to you or if you will need to build everything from scratch. Cloud deployment types also establish the relationships between your users and the infrastructure.

* Public Cloud
* Private Cloud
* Community Cloud
* Multi Cloud

**Public Cloud**

Access to systems and services is possible for everyone thanks to the public cloud. Since the public cloud is accessible to all users, it might be less secure. A public cloud is one where big industrial groups or the general public can access cloud infrastructure services via the internet. In this cloud model, the company providing the services owns the infrastructure, not the customer. It is a kind of cloud hosting that makes systems and services easily accessible to users and consumers. This type of cloud computing is a great illustration of cloud hosting, where service providers offer their services to different kinds of clients. Storage backup and retrieval services are provided under this arrangement either as a free service, as part of a subscription, or on a per-user basis. Take Google App Engine, for instance.

**Public cloud model benefits:**

Low investment for businesses that need quick access to resources, this pay-per-use service is great because there isn't a large upfront cost. There are no setup fees because cloud service providers fully fund the entire infrastructure, therefore no hardware needs to be installed. There is no need for infrastructure management. No infrastructure management is required while using the public cloud. Not requiring maintenance Service providers, not consumers, perform maintenance tasks.

Flexibility in Scaling: On-demand resources are available to meet your business's needs.

**Private Cloud**

The public cloud deployment model and the private cloud deployment methodology are diametrically opposed to each other. It's a solitary user (client) environment. Sharing your hardware with other people is not necessary. The way you manage all of the hardware makes a difference between private and public clouds. It is also referred to as the "internal cloud" and describes the capacity to access services and systems inside a specific organisation or boundary. The cloud platform is deployed within a highly secure cloud environment, safeguarded by robust firewalls and managed by the IT department of an organisation. More control over cloud resources is available with the private cloud.

**Private Cloud Model Benefits**

Improved Management The property belongs to you alone. You acquire total control over IT operations, user behaviour, policies, and service integration. It works well for keeping company data private, to which employees with permission can only access. It is possible to provide better security and access by dividing up resources within the same infrastructure. The goal of this strategy is to support older systems that can't connect to the public cloud.

Customization: With a private cloud, an organisation can customise its solution to match its unique requirements, in contrast to a public cloud deployment.

**Hybrid Cloud**

Hybrid cloud computing combines the finest features of public and private cloud computing by linking the two worlds with a layer of proprietary software. By using a hybrid solution, you may benefit from the lower costs associated with the public cloud while hosting the app in a secure location. Depending on their needs, organisations might use a combination of two or more cloud deployment methods to move data and applications between multiple clouds.

**Hybrid Cloud Model Benefits**

Companies that possess greater flexibility are better equipped to provide customised solutions that cater to their specific requirements Since public clouds are scalable, you will only have to pay for additional capacity if you need it.

Security: Since data is appropriately partitioned, there is a significantly lower likelihood of data theft by malicious actors.

**Community Cloud**

It enables many organisations to have access to systems and services. In order to meet the unique requirements of a community, industry, or company, the services of several clouds are integrated to build this distributed system. Organisations that share concerns or tasks may choose to share the community's infrastructure. In most cases, it is run by a third party or a coalition of one or more local organisations.

**Community Cloud Model Benefits:**

The fact that numerous communities or organisations share the cloud makes it affordable. Higher security is offered by community clouds. You can collaborate with many organisations to share infrastructure, resources, etc.

Information sharing and cooperation: It is appropriate for information sharing and cooperation.

**Multi Cloud**

As the name suggests, using several cloud providers simultaneously is what we mean by this paradigm. The hybrid cloud deployment strategy, which blends public and private cloud resources, is comparable to this. Multiple public clouds are used in multi-cloud instead of combining private and public clouds. Even while public cloud companies offer a plethora of methods to enhance the dependability of their services, errors can happen. Two different clouds experiencing an occurrence at the same time is extremely uncommon. Thus, multi-cloud implementation further enhances your services' high availability.

**Multi Cloud Model Benefits**

By selecting various cloud providers, you can combine the finest elements of their offerings to meet the needs of your organisation, workloads, and applications. You can select cloud regions and zones that are close to your clients in order to reduce latency and enhance user experience.

High service availability: It is quite uncommon for two different clouds to experience an issue simultaneously. Thus, the multi-cloud implementation raises your services' high availability.

* + 1. **Cloud Computing Services**

Using a network of distant servers hosted on the Internet for data processing, storing, and management in place of a local server or a personal computer is known as cloud computing. Cloud providers are businesses that supply these kinds of cloud computing services; their fees are usually depending on use. The core components of cloud computing are clusters and grids.

Five major categories comprise the majority of cloud computing services:

* Software as a service (SaaS)
* Platform as a service (PaaS)
* Infrastructure as a service (IaaS)
* Anything/Everything as a service (XaaS)
* Function as a Service (FaaS)

Because they are stacked on top of one another, these are frequently referred to as the cloud computing stack. Achieving your goals is made easier when you understand what they are and how they differ from one another. These abstraction layers may also be thought of as a layered architecture in which SaaS can offer infrastructure and services from a lower layer can be combined to form services from a higher layer.

**Software as a service (SaaS)**

Service and application delivery over the Internet is known as Software-as-a-Service (SaaS). We relieve ourselves of the complexity of software and hardware maintenance by accessing content over the Internet rather than installing and maintaining it locally. It saves money on both hardware and software maintenance by doing away with the requirement for us to install and run apps on our personal computers or in data centres. Pay-as-you-go cloud service providers offer a whole software solution that falls under the category of SaaS. There is no need to download or install most SaaS software; they may be used immediately from a web browser. There are other names for SaaS apps, such as hosted, on-demand, or web-based applications.

A number of businesses, including Cloud9 Analytics, Salesforce.com, Cloud Switch, Microsoft Office 365, Big Commerce, Eloqua, dropBox, and Cloud Tran, offer software as a service.

**Platform as a service (PaaS)**

Platform as a Service (PaaS) is a subset of cloud computing that gives programmers a framework and platform to create online apps and services. Users only need to use their web browser to access PaaS services, which are hosted in the cloud. The hardware and software are hosted on the infrastructure of the PaaS provider. Thus, PaaS relieves users of the burden of installing hardware and software internally in order to create or execute new applications. As a result, the application is developed and deployed without regard to the hardware. Although the deployed apps and potentially the configuration settings for the application-hosting environment are under the consumer's control, the network, servers, operating systems, and storage systems that make up the cloud infrastructure are not managed or under their control. Platform as a service is offered by a number of businesses, including IBM Smart Cloud, Salesforce, Windows Azure, Google App Engine, Elastic Beanstalk on Amazon Web Services, and cloud Bees.

**Infrastructure as a service (IaaS)**

A service concept called infrastructure as a service (IaaS) provides computer infrastructure to support different functions through outsourcing. IaaS, or infrastructure as a service, refers to the provision of networking hardware, devices, databases, and web servers as an outsourced service to businesses. Hardware as a Service is another name for it (HaaS). Customers of IaaS make payments based on the number of users, usually in units of hour, week, or month. Customers may also be charged by certain providers according to the quantity of virtual machine space they utilise. All that is required to construct such applications and services, as well as to deploy development tools, databases, and other resources, are the underlying operating systems, security, networking, and servers that it offers. A number of businesses, including Amazon Web Services, Bluestack, IBM, Openstack, Rackspace, and VMware, offer infrastructure as a service.

