

A Cloud Secure Storage Based AES Method of Dispersion

Submitted in partial fulfillment of the requirements for the award of Bachelor of
Technology in Information Technology

By

KUMARAN B (Reg.No - 39120058)

LOKESH N (Reg.No – 39120062)



DEPARTMENT OF INFORMATION TECHNOLOGY

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**

MAY - 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

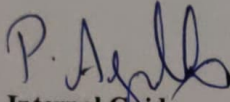
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

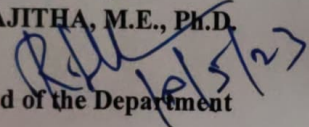
BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Kumaran B (39120058)** and **Lokesh N (39120062)** who carried out the Project Phase-2 entitled "**A CLOUD SECURE STORAGE BASED AES METHOD OF DISPERSION**" under my supervision from January 2023 to May 2023.



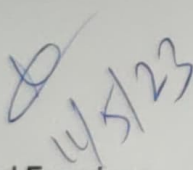
Internal Guide

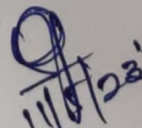
Dr. P. AJITHA, M.E., Ph.D.


Head of the Department

Dr. R. SUBHASHINI, M.E., Ph.D.

Submitted for Viva voce Examination held on 11-05-2023


Internal Examiner

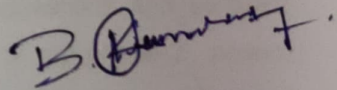

External Examiner

DECLARATION

I, **KUMARAN B** (Reg. No- 39120058) hereby declare that the Project Phase-2 Report entitled **A CLOUD SECURE STORAGE BASED AES METHOD OF DISPERSION** done by me under the guidance of **Dr.P. Ajitha, M.E., Ph.D.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in **Information Technology**.

DATE: 11-05-2023

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. R. Subhashini, M.E., Ph.D.** Head of the Department of Information Technology 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. P. Ajitha M.E., Ph.D**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-2 project work.

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

ABSTRACT

Cloud storage has become an essential component of modern-day computing, providing convenient and cost-effective storage solutions for individuals and organizations. However, the security and privacy of data in the cloud remain a major concern due to the risks of data loss and unauthorized access. This paper presents a novel method of secure cloud storage based on the Advanced Encryption Standard (AES) and dispersion techniques. The proposed method involves encrypting data using AES and then dispersing the encrypted data across multiple cloud storage providers. This ensures redundancy and reduces the risk of data loss. The dispersion technique used ensures that even if an attacker gains access to one cloud storage provider, they will not be able to retrieve the complete data as it is distributed across multiple providers. The proposed method offers enhanced data security and privacy and can be implemented in various cloud storage applications. The effectiveness of the proposed method is demonstrated through experimental results, which show that it is a reliable and efficient solution for cloud storage security.

Chapter No	TITLE		Page No.
	ABSTRACT		v
	LIST OF FIGURES		viii
1	INTRODUCTION		8
2	LITERATURE SURVEY		10
	2.1	Inferences from Literature Survey	10
	2.2	Open problems in Existing System	13
3	CLOUD SECURE STORAGE MECHANISM ANALYSIS		15
	3.1	Requirements Analysis	15
	3.2	Requirement Specification in CSSM	16
4	DESCRIPTION OF PROPOSED SYSTEM		17
	4.1	Selected Methodology or process model	17
	4.2	Architecture Design of Proposed System	18
	4.3	Description of Data Dispersion and Data Encryption	19
	4.4	Prototype Implementation	25
5	EXPERIMENTAL ANALYSIS		27
	5.1	Advanced Encryption Standard (AES)	27
	5.2	Python and Django	27
	5.3	Machine Learning Programming	31
	5.4	System Study	33
6	RESULT AND CONCLUSION		37
	REFERENCES		38
	APPENDIX		
	A. SCREENSHOT		39
	B. RESEARCH PAPER		44

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
4.2.1	CSSM system model architecture	16
4.3.1	CSSM system mechanism	18
4.3.2	CSSM system implementation architecture	19
7.1.1	Home Page in CSSM	32
7.1.2	New Register Page in CSSM	32
7.1.3	User Login Page in CSSM	33
7.1.4	Data Upload Page in CSSM	33
7.1.5	File Uploaded Successfully Page in CSSM	34
7.1.6	Admin Login Page in CSSM	34
7.1.7	Enter Port Number Page in CSSM	35
7.1.8	File Accept Page in CSSM	35
7.1.9	Verified by IPWeb Cam Page in CSSM	36
7.1.10	File Dowload Page in CSSM	36

