

PRIVATE DOCUMENTS VAULT

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

MUKKAPATI ASHOK KUMAR (Reg. No - 39110648)
MUTYALA SRI SATYA NIHANTH (Reg. No – 39110658)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHISALAI,
CHENNAI - 600119

APRIL - 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with —All grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **MUKKAPATI ASHOK KUMAR (39110648)** and **MUTYALA SRI SATYA NIHANTH (39110658)** who carried out the Project Phase-1 entitled “**Private Documents Vault**” under my supervision from January 2023 to April 2023.

Internal Guide

Dr. A. Christy MCA., Ph.D.

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.

Submitted for Viva voce Examination held on _____

Internal Examiner

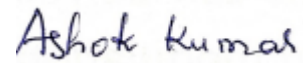
External Examiner

DECLARATION

I, **MUKKAPATI ASHOK KUMAR (Reg. No- 39110648)** and **MUTYALA SRI SATYA NIHANTH (Reg. No- 39110658)** hereby declare that the Project Phase-2 Report entitled **Private Documents Vault** done by me under the guidance of **Dr. A. Christy M.E., Ph.D.**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE:

PLACE: Chennai



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D, Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. A. Christy MCA, Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-1 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

With the evolution of computer systems, the amount of sensitive data to be stored as well as the number of threats on these data grows up, making data confidentiality increasingly important to computer users. Currently, with devices always connected to the Internet, the use of cloud data storage services have become practical and common, allowing quick access to such data wherever the user is. Such practicality brings with it a concern, precisely the confidentiality of the data which is delivered to third parties for storage. Documents may also be leaked by curious administrators. A simple solution is for the user to encrypt all documents before submitting them. This method, however, makes it impossible to efficiently search for documents as they are all encrypted. We propose a private document vault with server side encryption, which also enables customers to easily deploy encryption and other security solutions by offering robust, central management of encryption keys. In addition, this paper designs and implements a prototype system. Through the verification and analysis of its usability and security, it is proved that the solution can meet the data security protection requirements of sensitive E-documents in the open network environment. We use streamlit, a popular open-source framework for creating machine learning and visualization apps in Python. Our preliminary performance evaluation shows that this feature introduces acceptable computation overheads when compared to submitting documents directly to a cloud storage service.

Chapter No	TITLE	Page No.
	ABSTRACT	5
	LIST OF FIGURES	v
	LIST OF ABBREVIATIONS	8
1	INTRODUCTION	9
2	LITERATURE SURVEY 2.1 Inferences from Literature Survey	20
	2.2 Open problems in Existing System	20
3	REQUIREMENTS ANALYSIS	21
	3.1 Feasibility Studies/Risk Analysis of the Project	21
	3.2 Software Requirements Specification Document	22
	3.3 System Use case	22
4	DESCRIPTION OF PROPOSED SYSTEM	23
	4.1 Selected Methodology or process model	23
	4.2 Architecture / Overall Design of Proposed System	33
	4.3 Description of Software for Implementation and Testing plan of the Proposed Model/System	33
	4.4 Project Management Plan	37
	4.5 Financial report on estimated costing	37
	4.6 Transition/ Software to Operations Plan	36
5	IMPLEMENTATION DETAILS	38
	5.1 Development and Deployment Setup	38
	5.2 Algorithms	56
	5.3 Testing	61
6	RESULTS AND DISCUSSION	63
7	CONCLUSION	64
	7.1 Conclusion	64
	7.2 Future work	64
	REFERENCES	65
	APPENDIX	67
	A. SOURCE CODE	67

	B. SCREENSHOTS	77
	C. RESEARCH PAPER	81

LIST OF FIGURES

Figure No	Figure Name	Page No
1	Public key	10
2	System architecture	12
3	Web app	26
4	Creating and switching	27
5	Default encryption	30

LIST OF ABBREVIATIONS

ABBREVIATION

EXPLANATIONS

ML

MACHINE LEARNING

VPC

VIRTUAL PRIVATE CLOUD

RSA

Rivest-Shamir-Adleman

AES

ADVANCED ENCRYPTION STADARD

NLP

NATURAL LANGUAGE PROCESSING

TF

TERN FREQUENCY

IDF

INVERSE DOCUMENT FREQUENCY

CHAPTER 1

INTRODUCTION

Encryption is one of the most basic requirements for ensuring data privacy, especially for end-to-end protection of data transmitted across networks. Plain text is encrypted using an encryption algorithm and an encryption key. Encryption converts the readable text to an unreadable text which is called cyphertext (encrypted data).

Server-side encryption is the encryption of data at its destination by the application or service that receives it. Amazon S3 encrypts your data at the object level as it writes it to disks in its data centers and decrypts it for you when you access it. As long as you authenticate your request and you have access permissions, there is no difference in the way you access encrypted or unencrypted objects. For example, if you share your objects using a resigned URL, that URL works the same way for both encrypted and unencrypted objects. Additionally, when you list objects in your bucket, the list API returns a list of all objects, regardless of whether they are encrypted.

S3 Server-Side Encryption

- Server-side encryption is about data encryption at rest
- Server-side encryption encrypts only the object data.
- Any object metadata is not encrypted.
- S3 handles the encryption (as it writes to disks) and decryption (when objects are accessed) of the data objects

Server-Side Encryption with S3-Managed Keys - SSE-S3

Encryption keys are handled and managed by AWS

Each object is encrypted with a unique data key employing strong multi-factor encryption.

SSE-S3 encrypts the data key with a master key that is regularly rotated.

In Server-side encryption, the data is encrypted after being sent to the S3 bucket and before storing it in the S3 bucket.

Server-side encryption has the following three options:

1. **Use Amazon S3-managed keys (SSE-S3):** - In this, the key material and the key will be provided by AWS itself to encrypt the objects in the S3 bucket.
2. **Use CMK (Customer Master key) in AWS KMS (SSE-KMS):** - In this, key material and the key will be generated in AWS KMS service to encrypt the objects in S3 bucket.
3. **Use a customer provided encryption key (SSE-C):** - In this, the key will be provided by the customer and Amazon S3 manages the encryption and decryption process while uploading/downloading the objects into the S3 bucket.

With SSE-S3, Amazon S3 managed Server-side encryption uses one of the most secure block Ciphers, AES -256 (Advanced Encryption Standard) bit, to encrypt each object with a unique key which means no overlapping keys are used for encrypting the objects. Also, these unique keys are encrypted with a Master key which is rotated regularly to yield additional data security.

With SSE-KMS, Amazon S3 uses the AWS KMS functionality to encrypt the data in the S3 bucket. Combining Amazon S3 with the AWS KMS provides a perfect blend of security and availability.

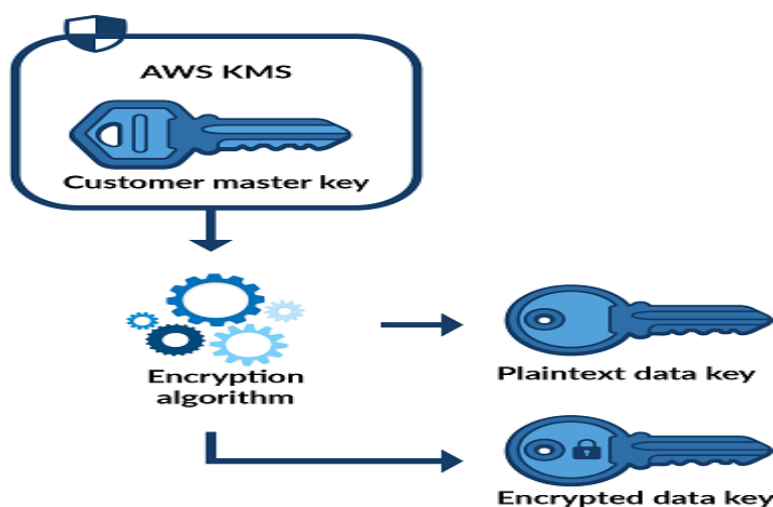


Fig 1

To encrypt the objects, you need a data key. Now to generate a data key you can specify a CMK (Customer Master Key) that you have already created otherwise S3 will request AWS KMS to create a default CMK which can be used to create a data key.

Now CMK using the encryption algorithm (AES-256) creates two keys, one is plaintext data key and the other is encrypted data key.

S3 encrypts the object with plaintext data key and deletes the key from memory. The encrypted object along with the encrypted data key is then stored in S3.

While retrieving the object S3 sends the encrypted data key to KMS. KMS matches the correct CMK then it decrypts the encrypted data key and sends the plaintext data key to S3. S3 then retrieves the object by decrypting the object with this plaintext data key.

Using a virtual private cloud (VPC) is common amongst enterprises looking to run scalable virtual networks in a private and completely customizable environment. This enables organizations to ensure security, isolation, and centralization for their virtual operations. AWS offers a VPC service in the form of Amazon VPC, a natural home for VPC users, given the plethora of fully managed, cost-efficient, and data-optimizing cloud services on the AWS platform. Even within a VPC, organizations must enforce granular permissions on data, to restrict access to sensitive data, meet compliance requirements, and minimize unnecessary data-control risks.

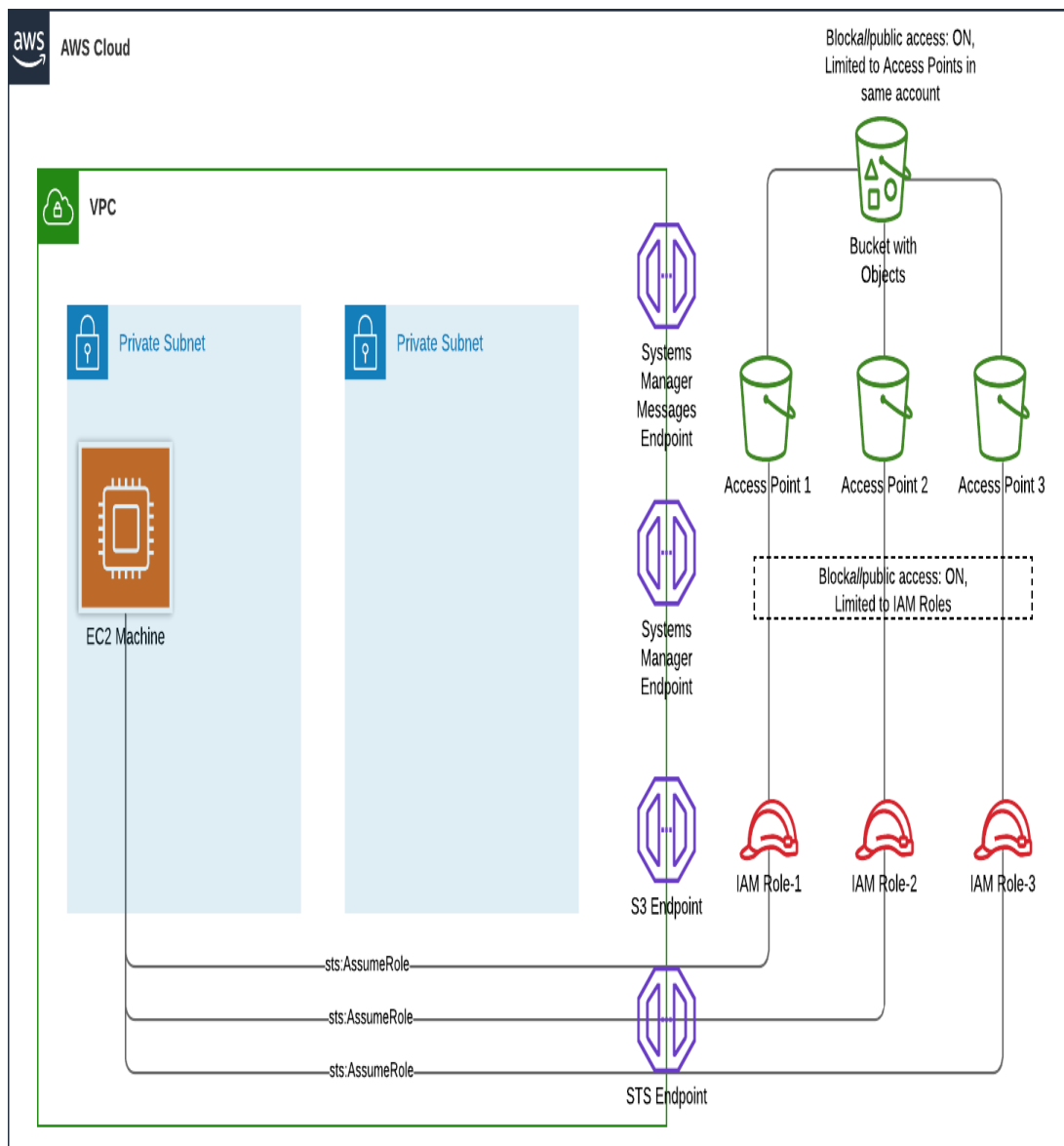


Fig 2

To meet compliance requirements and restrict access to sensitive data, many customers want to restrict data sharing within their Amazon VPC based upon AWS Identity and Access Management policies. Many of these customers must also allow for granular controls for data access within Amazon S3 for users assigned to particular IAM roles. With multiple S3 bucket policies to manage, controlling S3 bucket access on a granular level can become an easy task.

AWS Employees are using Amazon S3 Access Points in combination with VPC endpoint policies to simplify managing access to shared datasets on Amazon S3. In that post, you learned how to ensure that users could only access certain buckets from within a VPC, using a VPC endpoint policy. You also learned how to use an S3 Access Point to enforce your data management and access rules without having to constantly edit your VPC endpoint policy upon new bucket creation.

While you use Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3), each object is encrypted with a unique key. As an additional safeguard, it encrypts the key itself with a root key that it regularly rotates. Amazon S3 server-side encryption uses one of the strongest block ciphers available, 256-bit Advanced Encryption Standard (AES-256) GCM, to encrypt your data. For objects encrypted prior to AES-GCM, AES-CBC is still supported to decrypt those objects. For more information

Expanding further and demonstrate how using Amazon S3 Access Points, governed by IAM roles and policies, can help enforce access controls on S3 buckets. This can enable you to ensure that data is available within selected VPCs. We also cover automating the infrastructure deployment using AWS CloudFormation, and testing the effectiveness of different Access Point policies combined with various AWS IAM roles.

AWS has a very good role in captivating the data security and management. As of now, we need a very robust and digital system for securing the private data. Mainly, AWS has now a greater market capitalization in compares to others. It really configuring the market a with a huge amount users.

MACHINE LEARNING

The two primary categories of machine learning methods are unsupervised and supervised.

For input and the intended output, supervised algorithms require supervision by a person with machine learning expertise. The method will then be used on fresh data after the model training is finished. Unsupervised algorithms don't need any data

training. They do, nonetheless, employ an iterative process. Deep Learning is the name of this strategy.

Artificial neural network is another name for unsupervised learning techniques. These networks are employed in situations where there is greater complexity because they are more versatile than supervised learning systems. Such neural networks advance by sifting through the training set and automatically identifying associations between both the dataset's parameters.