**Anything/Everything as a service (XaaS)**

Another name for it is "everything as a service." These days, the majority of cloud service providers offer a single solution that combines all of the aforementioned features with a few extras.

**Function as a service (FaaS)**

One kind of cloud computing service is FaaS. For the purpose of developing, computing, running, and deploying code or the full programme as functions, it offers its users or customers a platform. Without having to worry about the upkeep of the supporting infrastructure, the user can fully develop and modify the code at any time. The developed code can be run in reaction to the particular occurrence. It is identical to PaaS as well. Event-driven execution is what FaaS is all about. The serverless container is where it is implemented. Now that the programme is finished being developed, the user will initiate the event to run the code. At this point, the triggered event responds and instructs the servers to start running it. All that is there on the servers are Linux servers or any other servers that are fully handled by the supplier. The serverless architecture comes from the fact that the customer has no knowledge of servers and hence does not need to maintain any servers. Though there is still some difference in terms of cost and scalability, both PaaS and FaaS offer the same functionality. With FaaS, scaling up and down automatically based on demand is possible. Although customers must select the scaling parameter based on demand, PaaS nevertheless offers scalability. Users merely pay for the number of execution times that occur when using FaaS. Whether they use more or less, users of PaaS are still required to pay the whole amount based on the pay-as-you-go pricing.

* 1. **DATA ENCRYPTION**

In the field of cryptography, encryption refers to the act of transforming plain text or information into cypher text, or a text that the intended recipient alone can decipher. The term "clipher" refers to the encryption algorithm that is employed. In addition to safeguarding communication networks, it assists in preventing unwanted access to sensitive data such as emails and customer information.

Data encryption is a technique used to protect secret data by converting it into cliphertext that can only be unlocked with a special decryption key generated either before or at the time of encryption. When data is encrypted, it is changed into a code that is only accessible to those with a password or secret key, which is also referred to as a decryption key. Data that has been encrypted is known as ciphertext, whereas data that has not been encrypted is known as plaintext. In today's business environment, encryption is one of the most popular and effective data protection solutions. One cannot stress the significance of encryption as a technique for preserving data integrity.

One cannot in any way emphasise the importance of encryption. It is still feasible for your data to be hacked even when it is kept in a conventional architecture. Data encryption will make your information much more safe even if there is always a danger that it will be compromised. For a moment, imagine it this way. Data encryption before transmission ensures data security even in case it is stored in a secure system. The level of security offered by sanctioned systems is lower.

What do you think the real-world outcome would be of this? Think about a scenario where a worker who utilises company data at work has access to private information. Without any encryption, the user can store the data on a portable disc and take it with them wherever they go. The user can still copy the data if the encryptions are set up beforehand, but it will become unreadable if they try to see it elsewhere. These are the advantages of data encryption that show how valuable it really is.

Digital signatures are employed in conjunction with encryption to verify the validity and integrity of data. Encryption is needed for copy protection as well as digital rights management. Data erasure is possible with encryption. However, even though data recovery programmes can occasionally restore erased data, if you encrypt the data beforehand and discard the key, the ciphertext rather than the original data is the only thing that can be recovered. When moving data over a network, data migration is utilised to make sure that the data cannot be viewed by anybody else on the network. You should encrypt anything you keep in the cloud since VPNs (Virtual Private Networks) require encryption. This can encrypt voice calls in addition to the complete hard disc.

* + 1. **Symmetric Encryption**

The possibility of unwanted access to any type of data is constant in today's digital age. Payment card details and other personally identifiable information (PII) of consumers and clients are most at danger when it comes to financial and payment system data. Protection of personally identifiable information and reduction of risks are two things that firms that handle payments on a daily basis must do with encryption. In cryptography algorithms, there are a few different approaches. Certain algorithms utilise a distinct key for both encryption and decryption procedures. In these kinds of actions, the system or the individual possessing the key has full authentication to decode the message for reading, hence the unique key needs to be protected. The term "symmetric encryption" refers to this method in the context of network encryption. When electronic data is encrypted using symmetric encryption, a single secret key is utilised for both encryption and decryption. To use the key in the decryption process, the communication organisations using symmetric encryption must exchange it. This encryption technique is not the same as asymmetric encryption, which encrypts and decrypts data using a pair of keys—one public and one private.

Data is "scrambled" using symmetric encryption methods to prevent anyone without the secret key to decrypt it from understanding it. The algorithm reverses its activity such that the message is returned to its original readable form once the intended receiver who has the key receives it. A secure random number generator (RNG) can generate a random string of letters or numbers that is used as the secret key, or the sender and recipient can use a predefined password or code. The symmetric keys for banking-grade encryption have to be generated with a RNG approved by industry standards like FIPS 140-2.

* + 1. **Asymmetric Encryption**

A key is used for data encryption and a different key is used for data decryption in some cryptography techniques. Such a public message will therefore be impossible to decipher or read for everyone who obtains it. Most protocols for internet security use this kind of cryptography, which is referred to as "public-key" encryption. This sort of encryption is known as "asymmetric encryption”. In order to encrypt and decrypt a message and safeguard it from unwanted access or use, asymmetric cryptography, commonly referred to as public-key cryptography, employs a pair of linked keys: one public key and one private key. An individual can encrypt a communication using a public key, which is a cryptographic key that only the intended receiver can decrypt using their private key. One's initiator is the only person who has access to a private key, also referred to as a secret key. Sending an encrypted message is as simple as taking the public key of the target receiver and encrypting it with it before sending it out using a public directory.

The communication can then be decrypted by the intended recipient using their associated private key. When a communication is encrypted using the sender's private key, only the sender's public key can be used to decrypt it, proving the sender's identity. Users do not need to physically lock and unlock the message because these encryption and decryption processes take place automatically. Asymmetric cryptography is the foundation of several protocols, such as the secure sockets layer (SSL) and transport layer security (TLS) protocols that enable HTTPS. Software applications that need to verify a digital signature or create a secure connection over an unsecured network, such internet browsers, also use the encryption process.

* 1. **MOTIVATION FOR THE WORK**

The way we live, work, learn, and communicate has all changed dramatically with the advent of the digital age. The digital world has given people all across the world new opportunities, from remote job and e-learning to online cooperation. But this accessibility and ease also mean that safe file sharing is required. Threats to steal private information are growing in frequency as more of it is posted online. When sensitive data is transferred over the internet, secure file sharing helps to protect its integrity, confidentiality, and privacy. We will discuss the value of safe file sharing in the digital era and the reasons it is so important to safeguard private data from online attacks in this post. It is simple to declare that sharing files in a safe manner is advantageous, but how significant is it really? In the digital age, let's examine the main justifications for using secure file sharing.

### **Protecting Confidential Information**

In the current digital age, it is imperative for both individuals and businesses to safeguard sensitive data from cyberattacks, including financial information, legal papers, personal information, and intellectual property. Cybercriminals are constantly searching for ways to circumvent security measures in digital networks and obtain unauthorised access to private information. Identity theft, extortion, and financial damage may ensue from a cybercriminal successfully intercepting a file transfer and obtaining customer information. With access control, authentication, and encryption to guarantee that only authorised users may access the data, secure file sharing provides a means to safeguard private information. Data that has been encrypted is jumbled into unintelligible code that can only be unlocked by authorised users who possess the encryption key. Authentication verifies that users are who they say they are, while access control restricts file access to only those with permission.