CHAPTER 1

INTRODUCTION

Cloud computing has shown remarkable development in recent decades. When the storage as a service, it occupies the center stage and backbone for many applications, such as pattern recognition, image forensic and forgery detection. As a result, larger volumes of data will be a part of the cloud area. In the cloud industry, Amazon Web Service (AWS) has become the de facto standard. As the core component of the Open Stack that follows this standard, Swift has become one of the most popular cloud storage mechanisms.

However, Open stack Swift mechanism still faces many real security threats while providing convenient services. According to Cloud Security Alliance's top threat case analysis report released in 2018, two thirds of the cases will cause user data leakage, mainly due to management negligence and malicious attacks. For instance, under default configuration, Open Stack Swift mechanism typically stores data in plaintext for the sake of performance. That will lead unauthorized access to user data at storage layer. In addition, Security Report released by Open stack Vulnerability Management Team VMT, the Swift mechanism may leak user data or configuration information in virtue of security vulnerabilities.

proposed a cloud-oriented data security storage mechanism under the framework of Apache Spark, which prevents data leakage and improves the security of Apache Spark framework. To protect user data on the cloud, different encryption schemes have been adopted to avoid information leakage during machine learning process. Nevertheless, above researches require secure key management mechanisms to prevent cryptographic materials exposure.

Zerfos et al. constructed a secure distributed storage system based on Hadoop system, which keep the confidentiality of cloud data through data dispersion and encryption. It performs the data decryption and assembly tasks before reading data. To prevent the keys from being stolen, this method requires key cache server and all keys should be stored in memory only. Some approaches introduced independent third party to manage the key. It is assumed that third parties stay

trusted. However, the assumption cloud not always exists in the real cloud storage environments.

presented a data privacy preserving scheme for sensor-cloud system, based on edge computing and differential storage method. In this scheme, user data would be divided into different parts and stored in local, edge and cloud layer respectively. But the scheme relies on the characteristics of data from wireless sensor networks, and requires skilled users to manage the edge servers.

To improve the efficiency and decrease the redundancy, Zheng et al. provided a cloud data deduplication scheme to detect and remove identical user data in the cloud. However, from the perspective of preventing data loss due to disaster, a certain number of copies should be sent to multiple regions. In a word, to protect cloud data from leakage at storage layer, this paper presents CSSM, a Cloud Secure Storage Mechanism. CSSM combines data dispersion with data encryption, so that large-scale cloud data and keys would be stored in chunked cipher texts. On this basis, user password and secret sharing are introduced to further protect keys security. We implemented CSSM based on Open Stack Swift mechanism and made several tests.

The major contributions of this work are listed below:

- 1) Data Secure Storage: In order to prevent data leakage and increase the difficulty of attack, this paper presents a method combining data distribution and data encryption to improve data storage security.
- 2) Hierarchical Key Management: To protect the key and prevent the attacker from using the key to recover the data, this paper introduces secret sharing and key hierarchy derivation algorithm in combination with user password to enhance key security.
- 3) Experimental Evaluation and Analysis: The security analysis and experimental results show that CSSM can effectively guarantee the security of data storage, and the increased performance cost is acceptable to users.

CHAPTER 2

LITERATURE SURVEY

2.1 INFERENCES FROM LITREATURE SURVEY

This problem statement has been extensively studied over the past 5 years by researchers and automotive companies in a bid to create a solution, and all their solutions vary from analyzing various patterns of distractive habits to analyzing health vitals of the driver.

1) Title: Capturing-the-Invisible (CTI): Behavior-Based Attacks Recognition in IoT-Oriented Industrial Control Systems

Author: Akashdeep Bhardwaj, Fadi Al-Turjman

Year: 2020

Description: Industrial Control Systems monitor, automate, and operate complex infrastructure and processes that integrate into critical industrial sectors that affect our daily lives. With the advent of networking and automation, these systems have moved from being dedicated and independent to centralized corporate infrastructure. While this has facilitated the monitoring and overall management using traditional detection methods, Web Application Firewalls or Intrusion Detection Systems has exposed the networks subjecting them to Behavior-based cybersecurity attacks. Such attacks alter the control flow and processes and have the malicious ability to alter the functioning of these systems altogether. This research focuses on the use of process analytics to detect attacks in the industrial control infrastructure systems and compares the effectiveness of signature-based detection methods. The proposed work presents a pattern recognition algorithm aptly named as “Capturing-the-Invisible (CTI)” to find the hidden process in industrial control device logs and detect Behavior-based attacks being performed in real-time.