- Supervised machine learning algorithms are applied to the previously studied data in the past and then to new data. Such a system is able to provide outputs for any new input once sufficient training is done. This algorithm also compares its output with the correct, intended output and finds discrepancies, and then modifies the model accordingly.
- On the other hand, unsupervised machine learning algorithms have been when the information has neither been classified nor labeled. This category studies how systems can make inferences into a function that describes a hidden structure in unlabelled data. But there is a drawback, this system doesn't tell the right output, rather it explores the data and draws inferences from datasets.
- Semi-supervised machine learning algorithms lie in between the above-mentioned systems. This is because they use both labeled and unlabelled data for training purposes -a smaller amount of labeled data and a larger amount of unlabelled data in combination. This system uses techniques that are able to improve accuracy. Generally, semi-supervised learning is opted when the acquired dataset (labeled) requires skilled resources to train it.
- A learning method that engages with the surroundings is reinforcement machine learning. Generating activities, then looking for inconsistencies, is how it is done. Using this technique, behavioral patterns in the dataset can be contextually computed by machines and software agents to improve performance. The best way to proceed is then determined by analyzing simple reward responses. The aforementioned method is referred to as a reinforced signal.

Although it produces results more quickly and accurately, it may also require more time and resources during training. It is considerably more effective at processing massive datasets when deep learning, AI, and intelligent systems are combined.

CHAPTER 2

LITERATURE SURVEY

[1] I Made Ari Dwi Suta Atmaja, I Nyoman Gede Arya Astawa, Ni Wayan Wisswani, I Made Riyan Adi Nugroho, "Document Encryption Through Asymmetric RSA Cryptography", International Conference on Applied Science and Technology (iCAST), 2021

Cryptography is one method for securing digital documents and data. The most secure data security technique is asymmetric key cryptography. The RSA (Rivest-Shamir-Adleman) algorithm is one of the most widely used asymmetric cryptography algorithms. The most frequently attached document when sending e-mails is an encrypted document. The document formats are.docx,.pptx,.xlsx,.pdf,.jpg, and.mp4. A public key and a private key will be generated during the encryption process and can be sent individually by sending encrypted digital documents. The decryption of digital documents is performed from the receiving end of the file using a private key generated during the encryption process. The encrypted file is larger in size than the original file. Because it has been encoded in a different manner using the RSA algorithm. The longer and larger the input size, the longer it will take for encryption to complete.

[2] Yuxiang Lin, Xin Xia, Jingyi Yang, "Document Encryption Method with Mechanism of Enigma Machine", International Conference on Artificial Intelligence, Big Data and Algorithms (CAIBDA), 2021

Recent encryption methods heavily rely on the internet or computers' powerful calculation abilities. However, information is shared both online and offline, necessitating encrypting methods that are relatively easier to implement, such as paper or flash drives in some cases. To address the issue, this paper presents a method for document decryption and encryption based on the Enigma machine's mechanism. First, the document was read and processed so that only characters remain. The document is then encrypted into ciphertext using a given key. The key to decrypt the ciphertext was used to obtain the plaintext or original document. Finally, encryption of documents and decryption of the ciphertext was done to test the effectiveness of the proposed method.

[3] S. D. Sanap, Vijayshree More, "Analysis of Encryption Techniques for Secure Communication", International Conference on Emerging Smart Computing and Informatics (ESCI), 2021

The rapid and continuous increase in data exchange over networks and the cloud has provoked activities such as unauthorized access, illegal usage, and data alteration. Data security issues are becoming increasingly important as society transitions to a digital age. Encryption techniques are critical in overcoming this problem. These issues are addressed by modern encryption techniques. This paper gives an overview of encryption techniques that can be used to improve data security. This paper analyses the security features and stages involved in designing the most commonly used symmetric encryption algorithms such as DES, 3DES, and AES.

[4] Alpana A. Ingale, Sunil K. Moon, "E-Government Documents Authentication and Security by Utilizing Video Crypto-Steganography", IEEE Global Conference on Wireless Computing and Networking (GCWCN), 2019

E-government must be secure and reliable as technology advances, and users must follow procedures to ensure the security of their own transactions. This paper addresses the difficulties and roadblocks to improving information security in e-government. As a result, data hiding techniques have been found to be reliable for achieving security. Reversible data hiding (RDH) is a new technique that aids in the preservation of the cover image's quality. As a result, it is preferred over traditional data concealment techniques. In order to improve the results, the existing algorithm for image encryption and data hiding schemes is modified. To accomplish this, data is divided into 20 parts and data concealment is performed on each part.

[5] Bryan H. Wodi, Carson K. Leung, Alfredo Cuzzocrea, S. Sourav, "Fast Privacy-Preserving Keyword Search on Encrypted Outsourced Data", IEEE International Conference on Big Data (Big Data), 2020

To avoid privacy concerns, sensitive information should be encrypted before being stored on a cloud server. Document encryption makes it impossible for data owners to fetch documents using keyword searches, as they can with plain text documents. As a result, it is preferable to conduct a multi-keyword search on encrypted data. To achieve the goal presented in this paper a swift privacy-preserving model for keyword search on encrypted outsourced data. In particular, the model performs a keyword

search on encrypted files before determining its support for dynamic operations. It then sorts all relevant data documents based on the number of keywords matched for a given query based on the keyword search results.

[6] Aruna Guruvaya Mogarala, K. G. Mohan, "Security and Privacy Designs Based Data Encryption in Cloud Storage and Challenges: A Review", 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2018

Many businesses and individuals use cloud computing technology, and Cloud Service Providers (CSPs) store their user data in third-party storage. This poses a significant risk to user data security, and there is a high demand for user data security. Numerous data encryption methods have been suggested to protect user data from intruders. In this paper, an analysis of data security in cloud storage research with various output parameters is provided. The Bloom filter technique is used in the data encryption and key generation described in this paper. In the study of Wei Wu et al., encryption was performed with a computational time of 0.037s when the database dimension was 500 by using K-nearest neighbor classification.

[7] Daniel Mahardika Yusuf, De Rosal Ignatius Moses Setiadi, Eko Hari Rachmawanto, Christy Atika Sari, Rabei Raad Ali, "Dual Encryption Method for File Security", 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), 2020

To protect files, Daniel Mahardika Yusuf proposes a cryptography method that combines the RSA and RC4 algorithms. The RSA method's power is determined by the key quality, which is determined by computing two different prime numbers, p and q . The encryption result can encrypt the document byte but not beyond the byte limit. The RSA algorithm must be modified so that the key is more diverse and works on file bytes. Encrypt the file using the RSA algorithm in conjunction with the RC4 algorithm for increased file security. To evaluate the proposed algorithm's performance, entropy, avalanche effect counting, bit error ratio (BER), and time required for the computation were measured.

[8] Anil Kumar, Priyanka Dahiya, Garima, Sarvesh Tanwar, “A New Hybrid Approach for Encrypting XML Documents”, 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2021

Anil Kumar proposes a method for encrypting XML documents that uses elliptic curve cryptography (ECC) and chaos to solve the problem of allocating secret keys and to create a fast encryption mechanism. The ECC is used to create and allocate secret keys as well as to create digital signatures. The generated secret keys are then utilized to encrypt the XML documents using chaos. An experimental example demonstrates the feasibility of the proposed approach.

[9] Prachi More, Shubham Chandugade, Shaikh Mohammad Shafi Rafiq, Priya Pise, “Hybrid Encryption Techniques for Secure Sharing of a Sensitive Data for Banking Systems Over Cloud”, International Conference On Advances in Communication and Computing Technology (ICACCT), 2018

Because of recent advancements in cloud computing and the widespread use of computing systems, correspondence arrangement has increased the number of clients who exchange documents and sensitive data through the system; however, this delicate information necessitates special handling. By combining Attribute-Based Encryption and Byte Rotation Encryption Algorithm, this work demonstrates a security framework that can provide security and integrity when trading sensitive data through the cloud or correspondence systems. The goal of the work is to create a simple platform that can provide integrity, protection, and performance for peer-to-peer data exchange. The proposed framework makes use of symmetric cryptography. End-to-end visibility, safety, and consistency must be provided by data exchange.

[10] Jie Tong, Yihong Long, Quan Liu, “A File Encryption System Based on Attribute Based Encryption”, 17th International Conference on Computational Intelligence and Security (CIS), 2022

Attribute Based Encryption (ABE) is a type of public-key cryptosystem that is widely used to secure file data and achieve fine-grained file sharing. However, existing attribute-based encryption algorithms are extremely complex and require a large amount of additional cryptographic data, making them difficult to implement. To address this issue, this paper proposes an attribute-based data encryption scheme for secure file storage and access control. This system is simple to set up and reliable to

operate. The attribute private key is split into a server-side secret share and a user-side secret share in this file encryption system. When a user is granted access to a file, the client and server decrypt it collaboratively using the user-side secret shares and server-side secret shares of the user's attribute private keys.

2.1 INFERENCES FROM LITREATURE SURVEY

From the above-mentioned literature works, it is clear that there has been effective research on Private cloud systems and many models have been proposed. It is evident that the above-mentioned systems have their own pros and cons. While some of the recent works involve hybrid technologies and provide better accuracies, they are still far from what is needed. With higher accuracy, comes the need for low computational costs, high processing speed, and most of all, the convenience of use.

2,2 OPEN PROBLEMS IN EXISTING SYSTEM

While much effort has been directed at securing network communications, security of stored data remains a largely neglected area both in the development and use of such systems. Nonetheless, various implementations of encrypting file systems exist. As observed the current encryption software doing the encryption task is all very complicated in its functionality. The method of encryption and key generation of the current system for a new user to understand is complex in nature. The existing systems fail to provide a higher level of security and make private documents more prone to theft and attacks.

CHAPTER 3 REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT

FEASIBILITY STUDY

The feasibility of the project is server performance increase in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis, the feasibility study of the proposed system is to be carried out. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- Economical feasibility
- Technical feasibility
- Operational feasibility

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of funds that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have modest requirements, as only minimal or null changes are required for implementing this system.

OPERATIONAL FEASIBILITY

The aspect of the study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system

3.2 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

Hardware specifications:

- Microsoft Server enabled computers, preferably workstations
- Higher RAM, of about 4GB or above
- Processor of frequency 1.5GHz or above

Software specifications:

- Python 3.6 and higher
- Anaconda software

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

4.1 SELECTED METHODOLOGY OR PROCESS MODEL

The most complete and widely used cloud platform in the world, Amazon Web Services (AWS), provides over 200 fully functional services from data centers across the world. Millions of clients use AWS to save costs, increase agility, and accelerate innovation, including the largest corporations, most successful governmental organizations, and the fastest-growing startups.

Compared to other cloud providers, AWS offers a significantly greater number of services and features within those services, ranging from infrastructure technologies like compute, storage, and databases to cutting-edge technologies like artificial intelligence, machine learning, data lakes, and the Internet of Things. As a result, moving your current applications to the cloud is quicker, simpler, and more cost-effective, and you can construct almost anything you can think of.

AWS provides the most comprehensive functionality among such services. For instance, AWS has the greatest selection of databases that are created specifically for certain applications, allowing you to select the finest tool for the task at the best price and performance.

The most adaptable and safe cloud computing platform currently available is AWS, which is designed to be both. The security requirements for the military, major banks, and other highly sensitive organizations are met by our core infrastructure. With 230 security, compliance, and governance services and capabilities, this is supported by a comprehensive collection of cloud security tools. In addition to offering the option to

encrypt customer data across all 117 AWS services that host it, AWS supports 98 security standards and compliance certifications.

- **Implementing Machine Learning**

Encrypting data at rest and in transit (both structured and non-structured) is already common practice within companies, to protect confidential, secret and proprietary information. Encrypting data while in use, however, is a less common practice. Data in use is data that is stored in a non-persistent digital state and/or that is being processed, like in the lifecycle of a machine learning (ML) model.

Nowadays big cloud operators, such as Google, AWS, and Microsoft, and startups alike are offering Machine Learning as a Service (MLaaS). These services help small companies that lack the ML expertise or required infrastructure to build a predictive model with the data from the small companies. These data are valuable and sensitive. The customers get their services via API calls. Service providers don't want to open up about their model, which are black boxes to the customers. And because of the data sensitivity, customers may not be interested to share their raw data through API calls. Here comes the trust worthy "Encrypted Machine Learning" concept that helps protect the data and the model by encrypting it. Instead of merely providing MLaaS that might be leaky, service providers can introduce EMLaaS (Encrypted Machine Learning as a Service) to assure customers about their data security.

EMLaaS can help small companies to ensure data security and create amazing AI driven solutions with minimal cost. In an era when aggregated data has the power to propel research and innovation in many fields, encrypted and private AI could help protect data privacy, build trust, and enable different parties to collaborate together.

- **Setting up AWS**

1. the main page for Amazon Web Services (AWS).
2. Select AWS Account Creation.
3. Choose Sign in to the Console if you recently logged in to AWS. Choose Sign in to a different account first, then select Create a new AWS account if create a new AWS account isn't displayed.
4. Enter your email address in Root user email address, make any necessary changes to the AWS account name, and then select Verify email address. This address will get an email from AWS with a verification code.

- **Setup Billing**

Enter your payment method details on the billing information page, then click Verify and Add.

You must enter your CVV as part of the verification process when creating an Amazon Internet Services Private Limited (AISPL) account in India. Depending on your bank, you might also need to enter a one-time password. As part of the verification process, AISPL deducts two Indian Rupees (INR) from your payment method. Following completion of the verification, AISPL returns the two INR.

Select Utilize a new address if you want to use a different billing address for your AWS billing information. Select Verify and Continue after that.

- **Setup S3 Bucket in your nearest Region**

Across different AWS Regions, clients in Amazon Virtual Private Clouds (VPCs) must read and write data. Requests to S3 are automatically forwarded to the AWS Region with the lowest latency when you connect to your S3 Multi-Region Access Point from within a VPC. These clients can now access data using a single global S3 endpoint, including using AWS Private Link for S3, and no longer need to know which S3 bucket or AWS Region it is located in.

- **Starting WebApp Frontend**

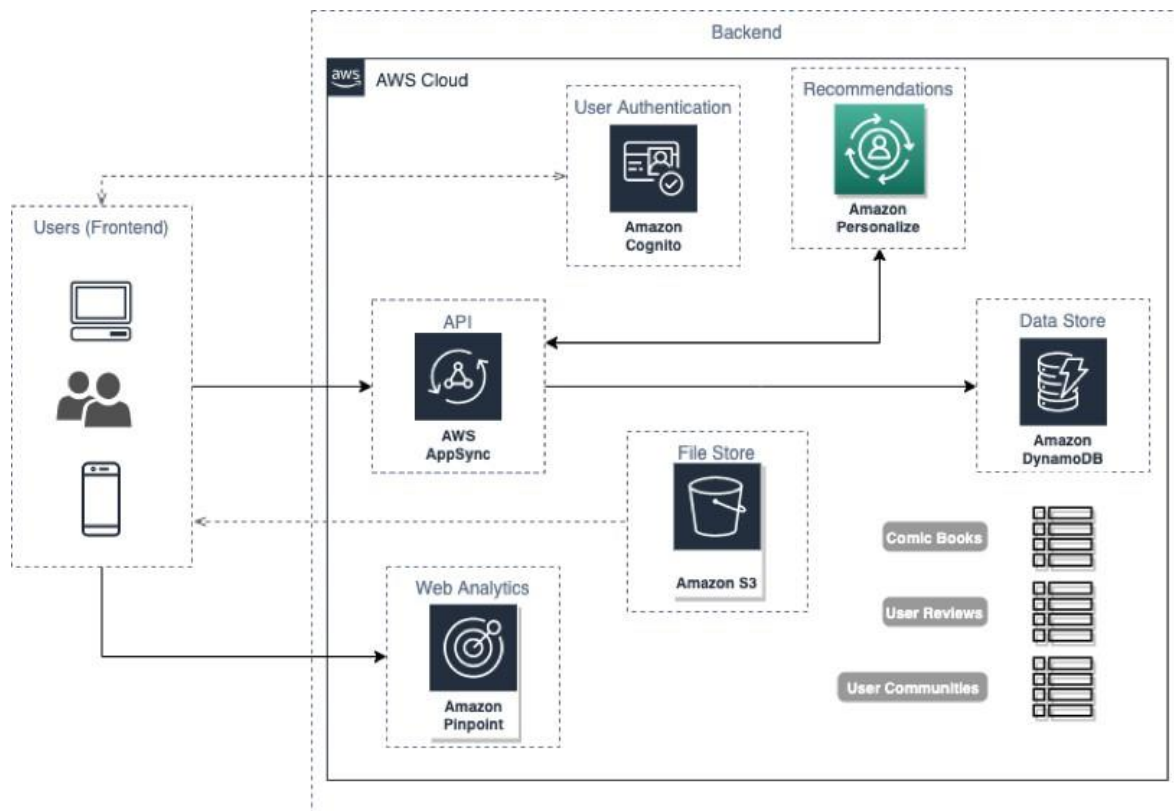


Fig 3

The programme can be divided into two parts: the frontend, which allows users to interact with application resources, and the backend, which houses all of your developed resources.