### **Ensuring Privacy and Compliance**

In order to protect privacy and adhere to data protection laws, secure file sharing is crucial. Strict rules govern the handling of sensitive information in a number of areas, including finance and healthcare. Legal repercussions and heavy fines may follow noncompliance with these regulations. For instance, the HIPAA law, which addresses the secure storage of patient data, mandates that practises in the healthcare sector adhere to it. There are severe HIPAA infractions that carry penalties of up to $1,500,000 USD in addition to up to five years in jail. Affected individuals may suffer serious repercussions if their identity is stolen as a result of exchanging private information, such as financial or medical records, without the proper privacy safeguards in place.

In order to prevent unwanted access to sensitive data, secure file-sharing systems can assist in ensuring compliance with data protection laws including GDPR, HIPAA, and PIPEDA. A lot of file-sharing programmes, like TitanFile, abide by these rules by default, therefore any data transmitted through TitanFile is compliant. You don't need to be concerned about if you're overlooking any crucial customer data protection procedures.

* 1. **PROBLEM STATEMENT**

One of the key issues in the world of cloud computing is cloud security. Inorder to conceal text data and send it safely to the recipient, no similar security method was used. Hackers could easily defeat the security of text communication by using a variety of combinations. A text was not transferred securely via a comparable channel. The risk of data theft and misuse by anyone with malice inclination is quite high when storing personal and sensitive information on a third-party storage media. Because of the size of the threat, many large organisations, including governments, have decided against moving their operations to a cloud platform. When using the cloud, it is unnecessary to use the conventional techniques of information and file security.

The goal of this project is to provide a reliable method for sending text-based data over the cloud between users. The requirement for assuring the confidentiality, integrity, and validity of transmitted data is growing along with the need for distant collaboration and communication.

**2. LITERATURE SURVEY**

**2.1 CRYPTOGRAPHY AND NETWORK SECURITY**

The study and application of secure communication methods in the presence of third parties, also referred to as adversaries, is called cryptography. In order to adhere to the several facets of information security, it deals with creating and evaluating mechanisms that stop malevolent third parties from obtaining information shared between two companies. When two parties share data or a message that cannot be viewed by an enemy, this is referred to as "secure communication." An adversary in the field of cryptography is a malevolent entity that seeks to obtain valuable information or data, therefore weakening the fundamentals of information security. The four main tenets of contemporary cryptography are data integrity, data confidentiality, authentication, and non-repudiation. The process of encrypting data and communications so that only the intended recipient can decipher and process it is known as cryptography. Consequently, information access that is not authorised is prevented. "Hiding" is the meaning of the prefix "crypt," and "writing" is the meaning of the suffix "graphy." The methods used in cryptography to safeguard data are derived from mathematical ideas and a collection of computations based on rules, or algorithms, which transform messages into forms that are difficult to decipher. These algorithms are used for the creation of cryptographic keys, digital signatures, data privacy protection, online browsing, and the protection of private transactions like debit and credit card purchases.

**Applications Of Cryptography**

**Computer passwords**

Cryptography plays a major role in computer security, especially in the creation and upkeep of passwords. A user's password is hashed and compared to the previously saved hash when they log in. Before being saved, passwords are encrypted and hashed. This method encrypts the passwords so that they cannot be read by a hacker, even if they manage to get access to the password database.

**Digital Currencies**

Cryptography is another tool used by digital currencies like Bitcoin to secure transactions and stop fraud. Transactions are secured by complex algorithms and cryptographic keys, which makes it almost impossible to tamper with or counterfeit the transactions.

**Secure web browsing**

Cryptography provides online browsing security by protecting users from man-in-the-middle attacks and eavesdropping. In order to create a secure channel for communication, the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols encrypt data exchanged between the web server and the client using public key cryptography.

**Electronic signatures**

Signing papers electronically involves using electronic signatures, which are the digital version of handwritten signatures. Cryptography is used in the creation of digital signatures, and public key cryptography is used in their validation. Legally binding in numerous countries, electronic signatures are becoming more and more common.

**Authentication**

When entering onto a computer, accessing a bank account, or using a secure network, among other scenarios, cryptography is utilised for authentication. Authentication protocols use cryptographic techniques to verify that the user is who they say they are and that they have the authorization to access the resource.

**2.2 CLOUD CRYPTOGRAPHY**

Data handled and stored in cloud computing environments can be secured using a set of methods called cloud cryptography. The process of changing data to appear different until a legitimate user enters in and sees the "plaintext" (i.e., true) version of the data is known as encryption. The process of converting plaintext into ciphertext, or a random string of characters, requires the usage of cryptographic keys. Because it protects your data even after it leaves your company's IT structure, cloud encryption is a smart move. This implies that the data is secure wherever it goes within your cloud computing services. With encryption, your organization's cybersecurity is enhanced since the data itself is protected rather than the locations where it is kept. Through the use of encryption and secure key management systems, it offers data privacy, data integrity, and data secrecy. In cloud cryptography, common techniques include:

* Symmetric encryption: encrypts and decrypts data using the same key.
* Asymmetric encryption: uses two different keys, a public key for encryption and a private key for decryption.
* Hash functions: create a unique digest of a message to ensure its integrity.
* Key management: securely stores and manages encryption keys to ensure the security of encrypted data.

In order to safeguard confidential data and guarantee adherence to laws like GDPR and HIPAA, cryptography must be used in cloud computing environment. Data saved on the cloud is protected by encryption known as cloud cryptography. In order to prevent data breaches, hacks, or malware infections, cloud cryptography employs a number of safeguards that provide an additional layer of safety. It is safe and convenient for customers to utilise shared cloud services because all data hosted by cloud providers are encrypted. Sensitive information is protected using cloud cryptography without affecting information delivery.

**2.2.1 How does cryptography works in cloud?**

Cloud cryptography uses encryption, which is the process of converting text into ciphertext using computers and algorithms. Then, by decoding this ciphertext with a string of bits and applying an encryption key, it may be transformed back into plaintext. There are several methods via which data can be encrypted :

**Pre-encrypted data which is synced with the cloud-**

Software is available to pre-encrypt data before it is sent to the cloud, rendering it unreadable for any attempt at hacking.

**End-to-end encryption-**

Messages are sent and received, and only the sender and the recipient can read them.

**File encryption-**

When data is encrypted, it means that even if someone were to try to intercept the file, they would not be able to access the contents within. This is known as file encryption.

**Full disk encryption-**

Any files saved to an external device will come with an automated encryption patch applied. Computer hard drives can be secured using this key approach.

**2.2.2 Advantages of cloud cryptography**

By encrypting data, cloud cryptography provides cloud services with an equivalent level of protection. Sensitive cloud data can be safeguarded without causing data transfer delays. To maintain a balance between efficiency and security, several organisations specify different cryptographic algorithms for their cloud computing.

• The users' access to the data is private. As a result, there is less hacker cybercrime.