2) Title: Image Forensics Based on Lighting Estimation

Author: Manoj Kumar, Anuj Rani and Sangeet Srivastava

Year: 2019

Description: Computer generated images are assumed to be a key part in each person’s life in this era of information technology, where individuals effectively inhabit the advertisements, magazines, websites, televisions and many more. At the

point when digital images played their role, the event of violations in terms of misrepresentation of information, use of their wrong doings winds up and also becomes easier with the help of image editing application programs. To be legitimate, if anyone does wrong anything then the proposed method can be used for a correct identification of the forgery and the imitations in the digital images. In existing techniques, researchers have suggested most well-known types of digital photographic manipulations based on source, meta-data, image copying, splicing and many more. The proposed approach is inspired by physics-based techniques and requires less human involvement. The presented approach works for images having any type of objects present in the scene, i.e., not only limited to human faces and selection of same intensity regions of the image. By assessing the lighting parameters, the proposed technique identifies the manipulated object and returns angle of incidence w.r.t light source direction. The demonstrated result produces forgery recognition rate of 92% on an image dataset comprising of various types of manipulated images.

3) Title: Secure attribute-based data sharing for resource-limited users in cloud computing

Author: JinLi^a, YinghuiZhang^{bcd}

Year: 2018

Description: Data sharing becomes an exceptionally attractive service supplied by cloud computing platforms because of its convenience and economy. As a potential technique for realizing fine-grained data sharing, attribute-based encryption (ABE) has drawn wide attentions. However, most of the existing ABE solutions suffer from the disadvantages of high computation overhead and weak data security, which has severely impeded resource-constrained mobile devices to customize the service. The problem of simultaneously achieving fine-grainedness, high-efficiency on the data owner's side, and standard data confidentiality of cloud data sharing actually still remains unresolved. This paper addresses this challenging issue by proposing a new attribute-based data sharing scheme suitable for resource-limited mobile users in cloud computing. The proposed scheme eliminates a majority of the computation task by adding system public parameters besides moving partial encryption computation offline. In addition, a public ciphertext test phase is performed before the decryption phase, which eliminates most of computation

overhead due to illegitimate ciphertexts. For the sake of data security, a Chameleon hash function is used to generate an immediate ciphertext, which will be blinded by the offline ciphertexts to obtain the final online ciphertexts. The proposed scheme is proven secure against adaptively chosen-ciphertext attacks, which is widely recognized as a standard security notion. Extensive performance analysis indicates that the proposed scheme is secure and efficient.

4) Title: Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing

Author: Yinghui Zhang, Xiaofeng Chen

Year: 2017

Description: Although many users outsource their various data to clouds, data security and privacy concerns are still the biggest obstacles that hamper the widespread adoption of cloud computing. Anonymous attribute-based encryption (anonymous ABE) enables fine-grained access control over cloud storage and preserves receivers' attribute privacy by hiding attribute information in ciphertexts. However, in existing anonymous ABE work, a user knows whether attributes and a hidden policy match or not only after repeating decryption attempts. And, each decryption usually requires many pairings and the computation overhead grows with the complexity of the access formula. Hence, existing schemes suffer a severe efficiency drawback and are not suitable for mobile cloud computing where users may be resource-constrained. In this paper, we propose a novel technique called match-then-decrypt, in which a matching phase is additionally introduced before the decryption phase. This technique works by computing special components in ciphertexts, which are used to perform the test that if the attribute private key matches the hidden access policy in ciphertexts without decryption. For the sake of fast decryption, special attribute secret key components are generated which allow aggregation of pairings during decryption. We propose a basic anonymous ABE construction, and then obtain a security-enhanced extension based on strongly existentially unforgeable one-time signatures. In the proposed constructions, the computation cost of an attribute matching test is less than one decryption operation, which only needs small and constant number of pairings. Formal security analysis and performance comparisons indicate that the proposed solutions simultaneously

ensure attribute privacy and improve decryption efficiency for outsourced data storage in mobile cloud computing.