You can utilize many frontends for your application to provide users the greatest experience possible and to benefit from the special capabilities that each platform provides. For instance, you may design several layouts that make use of the wider screen space available in desktop web browsers or mobile applications that make use of the cameras and GPS features of mobile devices.

We take the example of a hypothetical online comic book shop as a guide to examine the key components of a standard web application. This website's function is to let users buy and sell their preferred comic books, receive alerts when new comics are published, and form communities with other users who share their interests.

However, we do establish a connection between the business need or use case and the technology that may be used to make it a reality. We don't go into implementing

all these things. We use AWS Amplify, a set of tools and services that let developers of mobile and web apps create safe, scalable cloud-powered applications fast and easily, to assist us along the way.

The speed at which you iterate and distribute new versions of the programme to your users is crucial while developing an application. The greatest user experience is ensured by having a consistent and seamless workflow from development to deployment, which enables considerably faster iterations and shorter time between releases. A combination of guidelines, tools, and best practises known as continuous integration and continuous deployment, or CI/CD, enables teams to release code updates more quickly and effectively using CI/CD pipelines.

The graphic below depicts a typical development workflow from development to testing and then to production utilizing Amplify to quickly handle various environments and development steps while keeping track of development using source control.

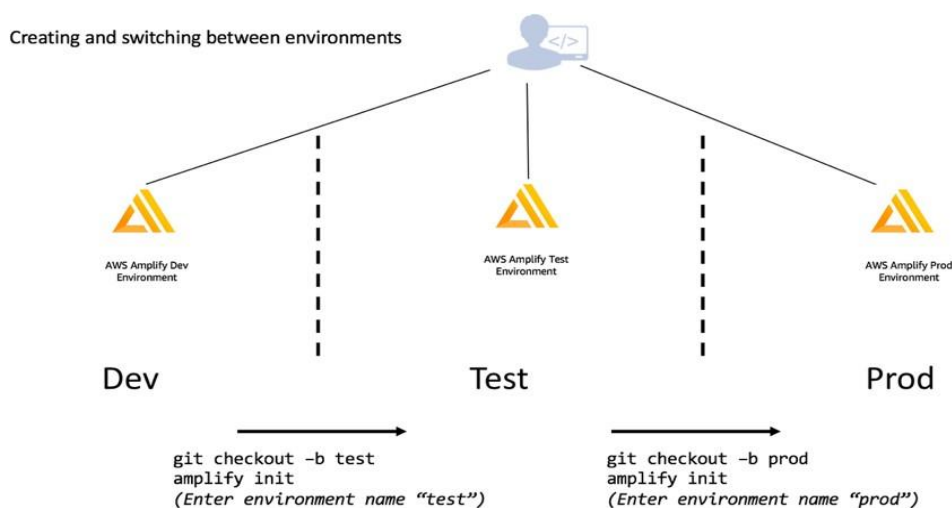


Fig 4

- **Upload Document**

1. Upload multiple files to AWS Cloud Shell using Amazon S3

a. In AWS Cloud Shell, create an S3 bucket by running the following s3 command:
`aws s3api create-bucket --bucket your-bucket-name --region us-east-1`

b. Next, you need to upload the files in a directory from your local machine to the bucket. You have two options for uploading files:

- Drag and drop to upload files and folders to a bucket using the AWS Management Console.
- AWS CLI: Use the command line to upload files and folders to the bucket using the version of the programme that is installed on your local computer.

c. To synchronize the directory in the shell environment with the contents of the S3 bucket, go back to the AWS Cloud Shell command line and type the following command: `s3:/your-bucket-name, aws s3 sync`

▪ Download Document

1. To sync an S3 bucket with the contents of the current directory in the shell environment, use the AWS Cloud Shell command line and the `aws s3` command:
`S3:/your-bucket-name. aws s3 sync`

2. The bucket's contents must now be downloaded to your local computer. You must use the AWS CLI tool that is installed locally because the Amazon S3 console does not support downloading several objects at once. Use the command: `aws s3 sync s3:/your-bucket-name.`

• Getting Secret AWS Access Keys

To get your access key ID and secret access key:

1. Go to <https://console.aws.amazon.com/iam> to access the IAM console.
2. Click Users on the navigation menu.
3. Select a user name for IAM (not the check box).
4. Select Create access key from the Security Credentials page after opening it.
5. Select Show to display the new access key. Your credentials look like these:
6. Key ID for access: AKIAIOSFODNN7EXAMPLE

7. Access code: bPxRfiCYEXAMPLEKEY/K7MDENG/wJalrXUtnFEMI
8. Select Download.csv file to download the key pair. Keep the.csv file with the keys in a safe place.

- **Connecting To S3 instance using BOTO3 client**

Boto3 has both low-level clients and higher-level resources. For Amazon S3, the higher-level resources are the most similar to Boto 2.x's **s3** module:

```
import boto3
s3 = boto3.resource('s3')
```

Creating a bucket in Boto 2 and Boto3 is very similar, except that in Boto3 all action parameters must be passed via keyword arguments and a bucket configuration must be specified manually:

```
s3.create_bucket(Bucket='mybucket')
s3.create_bucket(Bucket='mybucket', CreateBucketConfiguration={
    'LocationConstraint': 'us-west-1'})
```

- **Encryption algorithm**

1. To use Amazon S3 AES-256 encryption, the first thing you need to do is navigate to the S3 service.
2. Go to the bucket that you want to secure with AES 256 encryption.
3. Once you're inside your S3 bucket, navigate to the Properties tab and go to the Default encryption section. Most likely, it will be marked as Disabled. Meaning, no encryption is currently used. Click that Disabled label to edit the default encryption setting.

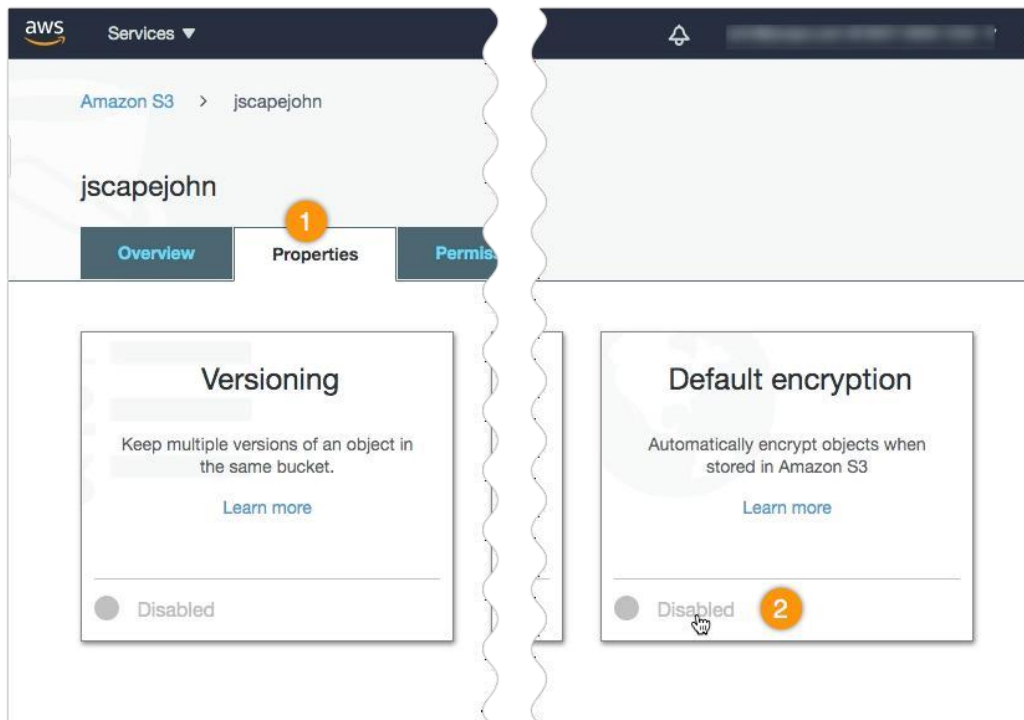


Fig 5

4. Next, select AES-256 as your default encryption and then click Save.

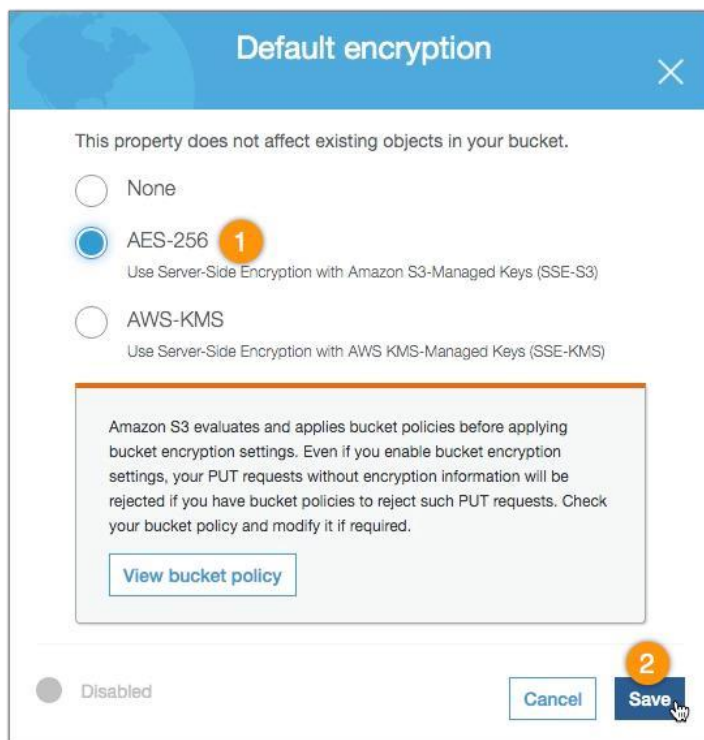


Fig 6

5. That Disabled label you saw earlier should now be replaced with AES-256. That means, any file you upload to this S3 bucket moving forward will already be encrypted with Amazon S3 AES-256 encryption.

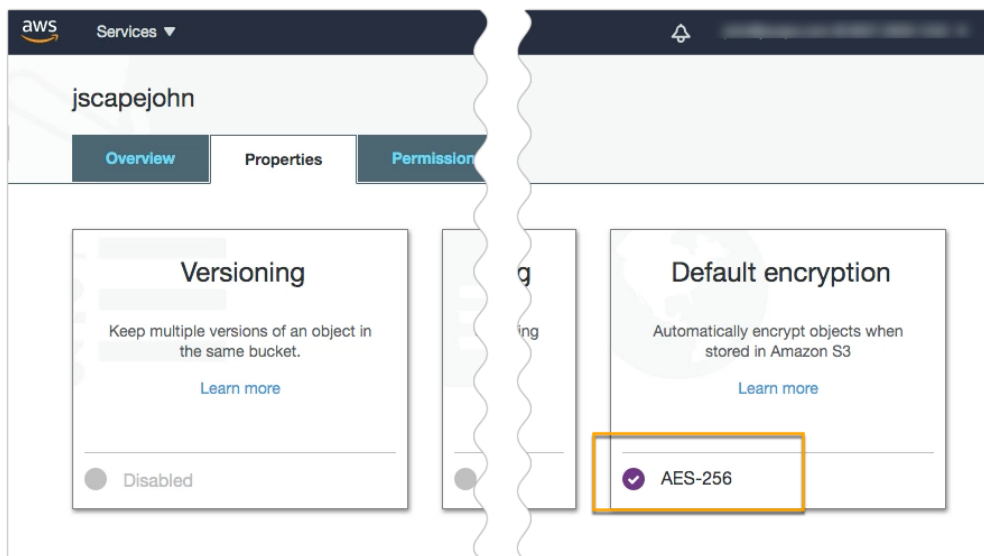


Fig 7

6. Scroll down to the Authentication section and then click the Use encryption checkbox.
7. Select the AES-256 option and then click the OK button.
8. With that, this S3 trading partner should now be ready to use S3 AES-256 encryption.

IMPLEMENTATION OF NLP

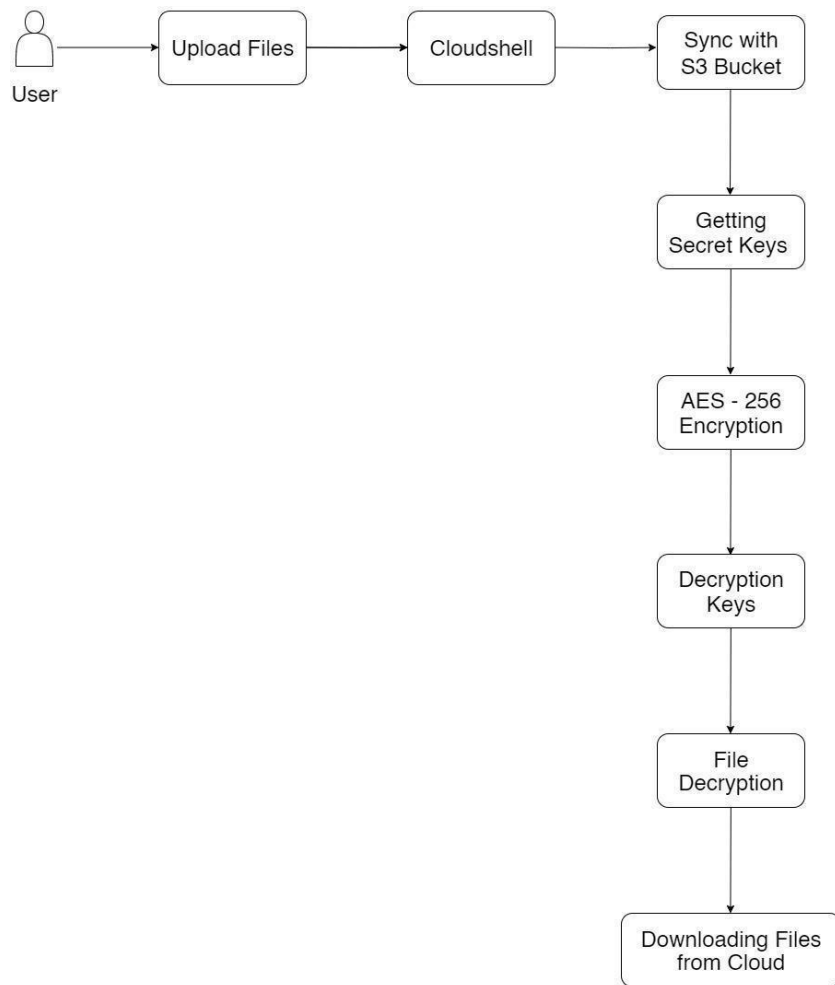
NATURAL LANGUAGE PROCESSING(NLP)

Data Cleaning such as removing clutter and unnecessary punctuation would be taken off, Feature Engineering would be performed for enhancement. This includes removing stop words, punctuation, and stemming. This process will construct a graph with sentences as the vertices. Importing necessary libraries is performed. This library creates a summary of the supplied information within the word limit. With the help of Natural language processing techniques, the application which extracts the names of people, places, and other entities from text, the main goal being to get a reduced version of it that retains the most important information. It is followed by Exploratory Data Analysis. Exploratory data analysis (EDA) is often a necessary task in uncovering hidden patterns, detecting outliers, and identifying important variables and any anomalies in data.