• If an unauthorised person attempts to make changes, the organisation is notified right away. Access is provided to those who possess cryptographic keys.

• Cloud encryption, which is now essential in today's data-driven world, enables organisations to be proactive in their defence against data breaches and cyberattacks.

• Cloud encryption, which has become essential in today's data-driven world, enables organisations to be proactive in their defence against data breaches and cyberattacks. • The encryption keeps data safe while it is transferred from one computer to another.

• One of the safest ways to store and transport data is through encryption, which meets with organization-imposed limits. Recipients of the data can determine whether the data they have received is corrupted, enabling a rapid response and solution to the attack.

**2.3 ENCRYPTION ALGORITHMS**

For the aim of protecting electronic data while it is being transferred over networks, encryption algorithms help with the process of converting plain text into encrypted text and back again. Data that has been coded or encrypted usually prevents hackers and other unauthorised people from accessing it. Certain encryption methods are thought to be faster than others, but as long as those who create them—many of whom have skills in mathematics—remain abreast of technological developments, this kind of encryption ought to endure as cybercriminals become more proficient.

**2.3.1 Block Cliper###**

The Feistel cipher structure incorporates block cyphers. Block ciphers generate ciphertext using a set number of rounds and keys. A block cipher is a sort of encryption method that generates ciphertext by processing fixed-size blocks of data, typically 64 or 128 bits. To guarantee the security and effectiveness of the algorithm, a block cipher's design must adhere to a number of crucial concepts. Among these ideas are:

**Number of Rounds:**

AES has 10 rounds, making it more secure, while DES has 16 rounds, making it more secure. The number of rounds is routinely taken into account while creating design criteria. It simply indicates the number of rounds that are appropriate for an algorithm to make it more sophisticated.

**Design of function F:**

The Round Function is the central component of the Feistel Block cypher construction. The Round function can be used to calculate the complexity of cryptanalysis, meaning that an increase in the round function's complexity will inevitably lead to a rise in complexity overall. The inclusion of the avalanche effect in the round function adds to its complexity, since the avalanche effect would cause a malicious output even if a single bit in plain text were changed.

**Confusion and Diffusion:**

It is important for the cypher to produce confusion and diffusion so that an attacker finds it difficult to figure out how the ciphertext and plaintext relate to each other. Confusion makes it harder to figure out the key since the ciphertext must be a complicated function of the plaintext. Diffusion makes it challenging to decipher the encryption pattern since it implies that a minor change in the plaintext should result in a large change in the ciphertext .

**Key Size:**

A sufficiently big key size is necessary to thwart brute-force attacks. An attacker will find it more difficult to guess the proper key if the key size is higher because there are more alternative keys. For the majority of applications, 128 bits is regarded as the secure key size.

**Key Schedule:**

The key size should be large enough to prevent brute-force attacks. A larger key size means that there are more possible keys, making it harder for an attacker to guess the correct one. A key size of 128 bits is considered to be secure for most applications.

**Block Size:**

Block sizes ought to be sufficiently large to thwart attacks that take advantage of statistical patterns found in plaintext. For most purposes, a block size of 128 bits is thought to be secure.

**Non-linearity:**

For the purpose of creating confusion, the cipher's S-box should not be linear. Attacks that take use of the linear characteristics of the cypher can target a linear S-box.

**Avalanche Effect:**

A slight change in the plaintext or key should result in a considerable change in the ciphertext if the cipher exhibits the avalanche effect. As a result, you can be sure that whatever modification you make to the input will completely alter the output.

**Security Analysis:**

It is important to examine the cipher's resilience to multiple types of attacks, including linear and differential cryptanalysis and brute-force attacks. It is important to assess the cipher's ability to withstand implementation attacks such side-channel attacks.

**2.3.2 DES**

A key-length block cypher that has been helpful in improving data security is the Data Encryption Standard (DES). Due to its vulnerability to extremely potent assaults, the data encryption standard (DES) has been seen to be slightly less popular. Being a block cypher, DES encrypts data in blocks of 64 bits each. This implies that 64 bits of plain text are fed into DES, which generates 64 bits of ciphertext. For encryption and decryption, the same algorithm and key are utilised, with a few slight variations. 56 bits comprise of the key. Early in the 1970s, an IBM team developed the symmetric-key block cypher known as the DES (Data Encryption Standard), which was later accepted by the National Institute of Standards and Technology (NIST). Using 48-bit keys, the algorithm translates the plain text into ciphertext in 64-bit blocks. Because it is a symmetric-key technique, the data is encrypted and decrypted using the same key. Different keys would be needed for encryption and decryption if the algorithm were asymmetrical.

**Initial Permutation (IP)**

The text in simple form is broken up into smaller 64-bit segments. The IP is carried out prior to the initial round. This stage explains how the transposition process is put into practise. The 58th bit, for instance, replaces the first bit, the 50th bit replaces the second bit, and so on. The resulting 64-bit text is divided into Left Plain Text (LPT) and Right Plain Text (RPT), which are equal parts of 32 bits each.

**Step 1: Key Transformation**

Since all the bits in every eighth place of a 64-bit key are eliminated, a 56-bit key—which is what the DES process uses—is already known to us. We generate a 48-bit key in this stage. Each half of the 56-bit key is divided equally, and the bits are shifted in a circular manner to the left based on the number of rounds.The result is a rearranging of all the pieces in the key. The shifting procedure results in the elimination of some bits, as we can see, giving rise to a 48-bit key. The term compression permutation refers to this procedure.

**Step 2: Expansion Permutation**

This RPT, which is formed during the IP step, is 32 bits in size. From 32 bits to 48 bits, it is enlarged in this stage. In order to create 48-bit data, the 32-bit RPT is divided into 8 chunks of 4 bits each, with an additional 2 bits added to each piece. The bits are then permuted inside each chunk. The 48-bit key that was acquired in step 1 and the 48-bit extended RPT are put together using an XOR function.

The algorithm process breaks down into the following steps:

• An initial permutation (IP) function receives the 64-bit plain text block at the start of the procedure.

• The plain text is thereafter subjected to the initial permutation (IP).

• The two halves of the permuted block, known as Left Plain Text (LPT) and Right Plain Text (RPT), are then created by the initial permutation (IP).

• There are sixteen encryption rounds for every LPT and RPT.

• Lastly, the LPT and RPT are reunited, and the newly combined block is subjected to a Final Permutation (FP).

The intended 64-bit ciphertext is produced as a result of this procedure.

**2.3.3 AES**

The AES Encryption algorithm, sometimes referred to as the Rijndael algorithm, uses a 128-bit block/chunk size symmetric block cypher. It uses keys with 128, 192, and 256 bits to convert each of these distinct blocks. The ciphertext is created by joining these blocks together once they have been encrypted. It is predicated on a substitution-permutation network, or SP network for short. It consists of a sequence of interconnected operations, such as bit shuffle operations (permutations) and substitutions, which are the replacement of inputs with particular outputs. For the computers of that period, the Data Encryption Standard algorithm—also referred to as the DES algorithm—made sense when it was developed and standardised. As can be seen in the graphic below, breaking into the DES algorithm got easier and faster every year, based on modern computational standards. With longer key sizes and more difficult-to-crack cyphers, a more resilient algorithm was urgently required. In order to address this issue, the triple DES was developed, but because to its very sluggish speed, it was never adopted widely. In order to address this issue, the Advanced Encryption Standard was created.