5) Title: DivORAM: Towards a practical oblivious RAM with variable block size

Author: ZheliLiu^a, YanyuHuang^a

Year: 2018

Description: Oblivious RAM (ORAM) is important for applications that require hiding access patterns. Many ORAM schemes have been proposed but most of them support only storing blocks of the same size. For the variable length data blocks, they usually fill them up to the same length before uploading, which leads to an increase in storage space and network bandwidth usage. To develop the first practical ORAM with variable block size, we proposed the “DivORAM” by remodeling the tree-based ORAM structure. It employs an additively homomorphic encryption scheme executing at the server side to save the client computing overhead and the network bandwidth cost. As a result, it saves network bandwidth 30% comparing with Ring ORAM and 40% comparing with HIRB ORAM. Experiment results show that the response time of DivORAM is 10×improved over Ring ORAM for practical parameters.

2.2 OPEN PROBLEMS IN EXISTING SYSTEM

While cloud storage providers do most of the heavy lifting when it comes to keeping your data secure, you have an important role to play as well. Here are four things you can do to make sure your cloud storage is as secure as possible.

Turn on two-factor authentication

The majority of cloud storage providers offer an option to protect your account with two-factor authentication. This requires that you have both your password and a one-time verification code, which is sent to your phone or email inbox, to log in to your cloud storage account.

As a result, attackers can't easily break into your account even if they crack your password. Two-factor authentication is typically turned off by default when you set up a new cloud storage account. Make sure you turn this setting on to better protect your data

Protect your encryption key

One major catch to end-to-end encryption is that without your encryption key, which is typically your account password, there's no way to access your data. If you lose your key, your cloud provider won't be able to help. Always keep a backup copy of your key available in case you lose or forget it.

However, you don't want to keep a copy of your encryption key somewhere that hackers could potentially get their hands on it. Consider writing down your key and keeping it somewhere secure. If you want to keep a digital copy of your key, you can use a physical device called a hardware security module. This puts your key behind a firewall, so that it cannot be accessed even if your computer is hacked.

Disconnect your devices

If your computer, smartphone, and other devices stay logged in to your cloud storage account when you're not using them, that's a potential entry point for intruders. By leaving your devices logged in, you allow anyone who gets hold of your device to break into your cloud storage. They don't even need to know your password or circumvent the two-factor authentication process.

Always log out of your cloud account when you're not using it. If your cloud provider offers an automatic time-out feature, turn it on.

Check your shared files

Cloud storage makes it easy to share files with friends, family, and colleagues. But creating links to your files that offer unlimited access can leave your data vulnerable.

Periodically audit which files and folders you've shared from your cloud account, and revoke access for anyone who no longer needs it. If your cloud provider offers links with passwords and expiration dates, put these features to use instead of sharing unrestricted access to your files.

CHAPTER 3

CLOUD SECURE STORAGE MECHANISM ANALYSIS

3.1 REQUIREMENT ANALYSIS

The main objective of the proposed mechanism is to secure cloud storage against data breach, which may be the result of targeted attack (e.g., disk cloning) or management negligence (e.g., misconfiguration), in case hackers even some malicious administrator is able to steal user data.

Aiming at this goal, data dispersion or encryption is the most commonly adopted way in numerous cases. Both techniques could provide privacy-preserving, but they also come with inherent risks. Data dispersion spreads data pieces across different storage areas, but there still lies an opportunity to recover data when attackers obtain enough pieces. Data encryption technology stores data in cipher texts by encrypting data with cryptographic keys. However, attackers can still recover the original data by stealing the keys.

That raises the problem of key protection and management. Therefore, to maximize the confidentiality of cloud data storage, the proposed mechanism should make full use of the advantages of the method and effectively control its disadvantages. Meanwhile, the increased cost of the mechanism should be within a reasonable range.

The project involved analyzing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigations from one screen to the other well-ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

3.2 REQUIREMENT SPECIFICATION IN CSSM

Functional Requirements

- Graphical User interface with the User.

Software Requirements

For developing the application, the following are the Software Requirements:

1. Python
2. Django
3. MySQL
4. MySQL client
5. WampServer 2.4

Operating Systems supported

1. Windows 7
2. Windows XP
3. Windows 8

Technologies and Languages used to Develop

- Python

Debugger and Emulator

- Any Browser (Particularly Chrome)

Hardware Requirements

For developing the application, the following are the Hardware Requirements:

- Processor: Pentium IV or higher
- RAM: 256 MB
- Space on Hard Disk: minimum 512MB

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

4.1 SELECTED METHODOLOGY OR PROCESS MODEL

1. LOGIN / LOGOUT:

In this module the user can login by using their unique username and graphical password. The login module verifies the user given username and password with the stored username and password in the cloud. If the username and password is matched the user can access the resources. If it does not match the user does not allow to access the resource.