TF-IDF

At this stage, a dynamic Script for the Tf-Idf approach is written. Term frequency-inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection. TF-IDF is word frequency scores that aim to emphasize phrases that are more interesting, e.g., common in a text but not across texts, without delving into the arithmetic. The TF-IDF Vectorizer tokenizes texts, learns vocabulary, inverts frequency weightings, and allows encoding new ones. It provides information on a word frequency in the documents. Higher the TF- IDF score of a term which is computed using the above equations represents more relevance in a document.

4,2 ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM



System Architecture

4.3 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED MODEL/SYSTEM

Anaconda is an open-source package manager for Python and R. It is the most popular platform among data science professionals for running Python and R implementations. There are over 300 libraries in data science, so having a robust distribution system for them is a must for any professional in this field. Anaconda simplifies package deployment and management. On top of that, it has plenty of tools that can help you with data collection through artificial intelligence and machine

learning algorithms. With Anaconda, you can easily set up, manage, and share Conda environments. Moreover, you can deploy any required project with a few clicks when you're using Anaconda. There are many advantages to using Anaconda and the following are the most prominent ones among them: Anaconda is free and open-source. This means you can use it without spending any money. In the data science sector, Anaconda is an industry staple. It is open-source too, which has made it widely popular. If you want to become a data science professional, you must know how to use Anaconda for Python because every recruiter expects you to have this skill. It is a must-have for data science.

It has more than 1500 Python and R data science packages, so you don't face any compatibility issues while collaborating with others. For example, suppose your colleague sends you a project which requires packages called A and B but you only have package A. Without having package B, you wouldn't be able to run the project. Anaconda mitigates the chances of such errors. You can easily collaborate on projects without worrying about any compatibility issues. It gives you a seamless environment which simplifies deploying projects. You can deploy any project with just a few clicks and commands while managing the rest. Anaconda has a thriving community of data scientists and machine learning professionals who use it regularly. If you encounter an issue, chances are, the community has already answered the same. On the other hand, you can also ask people in the community about the issues you face there, it's a very helpful community ready to help new learners. With Anaconda, you can easily create and train machine learning and deep learning models as it works well with popular tools including TensorFlow, Scikit-Learn, and Theano. You can create visualizations by using Bokeh, Holo views, Matplotlib, and Data shader while using Anaconda.

How to Use Anaconda for Python

Now that we have discussed all the basics in our Python Anaconda tutorial, let's discuss some fundamental commands you can use to start using this package manager.

Listing All Environments

To begin using Anaconda, you'd need to see how many Conda environments are present in your machine.

`conda env list`

It will list all the available Conda environments in your machine.

Creating a New Environment

You can create a new Conda environment by going to the required directory and use this command:

```
conda create -n <your_environment_name>
```

You can replace `<your_environment_name>` with the name of your environment. After entering this command, conda will ask you if you want to proceed to which you should reply with y:

proceed ([y])/n)?

On the other hand, if you want to create an environment with a particular version of Python, you should use the following command:

```
conda create -n <your_environment_name> python=3.6
```

Similarly, if you want to create an environment with a particular package, you can use the following command:

```
conda create -n <your_environment_name> pack_name
```

Here, you can replace `pack_name` with the name of the package you want to use.

If you have a `.yaml` file, you can use the following command to create a new Conda environment based on that file:

```
conda env create -n <your_environment_name> -f <file_name>.yaml
```

We have also discussed how you can export an existing Conda environment to a `.yaml` file later in this article.

Activating an Environment

You can activate a Conda environment by using the following command:

```
conda activate <environment_name>
```

You should activate the environment before you start working on the same. Also, replace the term `<environment_name>` with the environment name you want to activate. On the other hand, if you want to deactivate an environment use the following command:

```
conda deactivate
```

Installing Packages in an Environment

Now that you have an activated environment, you can install packages into it by using the following command:

```
conda install <pack_name>
```

Replace the term <pack_name> with the name of the package you want to install in your Conda environment while using this command.

Updating Packages in an Environment

If you want to update the packages present in a particular Conda environment, you should use the following command:

```
conda update
```

The above command will update all the packages present in the environment. However, if you want to update a package to a certain version, you will need to use the following command:

```
conda install <package_name>=<version>
```

Exporting an Environment Configuration

Suppose you want to share your project with someone else (colleague, friend, etc.). While you can share the directory on Github, it would have many Python packages, making the transfer process very challenging. Instead of that, you can create an environment configuration .yaml file and share it with that person. Now, they can create an environment like your one by using the .yaml file.

For exporting the environment to the .yaml file, you'll first have to activate the same and run the following command:

```
conda env export ><file_name>.yaml
```

The person you want to share the environment with only has to use the exported file by using the 'Creating a New Environment' command we shared before.

Removing a Package from an Environment

If you want to uninstall a package from a specific Conda environment, use the following command:

```
conda remove -n <env_name><package_name>
```

On the other hand, if you want to uninstall a package from an activated environment, you'd have to use the following command:

```
conda remove <package_name>
```

Deleting an Environment

Sometimes, you don't need to add a new environment but remove one. In such cases, you must know how to delete a Conda environment, which you can do so by using the following command:

```
conda env remove -name <env_name>
```

The above command would delete the Conda environment right away.

4.4 PROJECT MANAGEMENT PLAN

Introduction:	
Literature Survey:	
System Design:	
System Implementation:	

CHAPTER-5

IMPLEMENTATION OF MACHINE LEARNING

- **Implementing Machine Learning**

Encrypting data at rest and in transit (both structured and non-structured) is already common practice within companies, to protect confidential, secret and proprietary information. Encrypting data while in use, however, is a less common practice. Data in use is data that is stored in a non-persistent digital state and/or that is being processed, like in the lifecycle of a machine learning (ML) model.

Nowadays big cloud operators, such as Google, AWS, and Microsoft, and startups alike are offering Machine Learning as a Service (MLaaS). These services help small companies that lack the ML expertise or required infrastructure to build a predictive model with the data from the small companies. These data are valuable and sensitive. The customers get their services via API calls. Service providers don't want to open up about their model, which are black boxes to the customers. And because of the data sensitivity, customers may not be interested to share their raw data through API calls. Here comes the trust worthy "Encrypted Machine Learning" concept that helps protect the data and the model by encrypting it. Instead of merely providing MLaaS that might be leaky, service providers can introduce EMLaaS (Encrypted Machine Learning as a Service) to assure customers about their data security.

EMLaaS can help small companies to ensure data security and create amazing AI driven solutions with minimal cost. In an era when aggregated data has the power to propel research and innovation in many fields, encrypted and private AI could help protect data privacy, build trust, and enable different parties to collaborate together.

Machine learning is a subset of AI. It allows computer to learn and predict. A Machine Learning system learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately. Machine learning enables a machine to automatically learn from data, improve performance from experience, and predict things without being explicitly programmed.

A subset of machine learning is closely related to computational statistics, which focuses on making predictions using computers; but not all machine learning is statistical learning. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning. Some implementations of machine learning use data and neural networks in a way that mimics the working of a biological brain. In its application across business problems, machine learning is also referred to as predictive analytics.

The need for machine learning is increasing day by day. The reason behind the need for machine learning is that it is capable of doing tasks that are too complex for a person to implement directly. As a human, we have some limitations as we cannot access the huge amount of data manually, so for this, we need some computer systems and here comes the machine learning to make things easy for us. We can train machine learning algorithms by providing them the huge amount of data and let them explore the data, construct the models, and predict the required output automatically. The performance of the machine learning algorithms depends on the amount of data, and it can be determined by the cost function. With the help of machine learning, we can save both time and money.

The importance of machine learning can be easily understood by its use cases, currently, machine learning is used in self-driving cars, cyber fraud detection, face recognition, and friend suggestion by face book, etc. Various top companies such as Netflix and Amazon have build machine learning models that are using a vast amount of data to analyze the user interest and recommend product accordingly.

- ☐ Rapid increment in the production of data.
- ☐ Solving complex problems, which are difficult for a human.

- Decision making in various sector including finance.
- Finding hidden patterns and extracting useful information from data.

5.1 History of ML

Before some years (about 40-50 years), machine learning easy science fiction, but today it is the part of our daily life. Machine learning is making our day to day life easy from self-driving cars to amazon virtual assistant “Alexa”, However, the idea behind machine learning is so old and has a long history. Below some milestones are given which have occurred in the history of machine learning.

The early history of Machine Learning (Pre-1940):

In 1834, Charles Babbage, the father of the computer, conceived a device that could be programmed with punch cards. However, the machine was never built, but all modern computers rely on its logical structure. In 1936, Alan Turing gave a theory that how a machine can determine and execute a set of instructions.

Computer machinery and intelligence:

In 1950, Alan Turing published a seminal paper, “Computer Machinery and Intelligence,” on the topic of artificial intelligence. In his paper, he asked, “can machines think?”.

Machine intelligence in games:

In 1952, Arthur Samuel, who was the pioneer of machine learning, created a program that helped an IBM computer to play a checkers game. It performed better more it played.

Machine Learning from theory to reality:

In 1956, the first neural network was applied to a real-world problem to remove echoes over phone lines using an adaptive filter. In 1985, Terry Sejnowski and Charles Rosenberg invented a neural network NET talk, which was able to teach itself how to correctly pronounce 20,000 words in one week. In 1997, the IBM’s Deep blue intelligent computer won the chess game against the chess expert Garry Kasparov, and it became the first computer which had beaten a human chess expert.

Machine Learning at 21st century:

In the year 2006, computer scientist Geofferey Hinton has given a new name to

neural net research as “deep learning” and nowadays, it has become one of the most trending technologies. In 2012, Google created a deep neural network which learned to recognize the image of humans and cats in YouTube videos. Now machine learning has got a great advancement in its research, and it is present everywhere around us, such as self-driving cars, Amazon Alexa, Catboats, recommender system, and many more. It includes Supervised, unsupervised, and reinforcement learning with clustering, classification, decision tree, SVM algorithms, etc. Modern machine learning models can be used for making various predictions, including weather prediction, disease prediction, stock market analysis, etc.

5.2 Types of ML

At a broad level, machine learning can be classified into three types:

1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement learning

☐ Supervised Learning

Supervised learning is a type of machine learning method in which we provide sample labelled data to the machine learning system in order to train it, and on that basis, it predicts the output. The system creates a model using labelled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is predicting the exact output or not.

The goal of supervised learning is to map input data with the output data. The supervised learning is based on supervision, and it is the same as when a student learns things in the supervision of the teacher. The example of supervised learning is spam filtering.

☐ Classification

☐ Regression

☐ Unsupervised Learning

Unsupervised learning is a learning method in which a machine learns without any supervision. The training is provided to the machine with the set of data that has not been labelled, classified, or categorized, and the algorithm needs to act on that data without any supervision. The goal of unsupervised learning is to restructure the input data into new features or a group of objects with similar patterns.

In unsupervised learning, we don't have a predetermined result. The machine tries to find useful insights from the huge amount of data. It can be further classified into two categories of algorithms:

- ☐ Clustering
- ☐ Association

- ☐ Reinforcement Learning

Reinforcement learning is a feedback-based learning method, in which a learning agent gets a reward for each right action and gets a penalty for each wrong action. The agent learns automatically with these feedbacks and improves its performance. In reinforcement learning, the agent interacts with the environment and explores it. The goal of an agent is to get the most reward points, and hence, it improves its performance.

The robotic dog, which automatically learns the movement of his arms, is an example of Reinforcement learning.

5.3 Features of ML

- ☐ Machine learning uses data to detect various patterns in a given dataset.
- ☐ It can learn from past data and improve automatically.
- ☐ It is a data-driven technology.
- ☐ Machine learning is much similar to data mining as it also deals with the huge amount of the data.
- ☐ Application of ML

Machine learning is a buzzword for today's technology, and it is growing very rapidly day by day. We are using machine learning in our daily life even without knowing it such as Google Maps, Google assistant, Alexa, etc. Below are some

most trending real world applications of Machine Learning:

1. Image Recognition
2. Stock Market Trading
3. Online Fraud Detection
4. Self-Driving Cars
5. Traffic Prediction

Speech Recognition

IMPLEMENTATION OF NLP

NATURAL LANGUAGE PROCESSING(NLP)

Data Cleaning such as removing clutter and unnecessary punctuation would be taken off, Feature Engineering would be performed for enhancement. This includes removing stop words, punctuation, and stemming. This process will construct a graph with sentences as the vertices. Importing necessary libraries is performed. This library creates a summary of the supplied information within the word limit. With the help of Natural language processing techniques, the application which extracts the names of people, places, and other entities from text, the main goal being to get a reduced version of it that retains the most important information. It is followed by Exploratory Data Analysis. Exploratory data analysis (EDA) is often a necessary task in uncovering hidden patterns, detecting outliers, and identifying important variables and any anomalies in data.

TF-IDF

At this stage, a dynamic Script for the Tf-Idf approach is written. Term frequency-inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection. TF-IDF is word frequency scores that aim to emphasize phrases that are more interesting, e.g., common in a text but not across texts, without delving into the arithmetic. The TF-IDF Vectorizer tokenizes texts, learns vocabulary, inverts frequency weightings, and allows encoding new ones. It provides information on a word frequency in the documents. Higher the TF- IDF score of a term which is computed using the above equations represents more relevance in a document.

PROGRAM LANGUAGE Python concepts

- ❑ Opensource general-purpose language.
- ❑ Object Oriented, Procedural, Functional
- ❑ Easy to interface with C/Obj/Java/Fortran
- ❑ Easy to interface with C++ (via SWIG)
- ❑ Great interactive environment

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- ❑ Python is Interpreted: Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- ❑ Python is Interactive: We can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- ❑ Python is Object-Oriented: Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- ❑ Python is a Beginner's Language: Python is a great language for the beginner- level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games

History of Python

- ❑ Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

- Python is derived from many other languages, including ABC, Modula-3, C++, Algol-68, Small Talk, and UNIX shell and other scripting languages.
- Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).
- Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Features of Python

Python's features include :--

- Easy-to-learn

Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

- Easy-to-read

Python code is more clearly defined and visible to the eyes. Easy-to-maintain:

Python's source code is fairly easy to maintain.