**2.4 CLASSICAL ENCRYPTION TECHNIQUES**

**2.4.1 Symmetric Clipher Model**

The oldest and most fundamental type of encryption is called symmetric encryption. It encrypts and decrypts data using the same key in both processes. As a result, another name for it is Single-Key Encryption. There are five main components of a symmetric cypher model:

1. Plain Text (x): This is the first information or message that the sender wants to convey to the recipient. It is a component of the encryption algorithm's inputs.
2. Secret Key (k): This is a value, string, or text file that the encryption and decryption algorithms utilise, respectively, to encode and decode plain text to cypher text and vice versa. The encryption algorithm has no bearing on it. It sets the rules for all plain text conversions. The secret key is necessary for all changes and substitutions that are made.
3. The encryption algorithm (E) generates cypher text as an output after receiving the secret key and the plain text as inputs. It suggests utilising a secret key to do various operations on the plain text, including substitutions and transformations.
4. Cipher Text (y): This is the formatted version of the unintelligible plain text (x), which is used to encrypt data as it travels. It is entirely reliant on the encryption algorithm's secret key. A different cipher text is generated for every distinct secret key.
5. Decryption Algorithm (D): This algorithm reverses the encryption process at the recipient's end. In addition, it decrypts the cypher text that was sent by the sender using the secret key as input. The output is plain text.

Requirements for Encryption:

To conduct encryption, just two conditions must be satisfied. As they are

1. Encryption Algorithm: Even if an attacker manages to obtain one or more cypher texts, they should be unable to decipher the secret key due to the strength of the encryption algorithm that generates the cypher texts.
2. Robust and secure method for exchanging secret keys: The sender and the recipient need to be able to exchange secret keys in a reliable and safe manner. To prevent an adversary from accessing the secret key, it must be impenetrable.

**2.4.2 Substitution Clipher Model**

Encryption is a technique used to hide some data. Ciphertext, which is unintelligible plain text that has been encrypted, is obtained. A key determines which character from the predetermined set of characters in a substitution cypher is used to replace a particular plain text character. If we were to shift by 1, for instance, A would become B, B would become C, and so on.

Remark: Caesar cypher is a special example of substitution cypher in which 3 is assumed to be the key.

Mathematical representation

The encryption can be represented using modular arithmetic by first transforming the letters into numbers, according to the scheme, A = 0, B = 1,…, Z = 25. Encryption of a letter by a shift n can be described mathematically as.

Method:

* List every character that has been created.
* To save the replacement for every character, create a dictionary.
* Depending on whether the text is being decrypted or encrypted, change each character according to the specified rule.
* Print the newly created string.

**2.4.3 Columnar Transposition Clipher Model**

Use the Columnar Transposition Cypher to cypher or decipher the provided text given a plain-text message and a numerical key. Similar to the Rail Fence Cypher, the Columnar Transposition Cypher is a type of transposition cypher. Writing the plaintext out in rows and reading the ciphertext off one column at a time is known as columnar transposition. Writing the plaintext out in rows and reading the ciphertext off in columns is known as columnar transposition. In its most basic version, it is the Route Cypher, in which the path is to read each column sequentially.

**2.5 RC4 ENCRYPTION ALGORITHM**

RC4 is a stream cypher and variable-length key algorithm. One byte (or bigger units at a time) is encrypted by this algorithm. The key-stream, or output of the pseudorandom bit generator, is merged one byte at a time with the plaintext stream cypher via the X-OR operation. The key-stream is an unpredictable stream of eight bits that is produced without knowledge of the input key.

**Features of the RC4 encryption algorithm:**

* RC4 is an example of a symmetric key encryption algorithm, meaning that both encryption and decryption are performed using the same key.
* RC4 is classified as a stream cypher algorithm, meaning that data is encrypted and decrypted one byte at a time. In order to obtain the ciphertext, it creates a key stream of pseudorandom bits that is XORed with the plaintext.
* RC4 is adaptable to diverse security needs because to its ability for varying key sizes, ranging from 40 bits to 2048 bits.
* Quick and effective: The RC4 encryption technique is quick and effective, making it a good fit for low-power gadgets and applications that need to send data at high speeds.
* Widely used: Secure Sockets Layer (SSL), virtual private networks (VPN), wireless networks, and file encryption are just a few of the many applications that have made extensive use of RC4.
* Vulnerabilities: RC4 has a number of flaws, one of which is a bias in the initial few bytes of the keystream that can be used to retrieve the key. It is therefore not advised to utilise RC4 in any new applications.

**2.6 BLOWFISH ALGORITHM**

The 64-bit, symmetric Blowfish block cypher has configurable length. Created in 1993 as a "general-purpose algorithm," Bruce Schneier's goal was to offer a quick, free, and seamless replacement for the outdated Data Encryption Standard (DES) and International Data Encryption Algorithm (IDEA) encryption processes. With no patent and no cost for any purpose, Blowfish outperforms DES and IDEA in terms of speed. Because of its small block size, which is seen as insecure, it was unable to fully replace DES. By using a bigger block size of 128 bits, Twofish, its successor, addressed the security issue. Though many cypher suites and encryption products on the market today use the method, complete Blowfish encryption has never been cracked. Blowfish accepts a variable-length key with a maximum length of 448 bits and a block size of 64 bits. It is composed of sixteen Feistel-like iterations, each of which works on a 64-bit block divided into two 32-bit words. Data is encrypted and decrypted using the same encryption key via Blowfish.

**Data encryption**

A 16-round Feistel network is used to encrypt data; each round consists of a key- and data-dependent substitution as well as a key-dependent permutation. A crucial component of Blowfish's data encryption technology, large, key-dependent S-boxes operate with the replacement method. XORs, a kind of logic gate, and adds on 32-bit words are the foundation of all encryption processes.

**Key expansion and subkeys**

A total of four hundred and sixty-seven bytes of subkey arrays are created throughout the key expansion procedure from maximum size four48-bit keys. Subkeys are essential to the Blowfish algorithm, which makes extensive use of them. It is necessary to precompute these subkeys before encryption or decryption may occur.

Four 32-bit S-boxes with 256 entries each and eight 32-bit subkeys make up the P-array in Blowfish. The following is how the subkeys are computed:

* The fixed string of pi's hexadecimal digits is used to initialise the P-array and S-boxes.
* In order to XOR all of the items in the P-array with the key bits, the first element (P1) in the array is now XORed with the first 32 bits of the key, followed by P2 with the second 32 bits, and so on.
* The technique uses the stages listed above to encrypt all-zero strings.
* The output obtained from step 3 above is substituted for the P1 and P2 arrays.
* Using altered subkeys, Blowfish encrypts this output.
* Step 5's output changes P3 and P4 in the P-array.
* Until all four S-boxes and all P-arrays are changed, this process is repeated.

Blowfish generates all the subkeys and processes by running 521 times in total, which equates to roughly 4 KB of data.