- A broad standard library

Python's bulk of the library is very portable and crosses- platform compatible on UNIX, Windows, and Macintosh.

- Interactive Mode

Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

- Portable

Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

☐ Extendable

We can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

☐ Databases

Python provides interfaces to all major commercial databases.

☐ GUI Programming

Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

☐ Scalable

Python provides a better structure and support for large programs than shell scripting.

☐ A part from the above-mentioned features, Python has a big list of good features, few are listed below:

☐ Dynamic vs. Static

Types Python is a dynamic-typed language. Many other languages are static typed, such as C/C++ and Java. A static typed language requires the programmer to explicitly tell the computer what type of “thing” each data value is.

For example, in C if you had a variable that was to contain the price of something, you would have to declare the variable as a “float” type. This tells the compiler that the only data that can be used for that variable must be a floating point number, i.e, a number with a decimal point.

If any other data value was assigned to that variable, the compiler would give an

error when trying to compile the program. Python, however, doesn't require this. You simply give your variables names and assign values to them. The interpreter takes care of keeping track of what kinds of objects your program is using. This also means that you can change the size of the values as you develop the program. Say you have another decimal number (a.k.a. a floating point number) you need in your program. With a static typed language, you have to decide the memory size the variable can take when you first initialize that variable. A double is a floating point value that can handle a much larger number than a normal float (the actual memory sizes depend on the operating environment). If you declare a variable to be a float but later on assign a value that is too big to it, your program will fail; you will have to go back and change that variable to be a double. For example, say you are dividing two numbers. One is a floating point number and one is an integer. Python realizes that it's more accurate to keep track of decimals so it automatically calculates the result as a floating point number.

□ Variables

Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory. Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory. Therefore, by assigning different data types to variables, you can store integers, decimals or characters in these variables.

□ Standard Data Types

The data stored in memory can be of many types. For example, a person's age is stored as a numeric value and his or her address is stored as alphanumeric characters. Python has various standard data types that are used to define the operations possible on them and the storage method for each of them.

Python has five standard data types:

- Numbers
- String

- List
- Tuple
- Dictionary

☐ Python Numbers

Number data types store numeric values. Number objects are created when you assign a value to them.

☐ Python Strings:

Strings in Python are identified as a contiguous set of characters represented in the quotation marks. Python allows for either pairs of single or double quotes. Subsets of strings can be taken using the slice operator ([] and [:]) with indexes starting at 0 in the beginning of the string and working their way from -1 at the end.

☐ Python Lists

Lists are the most versatile of Python's compound data types. A list contains items separated by commas and enclosed within square brackets ([]). To some extent, lists are similar to arrays in C. One difference between them is that all the items belonging to a list can be of different data type. The values stored in a list can be accessed using the slice operator ([] and [:]) with indexes starting at 0 in the beginning of the list and working their way to end -1. The plus (+) sign is the list concatenation operator, and the asterisk (*) is the repetition operator.

☐ Python Tuples

A tuple is another sequence data type that is similar to the list. A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses. The main differences between lists and tuples are: Lists are enclosed in brackets ([]) and their elements and size can be changed, while tuples are enclosed in parentheses (()) and cannot be updated. Tuples can

be thought of as read-only lists.

☐ Python Dictionary

Python's dictionaries are kind of hash table type. They work like associative arrays or hashes found in Perl and consist of key-value pairs. A dictionary key can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object. Dictionaries are enclosed by curly braces ({}) and values can be assigned and accessed using square braces ([]).

☐ Different modes in python has two basic modes:

☐ Normal

☐ Interactive

The normal mode is the mode where the scripted and finished .py files are run in the Python interpreter. Interactive mode is a command line shell which gives immediate feedback for each statement, while running previously fed statements in active memory. As new lines are fed into the interpreter, the fed program is evaluated both in part and in whole.

☐ NumPy

NumPy main object is the homogeneous multidimensional array. It is a table of elements (usually numbers), all of the same type, indexed by a tuple of positive integers. In NumPy dimensions are called axes. The number of axes is rank.

☐ Offers Mat lab capabilities within Python

☐ Fast array operations

☐ 2D arrays, multi-D arrays, linear algebra etc.

☐ Mat plot lib

High quality plotting library.

☐ Python modules

Python allows us to store our code in files (also called modules). This is very useful for more serious programming, where we do not want to retype a long function definition from the very beginning just to change one mistake. In doing this, we are essentially defining our own modules, just like the modules defined already in the Python library. To support this, Python has a way to put definitions in a file and use them in a script or in an interactive instance of the interpreter. Such a file is called a module; definitions from a module can be imported into other modules or into the main module.

□ Testing code

As indicated above, code is usually developed in a file using an editor. To test the code, import it into a Python session and try to run it. Usually there is an error, so you go back to the file, make a correction, and test again. This process is repeated until you are satisfied that the code works. The entire process is known as the development cycle. There are two types of errors that you will encounter. Syntax errors occur when the form of some command is invalid. This happens when you make typing errors such as misspellings, or call something by the wrong name, and for many other reasons.

Python will always give an error message for a syntax error.

Functions of Python

It is possible, and very useful, to define our own functions in Python. Generally speaking, if you need to do a calculation only once, then use the interpreter. But when you or others have needed to perform a certain type of calculation many times, and then define a function. You use functions in programming to bundle a set of instructions that you want to use repeatedly or that, because of their complexity, are better self-contained in a sub-program and called when needed. That means that a function is a piece of code written to carry out a specified task. To carry out that specific task, the function might or might not need multiple inputs. When the task is carried out, the function can or cannot return one or more values. There are three types of functions in python `help()`, `min()`, `print()`.

□ Python Namespace:

Generally speaking, a namespace (sometimes also called a context) is a naming system for making names unique to avoid ambiguity. Everybody knows a name spacing system from daily life, i.e, the naming of people in first name and family name (surname). An example is a network: each network device (workstation, server, printer) needs a unique name and address. Yet another example is the directory structure of file systems. The same file name can be used in different directories; the files can be uniquely access via the path names. Many programming languages use name spaces or contexts for identifiers. An identifier defined in a name space is associated with that namespace. This way, the same identifier can be independently defined in multiple namespaces. (Like the same file names in different directories) Programming languages, which support namespaces, may have different rules that determine to which namespace an identifier belongs. Namespaces in Python are implemented as Python dictionaries; this means it is a mapping from names (keys) to objects (values). The user doesn't have to know this to write a Python program and when using namespaces. Some namespaces in Python--Global names of a module, local names in a function or method invocation, built-in names this namespace contains built-in functions (e.g. `abs()`, `cmp()`.) and built-in exception names.

□ Garbage Collection

Garbage Collector exposes the underlying memory management mechanism of Python, The automatic garbage collector. The module includes functions for controlling how the Collector operates and to examine the objects known to the system, either pending collection or stuck in reference cycles and unable to be freed.

□ Python-Data Collection

generally sensors and cameras collect the data send it to the Storage area, whereas in the simulator. Data is collected while driving manually and then record the whole surroundings, and then trained the model on that recorded video, so basically the data sets we used here are the video recordings while driving a car. We can use the recording and they don't have any specific format for the storage so we normally store them in the hard disk.

□ Python-Data Base Communication

Connector/Python provides a `connect ()` call used to establish connections to the MySQL server. The following sections describe the permitted arguments for `connect ()` and describe how to use option files that supply additional arguments. A database is an organized collection of data. The data are typically organized to model aspects of reality in a way that 35 supports processes requiring this information. The term "database" can both refer to the data themselves or to the database management system. The Database management system is a software application for the interaction between user's database itself. Databases are popular for many applications, especially for use with web applications or programs. Users don't have to be human users. They can be other programs and applications as well. We will learn how Python or better a Python program can interact as a user of an SQL database. This is an introduction into using SQLite and MySQL from Python. The Python standard for database interfaces is the Python DB-API, which is used by Python's database interfaces. The DB-API has been defined as a common interface, which can be used to access relational databases. In other words, the code in Python for communicating with a database should be the same, regardless of the database and the database module used. Even though we use lots of SQL examples, this is not an introduction into SQL but a tutorial on the Python interface. SQLite is a simple relational database system, which saves its data in regular data files or even in the internal memory of the computer, i.e, the RAM. It was developed for embedded applications, like Mozilla Firefox (Bookmarks), Symbian OS or Android. SQLITE is "quite" fast, even though it uses a simple file. It can be used for large databases as well. If you want to use SQLite, you have to import the module `sqlite3`. To use a database, you have to create first a Connection object. The connection object will represent the database. The argument of connection -in the following example "company.db" - functions both as the name of the file, where the data will be stored, and as the name of the database. If a file with this name exists, it will be opened. It has to be a SQLite database file of course! In the following example, we will open a database called company. MySQL Connector/Python enables Python programs to access MySQL databases, using an API that is compliant with the Python Database API Specification v2.0 (PEP249) It is written in pure Python and does not have any

dependencies except for the Python Standard Library. For notes detailing the changes in each release of Connector/Python, see MySQL Connector/Python Release Notes. MySQL Connector/Python includes support for:

1. Almost all features provided by MySQL Server up to and including MySQL Server version 5.7.
2. Converting parameter values back and forth between Python and MySQL data types, for example Python date time and MySQL DATETIME. You can turn automatic conversion on for convenience, or off for optimal performance.
3. All MySQL extensions to standard SQL syntax.
4. Protocol compression, which enables compressing the data stream between the client and server.
5. Connections using TCP/IP sockets and on Unix using Unix sockets
6. Secure TCP/IP connections using SSL.
7. Self-contained driver. Connector/Python does not require the MySQL client library or any Python modules outside the standard library.

Download Process

DOWNLOADING

1. Click Python Download.
2. Click the Windows link (two lines below the Download Python 3.7.0
3. Click on the Download Windows x86-64 executable installer link under the top-left Stable Releases.

- ☐ pop-up window titled Opening python 3.7.0-amd 64.exe will appear.
- ☐ Click the Save File button.
- ☐ The file named python
- ☐ 3.7.0-amd64.exe should start downloading standard into your download folder.

4. Move this file to a more permanent location, so that you can install Python (and reinstall it easily later, if necessary).

5. Feel free to explore this web page further; if you want to just continue the installation, you can terminate the tab browsing web page.

INSTALLING

1. Double-click the icon labelling the file python-3.7.0-amd64.exe.

- A Python 3.7.0 (64-bit) Setup pop-up window will appear
- Ensure that the Install launcher for all users (recommended) and the Add Python 3.7.0 to PATH check boxes at the bottom are checked.

2. Highlight the Install Now (or Upgrade Now) message, and then click it.

3. Click the Yes button.

- ☐ A new Python 3.7.0 (64-bit) Setup pop-up window will appear with a Setup Progress message and a progress bar.
- ☐ During installation, it will show the various components it is installing and move the progress bar towards completion. Soon, a new Python 3.7.0 (64-bit) Setup pop-up window will appear with a Setup was successfully message.

4. Click the Close button.

ALGORITHMS

1.FERNET

Cryptography is the practice of securing useful information while transmitting from one computer to another or storing data on a computer. Cryptography deals with the encryption of plaintext into ciphertext and decryption of ciphertext into plaintext. Python supports a cryptography package that helps us encrypt and decrypt data. The fernet module of the cryptography package has inbuilt functions for the generation of the key, encryption of plaintext into ciphertext, and decryption of ciphertext into plaintext using the encrypt and decrypt methods respectively. The fernet module guarantees that data encrypted using it cannot be further manipulated or read without the key.

Methods Used:

- **generate_key():** This method generates a new fernet key. The key must be kept safe as it is the most important component to decrypt the ciphertext. If the key is lost then the user can no longer decrypt the message. Also if an intruder or hacker gets access to the they can not only read the data but also forge the data.
- **encrypt(data):** It encrypts data passed as a parameter to the method. The outcome of this encryption is known as a “Fernet token” which is basically the ciphertext. The encrypted token also contains the current timestamp when it was generated in plaintext. The encrypt method throws an exception if the data is not in bytes.
- **decrypt(token, ttl=None):** This method decrypts the Fernet token passed as a parameter to the method. On successful decryption the original plaintext is obtained as a result, otherwise an exception is thrown.

2 . RSA :-

RSA algorithm is an asymmetric cryptography algorithm. Asymmetric actually means that it works on two different keys i.e, **Public Key** and **Private Key**. As the name describes that the Public Key is given to everyone and the Private key is kept private.

An example of asymmetric cryptography:

1. A client (for example browser) sends its public key to the server and requests some data.
2. The server encrypts the data using the client's public key and sends the encrypted data.
3. The client receives this data and decrypts it.

Since this is asymmetric, nobody else except the browser can decrypt the data even if a third party has the public key of the browser.

The idea! The idea of RSA is based on the fact that it is difficult to factorize a large integer. The public key consists of two numbers where one number is a multiplication of two large prime numbers. And private key is also derived from the same two prime numbers. So if somebody can factorize the large number, the private key is compromised. Therefore encryption strength totally lies on the key size and if we double or triple the key size, the strength of encryption increases exponentially. RSA keys can be typically 1024 or 2048 bits long, but experts believe that 1024-bit keys could be broken in the near future. But till now it seems to be an infeasible task.

SDLC

- ☐ What is SDLC?

SDLC stands for Software Development Life Cycle. A Software Development Life Cycle is essentially a series of steps, or phases, that provide a model for the development and lifecycle management of an application or piece of software. SDLC is the process consisting of a series of planned activities to develop or alter the software products.

- ☐ Benefits of the SDLC Process

The intent of a SDLC process is to help produce a product that is cost-efficient, effective, and of high quality. Once an application is created, the SDLC maps the proper deployment and decommissioning of the software once it becomes a legacy. The SDLC methodology usually contains the following stages: Analysis (requirements and design), construction, testing, release, and maintenance (response). Veracode makes it possible to integrate automated security testing into the SDLC process through use of its cloud based platform.

1. Requirements Gathering:

In this phase we gather all the requirements from the client, i.e, what are the client expected input, output.....

2. Analysis:

In this phase based upon the client requirements we prepare one documentation is called "High Level Design Document". It contains Abstract, Functional Requirements, Non Functional Requirements, Existing System, Proposed System, SRS.

3. Design:

It is difficult to understand the High-Level Design Document for all the members, so to understand easily we use "Low Level Design Document". To design this document, we use UML (Unified Modelling Language). In this we have Use case, Sequence, Collaboration.....

4. Coding:

In this phase we develop the coding module by module. After developing all the modules, we integrate them.

5. Testing:

After developing we have to check whether client requirements are satisfied or not. If not, we are again going to develop.

6. Implementation:

In testing phase if client requirements are satisfied, we go for implementation. i.e., we need to deploy the application in some server.

7. Maintenance:

After deployment, if at all any problems come from the client side. We are providing maintenance for that application.

FEASIBILITY STUDY

Preliminary investigation examine project feasibility, the likelihood the system will

be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Operational Feasibility
- Economical Feasibility

ECONOMIC FEASIBILITY

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, there is nominal expenditure and economic feasibility for certain

OPERATIONAL FEASIBILITY

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following:

Is there sufficient support for the management from the users?

- ☐ Will the system be used and work properly if it is being developed and implemented?
- ☐ Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into

consideration. So, there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

TECHNICAL FEASIBILITY

☐ The technical issue usually raised during the feasibility stage of the investigation includes the following:

- ☐ Does the necessary technology exist to do what is suggested?
- ☐ Do the proposed equipment have the technical capacity to hold the data required to use the new system?
- ☐ Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- ☐ Can the system be upgraded if developed?

Earlier no system existed to cater to the needs of 'Secure Infrastructure Implementation System'. The current system developed is technically feasible. It is a web-based user interface for audit workflow at NIC-CSD. Thus, it provides an easy access to the users.

The database's purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hardware requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing fast feedback to the users irrespective of the number of users using the system

TESTING

Testing is a process of executing a program with the aim of finding error. To make our software perform well it should be error free. If testing is done successfully, it will remove all the errors from the software.

Types of Testing

- White Box Testing
- Black Box Testing
- Unit Testing
- Integration Testing
- Alpha Testing
- Beta Testing
- Performance Testing and so on

➤ White Box Testing

Testing technique based on Knowledge of the internal logic of an application's code and includes tests like coverage of code statements, branches, paths, conditions. It is performed by software developers.

➤ Black Box Testing

A method of software testing that verifies the functionality of an application without having specific knowledge of the application's code/internal structure. Tests are based on requirements and functionality.

➤ Unit Testing

Software verification and validation method in which a programmer tests if

individual units of source code are fit for use.

➤ **Integration Testing**

The phase in software testing in which individual software modules are combined and tested as a group. It is usually conducted by testing teams.

➤ **Alpha Testing**

Type of testing a software product or system conducted at the developer's site. Usually, it is performed by the end users.

➤ **Beta Testing**

Functional Testing conducted to evaluate the compliance of a system or component with specified performance requirements. It is usually conducted by the performance engineer.

➤ **Black Box Testing**

Black box testing is testing the functionality of an application without knowing the details of its implementation including internal program structure, data structure etc. Test cases for black box testing are created based on the requirement specifications. Therefore, it is also called as specification-based testing.

When applied to machine learning models, black box testing would mean testing machine learning models without knowing the internal details such as features of the machine learning.

Model1, the algorithm used to create the model etc. The challenge, however, is to verify the test outcome against the expected values that are known beforehand.

CHAPTER – 6

RESULTS AND DISCUSSION

- File encryption is accomplished by the application of complicated algorithms. The information contained in a file is considered to be encrypted if it has been scrambled using an encryption algorithm.
- A scrambled file is unreadable, but there are still many cyber-attacks taking place in a variety of methods, threatening the secrecy of the information.
- In this paper, we looked into images as the key to file encryption and looked at a number of case examples. The experimental findings show that the proposed image as a key for encryption and decryption operations is secure and dependable, paying the way for its usage in high-security image communication applications.

CHAPTER- 7

CONCLUSION

Cloud computing is significant and successful assistance to give different assets to the clients particularly for putting away information. Delicate information is inclined to be more defenseless. Delicate information genuinely must ought to be put away secretly. Subsequently, security of data is the most major and critical point. The security can be achieved by applying cryptographic estimations. Here we have made & executed SDV system model that unexpected spikes popular for a multi-server SSE plot and showed that encoding records into blocks and moving to a couple of Distributed storage providers includes alright cost above diverged from moving special reports decoded. This proposed system is easy to use and there are many benefits utilizing these methodologies yet it's vital to figure out what level of encryption is expected to play out the errand well in a particular setting. Different encryption strategies are seen in this paper, and this shows that a ton of work has been finished for proposing the Cloud storage encryption components.

FUTURE WORK

While much effort has been directed at securing network communications, security of stored data remains a largely neglected area both in the development and use of such systems. Nonetheless, various implementations of encrypting file systems exist. As observed the current encryption software doing the encryption task is all very complicated in its functionality. The method of encryption and key generation of the current system for a new user to understand is complex in nature. The existing systems fail to provide a higher level of security and make private documents more prone to theft and attacks.

The future enhancement of the project is to store the file with more secure ways and user can easily understand the process of encrypting the file. We plan to store the file with more secure ways

CHAPTER-9

REFERENCES

- [1] I Made Ari Dwi Suta Atmaja, I Nyoman Gede Arya Astawa, Ni Wayan Wisswani, I Made Riyan Adi Nugroho, "Document Encryption Through Asymmetric RSA Cryptography", International Conference on Applied Science and Technology (iCAST), 2021
- [2] Yuxiang Lin, Xin Xia, Jingyi Yang, "Document Encryption Method with Mechanism of Enigma Machine", International Conference on Artificial Intelligence, Big Data and Algorithms (CAIBDA), 2021
- [3] S. D. Sanap, Vijayshree More, "Analysis of Encryption Techniques for Secure Communication", International Conference on Emerging Smart Computing and Informatics (ESCI), 2021
- [4] Alpana A. Ingale, Sunil K. Moon, "E-Government Documents Authentication and Security by Utilizing Video Crypto-Steganography", IEEE Global Conference on Wireless Computing and Networking (GCWCN), 2019
- [5] Bryan H. Wodi, Carson K. Leung, Alfredo Cuzzocrea, S. Sourav, "Fast Privacy-Preserving Keyword Search on Encrypted Outsourced Data", IEEE International Conference on Big Data (Big Data), 2020
- [6] Aruna Guruvaya Mogarala, K. G. Mohan, "Security and Privacy Designs Based Data Encryption in Cloud Storage and Challenges: A Review", 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2018
- [7] Daniel Mahardika Yusuf, De Rosal Ignatius Moses Setiadi, Eko Hari Rachmawanto, Christy Atika Sari, Rabei Raad Ali, "Dual Encryption Method for File Security", 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), 2020

- [8] Anil Kumar, Priyanka Dahiya, Garima, Sarvesh Tanwar, "A New Hybrid Approach for Encrypting XML Documents", 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2021
- [9] Prachi More, Shubham Chandugade, Shaikh Mohammad Shafi Rafiq, Priya Pise, "Hybrid Encryption Techniques for Secure Sharing of a Sensitive Data for Banking Systems Over Cloud", International Conference On Advances in Communication and Computing Technology (ICACCT), 2018
- [10] Jie Tong, Yihong Long, Quan Liu, "A File Encryption System Based on Attribute Based Encryption", 17th International Conference on Computational Intelligence and Security (CIS), 2022

APPENDIX

A. SOURCE CODE

User register

```
{% extends "main.html" %}
{% block title %}Register User{% endblock title %}
{% block content %}
{% load crispy_forms_tags %}
<div class="container">
<form method='POST', class="form-group">
    {% csrf_token %}
    {{ form.as_p }}
    <button type='submit' class="btn btn-primary form-control">Submit</button>
</form>
</div>
{% endblock content %}
```

Decrypt_file

```
{% extends "main.html" %}
{% block title %}Decrypt your file{% endblock title %}
{% block content %}
<div class="container">
    <form method='POST' enctype='multipart/form-data' class="form-group">
        {% csrf_token %}
        <div class="input-group mb-3">
            <label class="input-group-text" for='decryptlayer1'>Enter key for 1st layer
            Decryption(2nd key you were provided)</label>
            <input class="form-control" type='FILE' placeholder='enter key here.'
            name='key1' /><br/>
        </div>
        <div class="input-group mb-3">
            <label class="input-group-text" for='decryptlayer2'>Enter key for 2nd layer
            Decryption (1st key you were provided)</label><br/>
```

```

        <input class="form-control" type='FILE' placeholder='enter key here'
name='keyl2'/><br/>
    </div>
    <button class="btn btn-primary form-control" type='submit'>Upload</button>
</form>
</div>
{% if file_path %}
    <a class='btn btn-secondary form-control' href="{{file_path}}">Download decrypted
file here</a>
{% endif %}

{% endblock content %}

```

Decrypt_file_three

```

{% extends "main.html" %}
{% block title %}Decrypt your file{% endblock title %}
{% block content %}
    <div class="container">
        <form method='POST' enctype='multipart/form-data' class="form-group">
            {% csrf_token %}
            <div class="input-group mb-3">
                <label class="input-group-text" for='decryptlayer1'>Enter key for 1st layer
Decryption(3rd key you were provided)</label>
                <input class="form-control" type='FILE' placeholder='enter key here.'
name='keyl1'/><br/>
            </div>
            <div class="input-group mb-3">
                <label class="input-group-text" for='decryptlayer2'>Enter key for 2nd layer
Decryption (2nd key you were provided)</label><br/>
                <input class="form-control" type='FILE' placeholder='enter key here'
name='keyl2'/><br/>
            </div>
            <div class="input-group mb-3">

```

```

        <label class="input-group-text" for='decryptlayer2'>Enter key for 3rd layer
Decryption (1st key you were provided)</label><br/>
        <input class="form-control" type='FILE' placeholder='enter key here'
name='keyl3'/><br/>
    </div>
    <button class="btn btn-primary form-control" type='submit'>Upload</button>
</form>
</div>
{% if file_path %}
    <a class='btn btn-secondary form-control' href="{{file_path}}">Download decrypted
file here</a>
{% endif %}

{% endblock content %}

```

Encryption page

```

{% extends "main.html" %}
{% block title %}Encrypt your File{% endblock title %}
{% block content %}
<div class="container">
    <form class="form-group" method='POST' enctype='multipart/form-data'>
        {% csrf_token %}
        <div class="input-group mb-3">
            <label class="input-group-text" for=filename>Enter Your Filename with
extension</label>
            <input class="form-control" type='text' name='filename'/>
        </div>
        <div class="input-group mb-3">
            <label class="input-group-text" for='file'>Upload your file here: </label>
            <input class="form-control" type='file' name='file'/>
        </div>
        <br/>
        <button class="btn btn-primary form-control" type='submit'>Upload</button>
    </form>
</div>

```

```

    </form>
</div>
{% if key1 %}
    <div class="container">
        <h4>These keys are very important 2nd key will be required for 1st layer
        decryption and 1st key will be required for the 2nd layer decryption</h4>
        <a class="btn btn-secondary" href='/media/key/filekey.key' download>Download
        key 1</a>
        <a class="btn btn-secondary" href='/media/key/{{filename}}_RSA_private.pem'
        download>Download key 2</a>
        <br/>
    </div>
{% endif %}
{% endblock content %}

```

Encryption page three

```

{% extends "main.html" %}
{% block title %}Encrypt your File{% endblock title %}
{% block content %}
<div class="container">
    <form class="form-group" method='POST' enctype='multipart/form-data'>
        {% csrf_token %}
        <div class="input-group mb-3">
            <label class="input-group-text" for=filename>Enter Your Filename with
            extension</label>
            <input class="form-control" type='text' name='filename' />
        </div>
        <div class="input-group mb-3">
            <label class="input-group-text" for='file'>Upload your file here: </label>
            <input class="form-control" type='file' name='file' />
        </div>
        <br/>
        <button class="btn btn-primary form-control" type='submit'>Upload</button>
    </form>
</div>

```

```

    </form>
</div>
{% if key1 %}
    <div class="container">
        <h4>These keys are very important 3rd key will be required for 1st layer
        decryption and 2nd key will be required for the 2nd layer decryption and 1st key will
        be required for the 3rd layer</h4>
        <a class="btn btn-secondary" href='/media/key/filekey.key' download>Download
        key 1</a>
        <a class="btn btn-secondary" href='/media/key/filekeyl2.key'
        download>Download key 2</a>
        <a class="btn btn-secondary" href='/media/key/{{filename}}_RSA_private.pem'
        download>Download key 3</a>
    <br/>
    </div>
{% endif %}
{% endblock content %}

```

Home

```

{% extends 'main.html'%}
{% block title %} Home Page {% endblock title %}
{% block content %}
<div class="container">
    <div class="row">
        <div class="col-12">
            <div class="container">
                <h4 >Why encrypting files?</h4>
                <p>We need to encrypt our files so that we can store our files in cloud in
                safe way from the hacker. So if at any point of time if the database is hacked the files
                we stored are still safe from the hackers</p>
                <br/><br/>
                <h4>Why using hybrid process?</h4>
            </div>
        </div>
    </div>
</div>

```

<p>We use some hybrid algorithm to encrypt the file through multiple layers and make sure of proper safety of your files</p>

<h4> What is the Difference between 2 layer and 3 layer algorithm difference? </h4>

<p> what we are trying to do is passing a file through different cipher and encrypting the file in multiple layers, now why we are using different keys, just to make them more secure, because due to 1 data breach if our database is compromised and somehow key is compromised we will loose all our data.</p>

<h4>Performance analysis</h4>

<p> while using 2 layers the file getting decrypted when its get compromised is almost 20% but while using 3 layers the possibility comes down to only almost 9%
 Ref: Internet and IEEE research paper</p>

<h4>Size Compression analysis</h4>

<p> it is observed that in each layer the size of the file gets 5-10% lesser by getting compressed.

For example: initially file size is 100kb then it after 1st layer it will be 90kb at max and after 2nd layer it will be 81kb at max. and after 3rd layer will be 73kb at max.

</p>

</div>

</div>

</div>

</div>

{% endblock content %}

List file

{% extends "main.html" %}


```
{% block title %}Stored File list{% endblock title %}
{% block content %}
    {% if all_files %}
        <div class="container">
            {% for file in all_files %}
                <a class="btn btn-primary form-control"href="{% url 'decrypt' file.filename %}">
                    {{file.filename}} </a>
                <br/>
            {% endfor %}
        </div>
    {% endif %}
{% endblock content %}
```

List files3

```
{% extends "main.html" %}
{% block title %}Stored File list{% endblock title %}
{% block content %}
    {% if all_files %}
        <div class="container">
            {% for file in all_files %}
                <a class="btn btn-primary form-control"href="{% url 'decryptl3' file.filename
%}"> {{file.filename}} </a>
                <br/>
            {% endfor %}
        </div>
    {% endif %}
{% endblock content %}
```

Main

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
```

```

<meta http-equiv="X-UA-Compatible" content="IE=edge">
<title>{% block title %}{% endblock %}</title>
<meta name="description" content="">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
OERcA2EqJJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>
</head>
<body>
  {% include 'navbar.html' %}
  <div class="bg-image p-5 text-center shadow-1-strong rounded mb-5 text-white
vh-100" style="background-image: url('https://images.unsplash.com/photo-
1526374965328-7f61d4dc18c5?ixlib=rb-
4.0.3&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&
fit=crop&w=1170&q=80');">
    {% block content %}{% endblock %}
  </div>
</body>
</html>

```

Navbar

```

<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="{% url 'home' %}">f-cryptor</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">

```

```
<span class="navbar-toggler-icon"></span>
```

```
</button>
```

```
<div class="collapse navbar-collapse" id="navbarSupportedContent">
```

```
<ul class="navbar-nav mr-auto">
```

```
{% if request.user.is_authenticated %}
```

```
<li class="nav-item">
```

```
<a class="nav-link" href="{% url 'encrypt' %}">Encrypt file</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-link" href="{% url 'encryptl3' %}">Encrcrypt3.0</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-link" href="{% url 'fileList' %}">File List</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-link" href="{% url 'list3' %}">FileList3</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-link" href="{% url 'showpublickey' %}">Get your public key</a>
```

```
</li>
```

```
<li class="nav-link">
```

```
<a class="nav-link" href="{% url 'userLogout' %}">Logout</a>
```

```
</li>
```

```
{% else %}
```

```
<li class="nav-item">
```

```
<a class="nav-link" href="{% url 'login' %}">Login</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-link" href="{% url 'register' %}">Register</a>
```