**2.7 SHA1 HASH ALGORITHM**

A 160-bit (20-byte) hash value is produced by the cryptographic process known as SHA-1, or Secure Hash process 1. A message digest is what this hash value is called. A 40-digit hexadecimal number is often the result of this message digest. Developed by the National Security Agency of the United States, it is a Federal Information Processing Standard. Since 2005, SHA-1 has been regarded as untrustworthy. SHA-1 SSL certificates were no longer supported by major tech companies' browsers, including Microsoft, Google, Apple, and Mozilla, by 2017. Java uses the MessageDigest Class, which is part of the java.security package, to compute cryptographic hashing values. To get the hash value of a text, the MessageDigest Class offers the following cryptographic hash algorithm as follows:

MD2

MD5

SHA-1

SHA-224

SHA-256

SHA-384

SHA-512

The static function getInstance() is where these algorithms are initialised. Following algorithm selection, a byte array containing the results is returned once the message digest value has been computed. The resulting byte array is converted into its signum form using the BigInteger class. To obtain the anticipated MessageDigest, this representation is subsequently transformed into a hexadecimal format.

**2.8 EXISTING SYSTEM**

Cryptographic techniques are often classified into several groups according to the quantity of keys required for encryption and decryption. Among these techniques are public-key cryptography and hashing, which encrypt data permanently and separately using different keys for the two operations, respectively. All of the encryption and decryption processes in secret key cryptography use the same key. Alternatively, hackers might just use alternative combinations to circumvent the security of the SMS message. This was not the secure method of sending a text message.

**2.9 LIMITATIONS OF EXISTING SYSTEM**

* Brute-force attacks can be used against public key encryption.
* When a user misplaces their private key, this technique also fails, making public key encryption the most susceptible.
* Additionally vulnerable to a man-in-the-middle attack is public key encryption. By interfering with the public key communication, a third party can alter the public keys through this attack.
* Any subordinate certificate becomes completely insecure in the event that the user private key utilised to create certificates at a higher level in the PKI (Public Key Infrastructure) server hierarchy is compromised or unintentionally revealed. This is known as a "man-in-the-middle attack." This is another way that public key encryption is vulnerable.

**2.10 PROPOSED SYSTEM**

With this method, users can send text within a text file as a secure message. To send the file securely, users upload it, and to lock the text, they enter a password or key. Even if hackers manage to get access to the system, they will be unable to decrypt the content because of this key. The recipient will need the key in order to decipher the encrypted text.

A secret key is generated for enhanced security using the Diffie Hellman Key Exchange Algorithm. After that, the sender sends the recipient the text file along with the decryption key. The recipient opens the file, inputs the password or key to access the content, and then presses the decode key to access the sender's secret text. By using this method, you can confirm that there won't be any outside intervention from hackers or crackers when sending your secret message. If the sender sends this text file in plain sight, without disclosing its contents to any third parties, it will reach its intended destination. The system keeps all pertinent data in an online database.

**3. METHODOLOGY**

**3.1 DIFFIE-HELLMAN KEY EXCHANGE ALGORITHM**

The Diffie–Hellman key exchange (DH) protocol, named for Whitfield Diffie and Martin Hellman, is a secure means of exchanging cryptographic keys over a public channel. One of the first real-world applications of public key exchange in the realm of cryptography is DH. In a public key cryptosystem, the processes of encoding and decoding are controlled by separate keys, E and D, thus it is computationally impossible to calculate D from E (for example, by needing more than 10100 instructions). Therefore, it is possible to reveal the encrypting key E to the public without jeopardising the decrypting key D. The primary philosophy underlying the Diffie-Hellman Key Exchange Protocol was this. Consequently, each network user has the ability to store his encrypting key in a publicly accessible directory. This makes it possible for any user to communicate with any other user on the system by sending a message that is encrypted so that only the intended recipient can decode it. Consequently, a multiple access cypher is a public key cryptosystem. Therefore, any two people, regardless of whether they have ever talked before, can have a private chat. Each uses his own secret deciphering key to decipher communications he gets from the other after sending them to each other encrypted with the recipient's public enciphering key.

By using a Diffie-Hellman key exchange, two parties can create a shared secret that they can use for covert communication when transferring data over a public network. The conceptual diagram above uses colours rather than large numbers to show the main notion of the key exchange. Alice and Bob decide on an arbitrary starting hue that doesn't need to be kept a secret to start the procedure. In this instance, yellow is the colour. They each choose a secret colour, which they keep to themselves; in this instance, the colours are orange and blue-green. The most important step in the procedure is when Bob and Alice combine their mutually shared colour with their own secret colour to create orange-tan and light-blue mixes, respectively, and then publicly trade the two coloured combinations. Lastly, each of the two combines his or her personal colour with the colour they received from the other. The end product is a final colour mixture that is exactly the same as the partner's final colour mixture (in this case, yellow-brown). It would be computationally challenging for a third party to deduce the secret colours if they were to listen in on the conversation. Actually, it takes a lot of processing power for current supercomputers to do this task in an acceptable length of time when big numbers are used in place of colours.

**3.1.1 Prime**

If two smaller natural numbers cannot be multiplied to generate a larger natural number, the result is a prime number, often known as a prime number. The prime number that is chosen is the only user-defined pre-existing parameter in the Diffie-Hellman algorithm. To fend off the known attacks, the prime number p ought to be sufficiently large. A 232-digit number on the order of 2^768 has been successfully attacked using the most effective method, which is called an NFS (attack on the network file system). A much larger p would seem to be a good idea; preferably at least 1536 bits, but at least 1024 bits would be more reasonable. An additional characteristic of p is that p−1 must to have a big prime factor (q), and the factorization of p−1 ought to be known. We will presumably be safe if we choose a random prime number, p, and a random generator, g, but we won't know for sure. If the order of your random g happens to have some minor components, we might also leak some information about the private exponent.

**3.1.2 Method**

Pay attention to the Diffie-Hellman key exchange protocol's mathematical implementation.

The number p is prime.

Modulo p, g is a primitive root.

1. Bob and Alice decide to utilise a modulus of 23 and a base of 5.
2. Alice receives the created 4. as her private key, which she should keep private.
3. As a result, Alice's public key will be 54%23 = 625%23 = 4.
4. Bob receives his private key, which is generated as 3 and that he should not share with anybody.
5. Consequently, Bob's public key will be created as 53%23 = 125%23 = 10.
6. At this point, Alice generates a secret key using Bob's public key. As an example, (public key of Bob; private key of Alice) mod p = (104) % 23 => 10000 % 23 => 18
7. Conversely, Bob employs a comparable technique to produce a secret key, namely: (public key of AlicePrivate Key of Bob) mod p=> (43) % 23 => 64 % 23 => 18.

As a result, it has been demonstrated mathematically that Alice and Bob can produce the identical key without being aware of each other's private key. This is how the Diffie-Hellman Key Exchange Protocol is being implemented.

**3.2 RSA ALGORITHM**

Leonard Adleman, Adi Shamir, and Ron Rivest created the RSA public-key signature algorithm. The approach, which was originally described in their 1977 paper, uses logarithmic functions to maintain working complexity that can survive brute force while remaining post-deployment quick. The RSA approach is used to verify digital signatures, as seen in the graphic below. In addition to managing digital signature verification, RSA can also encrypt and decrypt generic data to facilitate safe data interchange. The RSA algorithm's full process is depicted in the graphic above. The next part will provide you with more information about it. It reverses the key set usage when general data encryption and decryption are performed using RSA. In contrast to signature verification, it encrypts the data using the recipient's public key and decrypts it using the recipient's private key. Therefore, in this case, there is no need to swap any keys.