```
</li>
```

```
{% endif %}
```

```

        {% if request.user.is_authenticated %}
        <li class="nav-item"><span class="nav-link mx-xxl-5">{{request.user}}</span></li>
        {% endif %}
    </ul>
</div>
</nav>

```

User login

```

{% extends "main.html" %}
{% block title %}User Login{% endblock title %}
{% block content %}
<div class="container" >
    <form method='POST' class='form-group'>
        {% csrf_token %}
        <label for='username'>Username:</label>
        <input name='username' class="form-control" type='text' placeholder='Enter
username here' />
        <label for='password'>Password</label>
        <input name='password' type='password' class='form-control' placeholder='Enter
password here' />
        <button type='submit' class="btn btn-primary form-control">Submit</button>
    </form>
</div>
{% endblock content %}

```

B. SCREENSHOTS

f-cryptor

[Encrypt file](#) [Encrypt3.0](#) [File List](#) [FileList3](#) [Get your public key](#) [Logout](#) 39110648

Enter Your Filename with extension

Upload your file here:

Choose File

No file chosen

Upload

Encryption page

f-cryptor

[Encrypt file](#) [Encrypt3.0](#) [File List](#) [FileList3](#) [Get your public key](#) [Logout](#) 39110648

Why encrypting files?

We need to encrypt our files so that we can store our files in cloud in safe way from the hacker. So if at any point of time if the database is hacked the files we stored are still safe from the hackers

Why using hybrid process?

We use some hybrid algorithm to encrypt the file through multiple layers and make sure of proper safety of your files

What is the Difference between 2 layer and 3 layer algorithm difference?

what we are trying to do is passing a file through different cipher and encrypting the file in multiple layers, now why we are using different keys, just to make them more secure, because due to 1 data breach if our database is compromised and somehow key is compromised we will loose all our data.

Performance analysis

while using 2 layers the file getting decrypted when its get compromised is almost 20% but while using 3 layers the possibility comes down to only almost 9% Ref. Internet and IEEE research paper

Size Compression analysis

it is observed that in each layer the size of the file gets 5-10% lesser by getting compressed. For example: initially file size is 100kb then it after 1st layer it will be 90kb at max and

Interface page

f-cryptor
Encrypt file
Encrypt3.0
File List
FileList3
Get your public key
Logout
39110648

These keys are very important 2nd key will be required for 1st layer decryption and 1st key will be required for the 2nd layer decryption

python.pdf.bin_R...pem
filekey (7).key

Show all

Decryption page

f-cryptor
Encrypt file
Encrypt3.0
File List
FileList3
Get your public key
Logout
39110648

machine learning.p...bin

Show all

File download page

f-cryptor Encrypt file Encrypt3.0 File List FileList3 Get your public key Logout 39110648

Enter key for 1st layer Decryption(3rd key you were provided)

Choose File

DBMS.pdf_RSA_private.pem

Enter key for 2nd layer Decryption (2nd key you were provided)

Choose File

filekey12 (4).key

Enter key for 3rd layer Decryption (1st key you were provided)

Choose File

filekey (12).key

Upload

Encryption3 page

f-cryptor Encrypt file Encrypt3.0 File List FileList3 Get your public key Logout 39110648

Enter key for 1st layer Decryption(3rd key you were provided)

Choose File

No file chosen

Enter key for 2nd layer Decryption (2nd key you were provided)

Choose File

No file chosen

Enter key for 3rd layer Decryption (1st key you were provided)

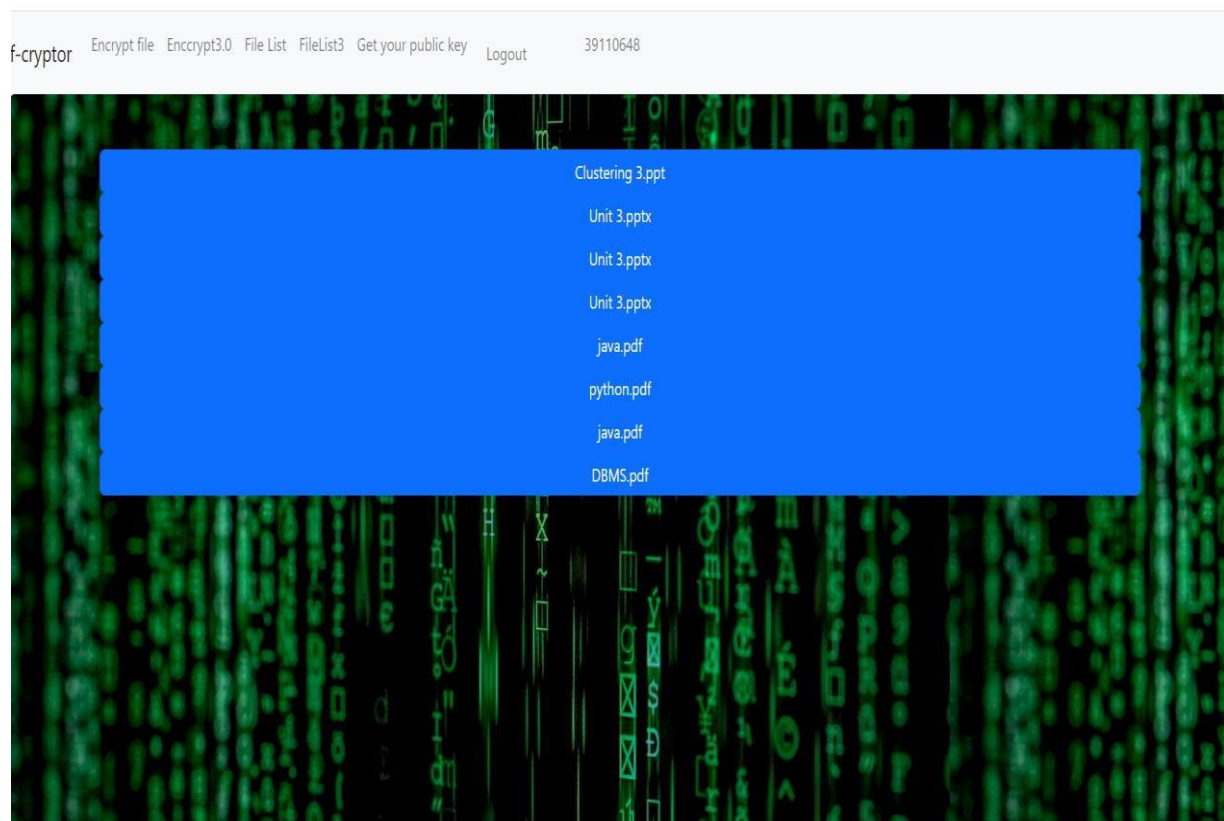
Choose File

No file chosen

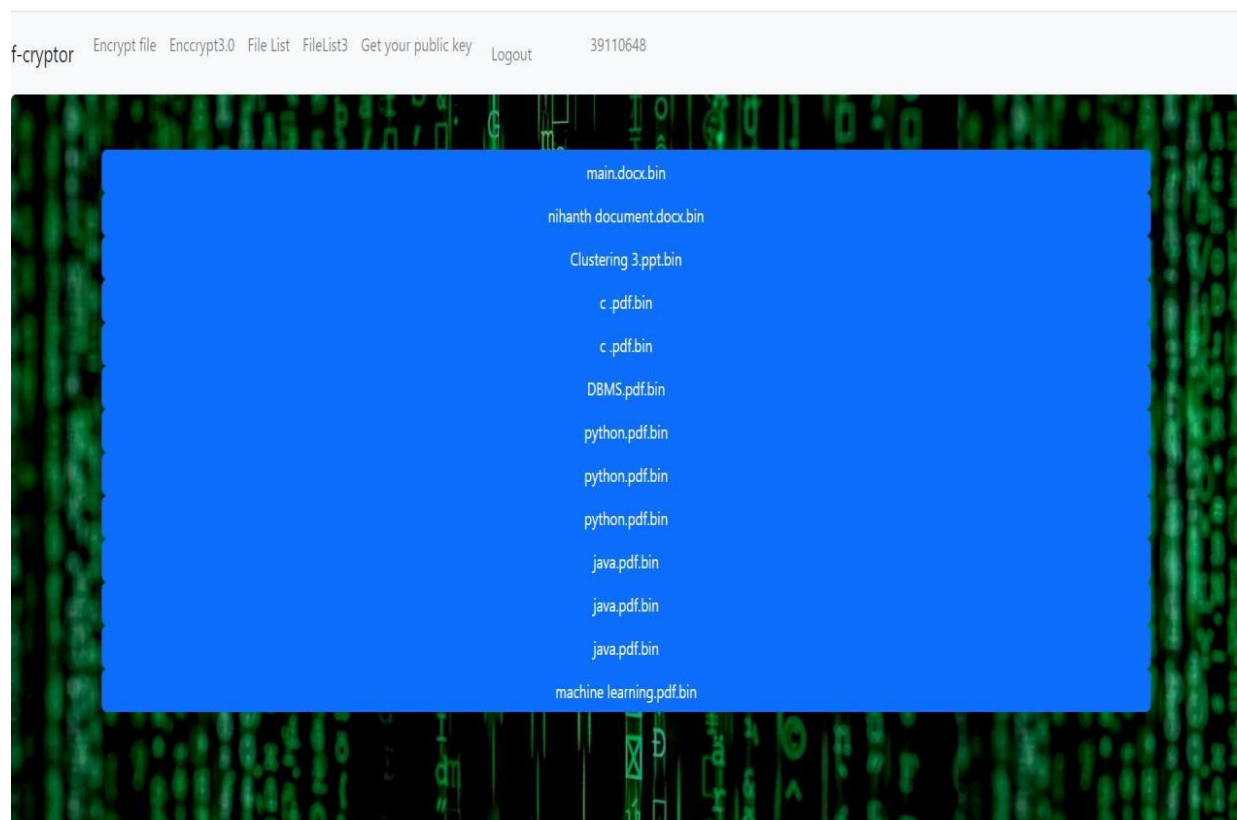
Upload

Download decrypted file here

File download page



File list3 page



File list page

PRIVATE DOCUMENTS VAULT

ASHOK KUMAR M^[1], SRI SATYA NIHANTH M^[2],

CHRISTY A^[3]

[1][2] UG Student, Dept. of CSE, Sathyabama Institute of Science and Technology, Chennai, India

[3] Associate professor, Dept. of CSE, Sathyabama Institute of Science and Technology, Chennai, India

mukkapatiasbok2001@gmail.com, nihanthsatya@gmail.com, christy.cse@sathyabama.ac.in

ABSTRACT: - *With the development of PC frameworks, how much delicate information to be put away as well as the quantity of dangers on this information grows up, making information secrecy progressively critical to PC clients. At present, with gadgets generally associated with the Web, the utilization of cloud information capacity administrations has become functional and normal, permitting fast admittance to such information any place the client is. Such common sense carries with it a worry, unequivocally the secrecy of the information which is conveyed to outsiders for capacity. Records may likewise be spilled by inquisitive chairmen. A straightforward arrangement is for the client to encode all reports prior to submitting them. This technique, in any case, makes it difficult to proficiently look for archives as they are completely encrypted. We propose a confidential record vault with server-side encryption, which likewise empowers clients to handily convey encryption and other security arrangements by offering hearty, focal administration of encryption keys moreover, this paper plans and carries out a model framework. As a result of confirmation, examination, convenience & security, the situation is demonstrated the arrangement be able to meet the information security assurance prerequisites of delicate E-records in the open organization climate. We use streamlit, a famous open-source structure for making AI and perception applications in Python. Our fundamental exhibition assessment shows that this element presents satisfactory calculation overheads when contrasted with submitting records straightforwardly to a Cloud storage administration.*

Keywords: Cloud computing, Documents, SSE, Security.

I INTRODUCTION

In Cloud storage and registering climate information protection and its security are the main pressing issue. To conquer this worry, we intertwine cryptography ideas into Cloud computing. Cryptography help in information encryption and decoding strategy is use to safeguard the information

in cloud. To guarantee security, information encryption is finished by the client. The client shares the document through cloud yet individual who realizes the key just can decodes the record. The interloper get the document yet they can't decryption. In advancement process, gatecrasher attempt to break the key utilizing look into table method, savage power strategy and so on.

Cloud storage administrations [1] are famous on account of the many advantages that are well known and empower clients to see their information from anyplace and whenever and get the data they need. In any case, clients actually don't have the trust in that frame of mind to rethink their delicate information. Cloud storage ought to give functional and solid answers for guarantee the security of these information and keep up with the capacity to look for clients. Information sent against penetration, by unlawful organizations is extremely essential, particularly in basic foundations. To safeguard delicate information, a straightforward way is to encode information prior to revaluating, however when the client expects to look through a catchphrase among the information, it will be challenging to look. During the time, two arrangements were utilized; the two of which today are dismissed. To start with, the client gets every one of the information, decryptions them, does the pursuit, then, at that point, re-scrambles the information, and recovers the cloud server. It is viewed as in inconsistency with the utilization of cloud administrations. The another method is to give encryption key to cloud server but it didn't work as brilliant because the server is not reliable, however their plan needs a great deal of association between the server and the client and has low execution, plot depends on symmetric encryption. Occasions of organizations consolidate OneDrive, G-drive, I-cloud, Dropbox lastly AwsS3. Outfits an individual with straightforward permission to user data wherever and at whatever point. These organizations are really confined from the ordinary cloud organizations introduced by the two providers and have with agree to serious government security rules.

Our fundamental obligations are as per the going

with, SDV licenses records, or bits of an encoded report, called blocks, to be dealt with in various Circulated stockpiling associations, rather than existing plans and frameworks that highlight on moving to a particular putting away. It guarantees with high likelihood that no single putting away supplier has a total arrangement of blocks, which it could use to become familiar with extra data about the report. Clearly, the focal open encryption framework gives such a section. A center piece of SDV is a regulator that manages demand records, report comfort, and recovery. The regulator is organized with the objective that the fundamental open SSE plot, used for valuable pursuit, is pluggable. It derives existing plans that cook for single cutoff supplier can be adjusted to use with our construction. The regulator might be executed and set in a request entrance. Here likewise further proposing a multi-server SSE plot with an execution for SDV. The solicitation record is supposed to take extraordinary thought of two-level word reference structure, where the significant level is for a solitary word question.

II REALTED WORK

Mell et al [1] characterized of giving admittance to shared assets. They characterized five attributes of cloud as on-request self-administration, expansive access organization, asset pooling, quick flexibility and estimated administration. They recognized that a cloud model depends on the three help models as Programming as-a-Administration (SaaS), Stage as-a-Administration (PaaS) and framework as-a-Administration (IaaS). They additionally suggested that cloud fundamentally deals with any of the four arrangement models-private cloud, public cloud, mixture cloud or local area cloud.

Atallah et al[2] Key task conspire expect to limit the cost in putting away and overseeing secret keys for general cryptographic use. Key task plots in all probability non-steady decoding key size, symmetric or public key for a predefined progressive system is utilized. Just hash capabilities are utilized for a hub to get a relative's key from its own key. The space intricacy of the public data is equivalent to that of putting away order and is asymptotically ideal; the confidential data at a hub comprises of a solitary key related with that hub and updates are taken care of locally in the progressive system [2]. Introduced an encryption plot which is initially proposed for compactly communicating enormous number of keys in broadcast situation. What's more, utilizes Symmetric-key encryption with Conservative Key. In this paper construct an effective framework that permits patients both to share halfway access freedoms with others, and to perform look through over their records. They formalize the necessities of a Patient Controlled Encryption plan, and give a few cases, in light of existing cryptographic natives and conventions, each accomplishing an alternate arrangement of

properties.

Chow et al[4] In this plan, key accumulation is obliged as in all keys to be amassed should come from various "personality divisions". While there are a dramatic number of characters and hence secret keys, just a polynomial number of them can be collected. This extraordinarily expands the expenses of putting away and sending ciphertexts, which is unfeasible as a rule like shared Cloud storage.

Chang et al[5] in like manner proposed an open encryption plot using record form and described proliferation built security model. This huge number of plans rely upon a lone server. Even more lately, plans that incorporate more than one server have been proposed, for instance, in

Bösch et al. [6] an inquiry middle person is familiar besides with the limit server. The inquiry middle person works with the goal that every request isn't exactly equivalent to the past requests, regardless, when a watchword is addressed usually. The chief idea is for the inquiry delegate to re-encode the request but again process the document each time a request is submitted. Along these lines, the arrangement may not be adaptable. *Ishai et al. [7]* he has presented about Helper Server. HS objective is to restrict spillage for instance and range questions. Its like scale to immense informational indexes. By and large less computationally capable ignorant PIR and multi-party estimation frameworks are shipped off a little piece of the data base.

III EXISTING SYSTEM

While great effort has gone into protecting network communications, the security of stored data remains a largely overlooked topic in both the construction and usage of such systems. However, multiple encrypting file system implementations exist. As can be seen, the present encryption software performing the encryption process is quite complex in its functioning. The existing system's encryption and key generation mechanism is difficult for a new user to grasp. Existing technologies fall short of providing a greater level of protection, making sensitive papers more vulnerable to theft and assault.

IV PROPOSED SYSTEM

4.1 SELECTED METHODOLOGY OR PROCESS MODEL

The most ridiculously complete and broadly involved cloud stage on the planet, Amazon Web Service, gives more than 200 completely utilitarian administrations from server farms across the world. A great many clients use AWS to save costs, increment deftness, and speed up development, including the biggest companies, best legislative associations, and the quickest developing new businesses. Contrasted with other cloud suppliers, AWS offers an essentially more prominent number of administrations and elements inside those administrations, going from framework innovations

like figure, stockpiling, and information bases to state of the advancements like man-made consciousness speedier more practical, develop.

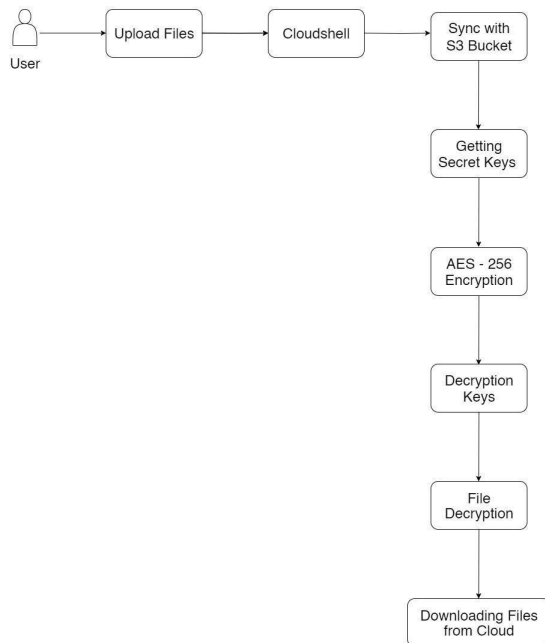


FIG 1 OVERVIEW OF THE PROPOSED SYSTEM

AWS supplies the most comprehensive usefulness among such administrations. For example, Amazon provides the finest selection of data sets that are specifically designed for various applications, allowing you to select the right device for the task at the optimum cost and execution. Amazon is the most adaptable and secure distributed computing platform currently available, and it is designed to be both. Our core structure meets the security requirements of the military, major banks, and other extremely sensitive organizations. With 230 security, consistence, and administration administrations and capacities, this is upheld by an exhaustive assortment of cloud security devices. As well as offering the choice to encryption client information across every one of the 117 AWS administrations that have it, AWS upholds 98 security guidelines and consistence affirmations.

4.2 IMPLEMENTING MACHINE LEARNING

Scrambling information very still and on the way (both organized and non-organized) is as of now normal practice inside organizations, to safeguard classified, mystery and exclusive data. Encoding information while being used, in any case, is a more uncommon practice. Information being used is information that is put away in a non-tenacious computerized state as well as that is being handled, as in the lifecycle of an AI (ML) model.

These days enormous cloud administrators, like Google, AWS, and Microsoft, and new businesses the same are offering AI as a Help (MLaaS). These

administrations help little organizations that miss the mark on ML skill or expected framework to construct a prescient model with the information from the little organizations. This information is significant and delicate. The clients get their administrations through Programming interface calls. Specialist organizations would rather not get serious about their model, which are secret elements to the clients. Furthermore, due to the information responsiveness, clients may not be intrigued to share their crude information through Programming interface calls. Here comes the trust commendable "Encrypted AI" idea that safeguards the information and the model by encoding it. Rather than only giving MLaaS that may be flawed, specialist organizations can present EMLaaS (Encrypted AI as a Help) to guarantee clients about their information security. EMLaaS can assist little organizations with guaranteeing information security and make astounding simulated intelligence driven arrangements with insignificant expense. In a time when accumulated information has the ability to move exploration and development in many fields, encoded and confidential simulated intelligence could assist with safeguarding information protection, fabricate trust, and empower various gatherings to team up.

4.3 SETTING UP AWS

The principal page for Amazon Web Administrations (AWS). Then, at that point, need to choose AWS Record Creation, Pick Sign in to the Control center assuming you as of late signed in to AWS. Pick Sign in to an alternate record first, then select Make another AWS account if make another AWS account isn't shown. Enter your email address in Root client email address, roll out any fundamental improvements to the AWS account name, and afterward select Check email address. This address will receive an email from AWS with a check code.

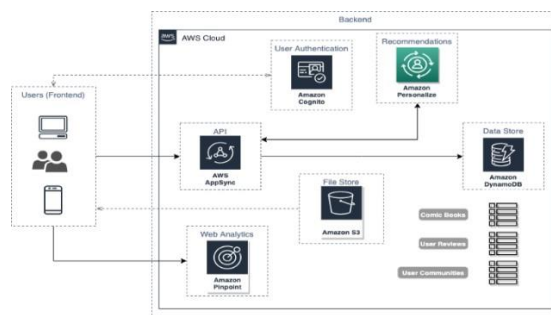
4.4 SETUP BILLING

Enter your payment method details on the billing information page, then click Verify and Add. You must enter your CVV as part of the verification process when creating an Amazon Internet Services Private Limited (AISPL) account in India. Depending on your bank, you might also need to enter a one-time password. As part of the verification process, AISPL deducts two Indian Rupees (INR) from your payment method. Following completion of the verification, AISPL returns the two INR. Select Utilize a new address if you want to use a different billing address for your AWS billing information. Select Verify and Continue after that

4.5 SETUP S3 BUCKET IN YOUR NEAREST REGION

Across various AWS Region, clients in Amazon Virtual Confidential Mists (VPCs) should peruse and compose information. Solicitations to S3 are consequently sent to the AWS Locale with the most reduced inactivity when you interface with your S3 Multi-District Passage from inside a VPC. These clients can now get to information utilizing a solitary worldwide S3 endpoint, including involving AWS Private Connection for S3, and never again need to know which S3 container or AWS Locale it is situated in.

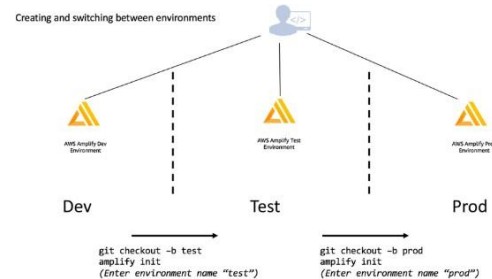
4.6 STARTING WEBAPP FRONTEND



The programme can be divided into two parts: the frontend, which allows users to interact with application resources, and the backend, which houses all of your developed resources. You can utilize many frontends for your application to provide users the greatest experience possible and to benefit from the special capabilities that each platform provides. For instance, you may design several layouts that make use of the wider screen space available in desktop web browsers or mobile applications that make use of the cameras and GPS features of mobile devices.

We take the example of a hypothetical online comic book shop as a guide to examine the key components of a standard web application. This website's function is to let users buy and sell their preferred comic books, receive alerts when new comics are published, and form communities with other users who share their interests. However, we do establish a connection between the business need or use case and the technology that may be used to make it a reality. We don't go into implementing all these things. We use AWS Amplify, a set of tools and services that let developers of mobile and web apps create safe, scalable cloud-powered applications fast and easily, to assist us along the way. The speed at which you iterate and distribute new versions of the programme to your users is crucial while developing an application. The greatest user experience is ensured by having a consistent and seamless workflow from development to deployment, which enables considerably faster iterations and shorter time between releases. A combination of guidelines,

tools, and best practises known as continuous integration and continuous deployment, or CI/CD, enables teams to release code updates more quickly and effectively using CI/CD pipelines. The graphic below depicts a typical development workflow from development to testing and then to production utilizing Amplify to quickly handle various environments and development steps while keeping track of development using source control.



4.7 UPLOAD DOCUMENT

Transfer various records to AWS Cloud Shell utilizing Amazon S3. In AWS Cloud Shell, make a S3 pail by running the accompanying s3 order: `aws s3api make container --can your-can name --area us-east-1`. Then, you really want to transfer the records in a registry from your neighborhood machine to the container. You have two choices for transferring records: Simplified to transfer documents and envelopes to a pail utilizing the AWS The executives Control center. AWS CLI: Utilize the order line to transfer documents and envelopes to the can utilizing the variant of the program that is introduced on your nearby PC. To synchronize the registry in the shell climate with the items in the S3 container, return to the AWS Cloud Shell order line and type the accompanying order: `s3:/your-can name, aws s3 sync`.

4.8 DOWNLOAD DOCUMENT

To match up a S3 can with the items in the ongoing registry in the shell climate, utilize the AWS Cloud Shell order line and the aws s3 order: `S3:/your-container name. aws s3 sync`. The container's items should now be downloaded to your neighborhood PC. You should utilize the AWS CLI instrument that is introduced locally on the grounds that the Amazon S3 console doesn't uphold downloading a few items immediately. Utilize the order: `aws s3 sync s3:/your-can name`.

4.9 GETTING SECRET AWS ACCESS KEYS

Click Users on the navigation menu. Then select a user name for IAM (not the check box), Select Create access key from the Security Credentials page after opening it, Select Show to display the new

access key. Your credentials look like these: Key ID for access: AKIAIOSFODNN7EXAMPLE, Access code: bPxRfCYEXAMPLEKEY/K7MDENG/wJalrXUtnFEMI. Finally Select Download.csv file to download the key pair. Keep the.csv file with the keys in a safe place.

4.10 CONNECTING TO S3 INSTANCE USING BOTO3 CLIENT

Boto3 has both low-level and high-level customers and resources. The higher-level resources for Amazon S3 are the most similar to Boto 2. x's s3 module: `import boto3, s3 = boto3.resource('s3')`. Forming a bucket in Boto 2 and Boto3 is quite similar, with the exception that in Boto3, all action parameters must be given as keyword arguments, and a bucket configuration must be specified manually:

```
s3.create_bucket(Bucket="mybucket")
s3.create_bucket(Bucket="mybucket",
CreateBucketConfiguration={
'LocationConstraint': 'us-west-1'})
```

4.11 ENCRYPTION ALGORITHM

The first step in utilizing Amazon S3 AES-256 encryption is to navigate to the S3 administration. Go to your preferred can to obtain AES 256 encryption. While you're within your S3 container, go to the Properties tab and scroll down to the Default encryption section. It will undoubtedly be labelled as incapacitated. That is, no encryption is currently used. To change the default encryption setting, click the Impaired checkbox.

Next, choose AES-256 as your default encryption and click Save.

That Disabled mark you saw before should now be replaced with AES-256. It indicates that any data you transfer to this S3 will be encrypted with Amazon S3 AES-256 encryption. Go down to the Verify section and thereafter click the Use encryption checkbox. Choose the AES-256 option and then click the OK button. As a result, this S3 trading partner should now be prepared to use S3 AES-256 encryption.

V CONCLUSION

Cloud computing is significant and successful assistance to give different assets to the clients particularly for putting away information. Delicate information is inclined to be more defenseless. Delicate information genuinely must ought to be put away secretly. Subsequently, security of data is the most major and critical point. The security can be achieved by applying cryptographic estimations. Here we have made & executed SDV system model that unexpected spikes popular for a multi-server

SSE plot and showed that encoding records into blocks and moving to a couple of Distributed storage providers includes alright cost above diverged from moving special reports decoded. This proposed system is easy to use and there are many benefits utilizing these methodologies yet it's vital to figure out what level of encryption is expected to play out the errand well in a particular setting. Different encryption strategies are seen in this paper, and this shows that a ton of work has been finished for proposing the Cloud storage encryption components.

REFERENCES

- [1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing (draft)," NIST Spec. 800.145, p. 7, 2011.
- [2] M. J. Atallah et.al, "Dynamic and Efficient Key Management for Access Hierarchies," ACM Transactions on Information and System Security (TISSEC), vol. 12, no. 3, 2009.
- [3] J. Benaloh et.al, "Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records," in Proceedings of ACM Workshop on Cloud Computing Security (CCSW '09). ACM, 2009, pp. 103–
- [4] S. S. M. Chow et.al, "Practical Leakage-Resilient Identity-Based Encryption from Simple Assumptions," in ACM Conference on Computer and Communications Security, 2010, pp. 152–161.
- [5] Chang, Y.C.; Mitzenmacher, M. Privacy Preserving Keyword Searches on Remote Encrypted Data. In ACNS 2005; Ioannidis, J., Keromytis, A.D., Yung, M., Eds.; Springer: New York, NY, USA, 2015; Volume 3531, pp. 442–455.
- [6] Bösch, C.; Peter, A.; Leenders, B.; Lim, H.W.; Tang, Q.; Wang, H.; Hartel, P.H.; Jonker, W. Cloud Searchable Symmetric Encryption. In Proceedings of the Twelfth Annual Conference on Privacy, Security and Trust (PST), Toronto, ON, Canada, 23–24 July 2014; pp. 330–337.
- [7] Ishai, Y.; Kushilevitz, E.; Lu, S.; Ostrovsky, R. Private Large-Scale Databases with Cloud Searchable Symmetric Encryption. IACR Cryptol. ePrint Arch. 2015, 2015, 1190
- [8] Curtmola, R.; Garay, J.A.; Kamara, S.; Ostrovsky, R. Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions. In ACM CCS 2016; Juels, A., Wright, R.N., di Vimercati, S.D.C., Eds.; ACM: New York, NY, USA, 2006; pp. 79–88
- [9]. Song, D.X.; Wagner, D.; Perrig, A. Practical Techniques for Searches on Encrypted Data. In Proceedings of the IEEE Symposium on Security and Privacy (S&P 2000), Berkeley, CA, USA, 14–17 May 2000; p. 44.