When it comes to RSA cryptography, there are two main parts. These are:

* Key Generation: Key generation is the process of creating the keys needed to both encrypt and decrypt the data that has to be shared.
* Function of Encryption and Decryption: The actions required to jumble and recover the data.

Go ahead and see how the entire procedure functions, from generating the key pair to encrypting and decrypting the data, while keeping the above graphic in mind.

Key Generation

Prior to using the functions to produce your ciphertext and plaintext, you must generate your public and private keys. They employ a few parameters and variables, all of which are described below:

* Select p and q, two huge prime numbers.
* Determine z = (p-1) and n = p\*q.Q-1).
* Select a number e such that 1 < e < z.
* Determine that d = e-1mod(p-1)Q-1).
* A private key pair can be bundled as (n,d).
* A public key pair can be packaged as (n,e).

Encryption/Decryption Function

After creating the keys, you send the parameters to the functions, which use the appropriate key to compute your plaintext and ciphertext.

* The ciphertext is equal to me mod n if the plaintext is m.
* The plaintext is equal to cd mod n if the ciphertext is c.

**Advantages Of RSA**

* No Key Sharing: RSA encryption works by utilising the public key of the recipient, therefore exchanging secret keys is not necessary in order to receive communications from other people.
* Proof of Authenticity: Because the key pairs are connected, it is impossible for a recipient to intercept the message because they lack the necessary private key to decrypt the data.
* Faster Encryption: Compared to the DSA method, the encryption procedure is faster.
* No Data Can Be Changed: Since tampering with the data will change how the keys are used, the data will be impenetrable while in transit. Additionally, the information cannot be decrypted with the private key, warning the recipient of manipulation.

**3.3 ARCHITECTURE**

**Modules Description**

* Login: To authenticate a user, a login comprises a set of credentials.
* Authentication: Using a login and password, authentication is the process of confirming a person's or device's identification. Prior to using the cloud, various users register.
* Security Provider: The Security Provider generates a private key and verifies the user's identity before allowing them to upload a file belonging to the owner. We save the encrypted file on the cloud server.
* Availability of Files: The availability of a resource in the cloud storage is verified for any file that is requested by an authorised user.
* Reliability Check: The availability of resources is recorded in a distinct file named Reliability Check.
* Resource allocation: Either resources are assigned by the virtual machine, or they are not.
* Regeneration: In the event that the file is present, the end user can access it with ease; if it is not, it is generated and supplied to the user upon request.

**Algorithm:**

* Create an account.
* Diffie Hellman Key Algorithm is used to generate a public and private key.
* Please select the file that needs to be emailed.
* Upload the file after entering your public and private receiver keys.
* We'll encrypt the file you upload.
* The file will now be decrypted by the recipient using both the sender's public key and private key.

**3.4 PLANNING AND ANALYSIS**

Providing the most secure cloud file storage was the primary goal of this project. Thus, a number of problems needed to be resolved, such as:

* Where is the file that requires encryption?
* How does the user need to be verified?
* What will the AES encryption key be?
* What is the connection between this key and the Diffie Hellman Key exchange protocol?

Our first plan was to encrypt the text online, but the man in the middle of the attack forced us to change our minds. Additionally, we discovered that there is an attack known as NFS. NFS in a cyberattack when a computed key of order 2768 was possible. This amounted to about 232 digits. As a result, we concluded that the method requires a bigger prime number. Consequently, we employed a prime number with six hundred digits. We arrived at this plan of action after examining each of these questions. We used an application to encrypt the file on the owner's PC in order to give them more control. The file was created using the Diffie Hellman algorithm, and it could only be decrypted by users who possess the final identical key from this procedure. The basis for the key for AES encryption was the final Diffie Hellman key, which was the same for both intended participants. The various project components are covered in further detail in the subsections that follow.

**EXPERIMENTAL RESULTS**

**4.1 SYSTEM CONFIGURATION**

**4.1.1 Software Configuration**

* Python Tkinter
* AWS account
* Linux virtual machine
* Visual Studio Code
* Knowledge on RSA

**4.1.2 Hardware Configuration**

* Laptop or PC
* Internet connection

**4.2 PYTHON**

For general-purpose programming, Python is an interpreted high-level language. Python was designed with an emphasis on code readability, particularly through the use of substantial whitespaces. It was created by Guido van Rossum and initially released in 1991. It offers structures that make programming understandable at both small and big sizes. In Python comes with an autonomous memory management system and a dynamic type system. It boasts an extensive and well-rounded standard library and supports a variety of programming paradigms, including as imperative, procedural, and object-oriented. We were able to select Python as the language to employ for both the stand-alone and web-based application development because to its extensive and well-documented standard libraries. The libraries and frameworks that were heavily used to create the intended framework are briefly described below.

**4.2.1 Python-tkinter**

The de-facto standard GUI (Graphical User Interface) package for Python is called Tkinter. It is a thin layer on top of Tcl/Tk that is object-oriented. There are other Python GUI programming toolkits besides Tkinter. But it is the one that is most frequently utilised. The annual choice to maintain Tkinter is referred to by Cameron Laird as "one of the minor traditions of the Python world." Python's Tkinter is a set of GUI (graphical user interface) widgets. This article was produced for Linux's X Window system using Python 2.7 and Tkinter 8.5. We built the GUI for our standalone programme using Tkinter since it is user-friendly and facilitates the creation of cross-platform applications.

**4.2.2 Python-flask**

Flask is a Python microframework licenced under the BSD licence, derived from Werkzeug and Jinja 2. Although it is possible, "micro" does not imply that Flask is less functional or that your entire web application must fit into a single Python file. Flask strives to maintain a basic yet expandable core, as indicated by the "micro" in microframework. Not many decisions, like which database to utilise, will be made by Flask. What templating engine to use is one of the decisions it does make, and it is easily changed. To ensure that Flask is everything you need and nothing more, the rest is up to you. If there are other libraries available that can handle something like form validation or database abstraction, Flask does not by default include it. Rather of requiring it to be built in Flask itself, Flask allows you to add this kind of functionality to your application using extensions. Form validation, upload handling, database integration, many open authentication systems, and more are offered by a multitude of extensions. Flask is suitable for a range of production applications despite its "micro" size. Along with a few logical defaults and several configurable options, Flask also offers some initial conventions. Generally speaking, folders with the names templates and static are where static files and templates are kept within the Python source tree of the programme. Although this can be altered, you usually don't have to, particularly in the beginning.

**4.2.3 Python-crypo**

PyCrypto is another well-maintained Python package. It is a constantly evolving library that offers primitives and recipes for cryptography. It offers multiple encryption techniques and safe hash functions. Hash functions were crucial in the development of the framework. The hash function's primary job was to generate a private key for each new user that was smaller than the prime number that was predetermined. The reason was that the library hashlib was used, and a digest was produced after a variable length was supplied to the function. AES, blowfish, ARC2, and many other symmetric and asymmetric key encryption algorithms were already implemented in this library, which made it even more helpful. The ciphertext that is produced by encryption methods is altered in a manner that is reliant on a key or keys. We used symmetric-key encryption, specifically blowfish combined with AES, to encrypt the text. A two-layer encryption was used by us. This is where custom encryption could be utilised.

**4.3 AMAZON WEB SERVICES**

Amazon Web Services, offers consumers, businesses, and governments on-demand cloud computing platforms via subscription-based payment plans. Through the Internet, customers can access an always available virtual cluster of computers thanks to this technology. Virtual computers from AWS mimic several aspects of a physical computer, such as hardware (processing CPU(s) and GPU(s), RAM, local and SSD storage, networking, and operating system selection in addition to pre-installed applications like web servers, databases, and CRM. AWS customers can connect to their AWS system with a contemporary browser since every AWS system virtualizes its console I/O (keyboard, monitor, and mouse).

**4.3.1 AWS EC2**

Cloud computing services that offer safe, scalable computing power are offered by Amazon Elastic Compute Cloud (Amazon EC2). Developers are supposed to find web-scale cloud computing easier with it.

It's quite easy to obtain and configure capacity with Amazon EC2 because to its straightforward web service interface. It allows you operate in Amazon's tested computing environment and gives you total control over your computer resources. When your computing needs change, you can swiftly scale capacity both up and down using Amazon EC2, as it cuts down on the time it takes to obtain and boot new server instances to minutes. The ability to only pay for the capacity that you really use thanks to Amazon EC2 transforms the economics of computing. With the help of Amazon EC2, developers can create apps that are resistant to failure and keep them separate from typical failure situations. The primary computing solution provided by AWS is its EC2 instances, which provide a wide range of customizable features. In addition, it offers associated services like AWS Lambda, EC2 Container service, Elastic Beanstalk for app deployment, and autoscaling. Prices are generally comparable, particularly after AWS switched from hourly to second-hourly pricing for its EC2 and EBS services in 2017. Our ability to use Amazon for hosting and service deployment on its IaaS platform was made possible by its affordable prices, dependable and secure infrastructure, and extensive online support.

**4.4 STAND-ALONE APPLICATION**

A standalone programme was developed to use a variety of symmetric encryption techniques to both encrypt and decrypt the text. This graphical user interface (GUI) programme encrypts and decrypts text using the sender's and recipient's keys. Later, this text is posted to the directory's website.

**4.5 WEB APPLICATION**

Cloud storage that is secure was made possible by the project's web application. The following are the main steps the web application takes:

* For the user, registering on our platform was the first step. A randomly generated private key was provided to the user upon registration, which they would use to conduct transactions. The user's private key is not kept in the database for reasons of security.
* After choosing to upload a file, the user is directed to an additional page where they can upload an encrypted file.
* When the user selects the file directory option on the website, all of the various files saved on the cloud are displayed on the page above. Choosing various choices such as download-public key allows the user to access a list of registered users' public keys, which they can download and use to authenticate themselves when someone tries to open their submitted content.
* Selecting the "Register User" tab directs the user to a page where they can create an account on the site.

**4.6 HOSTING ON AWS EC2**

* To start an Amazon EC2 instance, fork this repository.
* The Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type can be chosen and created.
* The Configure Security Group menu should be toggled to while the machine is being created.
* Open ports SSH, HTTP, and RDP here. Then, modify the Source setting on the port to Anywhere.
* Putty and Puttygen should be installed.
* Download and save the publicKeyPair.pem file.
* To generate a private key, open Puttygen, choose the publicKeyPair.pem file.
* Put it away now as a private key save.
* To get started, open the AWS machine dashboard, copy the IPv4 Public IP of the new instance, and then open Putty. Open Putty, copy the IP in Host Name (or IP address).
* Toggle SSH>Auth in Putty, and choose the private key you generated.
* Simply click "open," and it's done! It opens the instance terminal.
* Input "login as:" ec2-user.
* Enter following command
* $sudo bash
* $yum install python-pip
* $yum install git
* $pip install flask
* Now, clone the local machine's forked repository.
* Comment on line 169 of src/web-application/app.py and uncomment on line 168.
* Make changes to the github repository.
* Copy the repository to the terminal, then type the following command:
* $python app.py

**4.7 SCREENSHOTS AND RESULTS**

**CONCLUSION AND FUTURE WORKS**

* 1. **CONCLUSION**

It concludes the document by saying that there will be a very subtle transition in the text between two or more authority. It is the recipient's responsibility to decode the text using the key that was sent by the sender. By employing that key, the recipient can find the secret text. Observation with the naked eye cannot be done directly. The text will not be seen unless the sender additionally provides the key, to prevent data theft or hacking without the sender's awareness. The proposed project focuses on finding a secure cloud file storage solution. Although it can be adjusted and customised to suit the needs, this strategy is a basic implementation of the recommended methodology. The goal is to give files saved in the cloud two layers of security by using encryption and Diffie Hellman.

In this research, we will look into the safe use of the RSA (Rivest-Shamir-Adleman) algorithm for text communication over the internet. To safeguard user-to-user and user-to-cloud data sharing, keys will need to be generated and encryption-decryption performed twice. This will ensure that neither the cloud nor an outside attacker can read the messages sent back and forth between the two users. The Diffie Hellman key exchange protocol makes it easier for us to solve the problems with key agreement and key exchange. Keep in mind that the key exchange protocol does not allow for encryption or decryption of any kind.

**5.2 FUTURE WORKS**

**6. REFERENCES**

[1] Jianbing Ni, Kuan Zhang, Yong Yu, Tingting Yang,

Identity-based Provable Data Possession from RSA

Assumption for Secure Cloud Storage, Published in: IEEE

Transactions on Dependable and Secure Computing,2020.

[2] R. Patil Rashmi ,Yatin Gandhi,Vinaya Sarmalkar, Prajakta

Pund,Vinit Khetani, RDPC: Secure Cloud Storage with

Deduplication Technique, Published in: 2020 Fourth

International Conference on I-SMAC (IoT in Social, Mobile,

Analytics and Cloud) (I-SMAC),2020.

[3] Nouman Kabir, Shaharyar Kamal, Secure Mobile Sensor

Data Transfer using Asymmetric Cryptography Algorithms,

Published in: 2020 International Conference on Cyber

Warfare and Security (ICCWS), 2020.

[4] Bindhu Raj L.R.Vandana, Santhosh Kumar B.J.Integrity

based Authentication and Secure Information Transfer over

Cloud for Hospital Management System, Published in: 2020

4th International Conference on Intelligent Computing and

Control Systems (ICICCS), 2020.

[5] Shafiq Riaz, Ali Haider Khan, Muhammad Haroon, Sadia

Latif, Sana Bhatti, Big Data Security and Privacy: Current Challenges and Future Research perspective in Cloud Environment, Published in: 2020 International Conference on Information Management and Technology (ICIMTech), 2020.

[6] 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.

[7] Kuhlman, Dave. "A Python Book: Beginning Python,

Advanced Python, and Python Exercises". Archived from the

original on 23 June 2012.

[8] "About Python". Python Software Foundation. Retrieved

24 April 2012., second section "Fans of Python use the phrase

"batteries included" to describe the standard library, which

covers everything from asynchronous processing to zip files."

**APPENDIX**