2. UPLOAD / DOWNLOAD:

In this module the buyer and seller can post the ads and also able to download the data's that are post by other sellers. This module is mainly used to upload and also download the big data files.

3. DATA DISPERSION:

Two data sets can have the same mean but they can be entirely different. Thus, to describe data, one needs to know the extent of variability. This is given by the measures of dispersion. Range, interquartile range, and standard deviation are the three commonly used measures of dispersion.

4. ENCRYPTION:

There are two basic encryption algorithms for cloud-based data: Symmetric encryption: The encryption and decryption keys are the same. This method is most commonly used for bulk data encryption.

4.2 ARCHITECTURE DESIGN OF PROPOSED SYSTEM

To realize primary object and properties above, this paper presents CSSM, a cloud secure storage mechanism. As shown in Figure 4.2.1, CSSM could be divided into three layers:

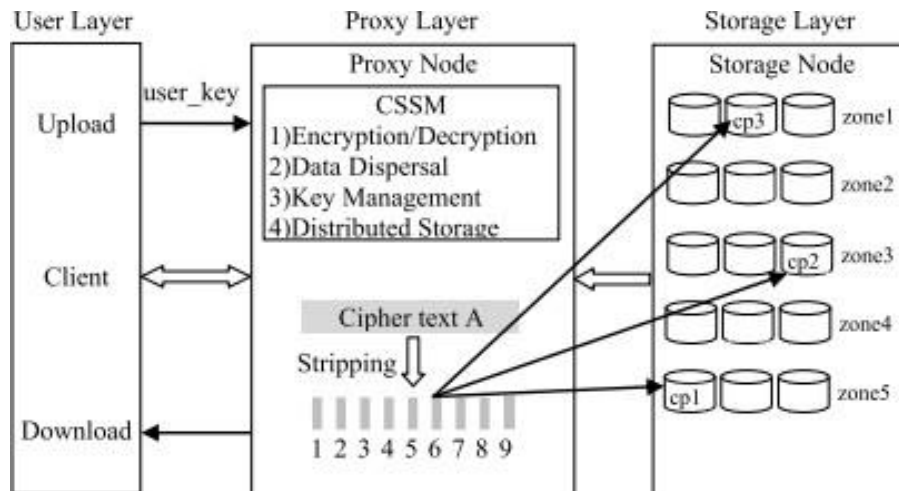


Fig 4.2.1: CSSM System Model Architecture

The user layer, the proxy layer, and the storage layer. Specially, the main functions of each layer are as follows:

1) User Layer: This layer is deployed on the user's machine, and the user operates (upload, download, etc.) cloud data through the client.

2) Proxy Layer: This layer is deployed in the cloud and composed of proxy nodes with trusted execution environments, such as Intel SGX technology and ARM Trust-Zone technology. In trusted execution environment, CSSM programs could perform as expected. CSSM in proxy layer includes four modules: data encryption/decryption, data dispersal, key management and distributed storage.

- Encryption/Decryption: This module is used to encrypt user uploaded data and decrypt user downloaded data.
- Data Dispersal: According to the data dispersal model, the cipher texts are divided into several small blocks.

- Key Management: This module is not only responsible for the generation and maintenance of the key, but also uses the hierarchical key management approach to protect the key.
- Distributed storage: This module distributes chunked and encrypted data to storage layer.

3)Storage Layer: This layer consists of a number of storage nodes that are used to store chunked and encrypted data. Considering data loss or unavailability caused by accident like equipment damage or natural disasters, cloud service providers divide large number of storage nodes into several zones, each of which acts as a failure boundary between multiple copies of the same data.

4.3 DESCRIPTION OF DATA DISPERSION AND DATA ENCRYPTION

From the perspective of improving the confidentiality of user data in the cloud, this paper presents CSSM, a cloud secure mechanism to ensure data security and avoid data breach. The core idea of CSSM mechanism is to increase the difficulty of stealing data. To this end, CSSM process data before stored to cloud storage nodes through two aspects of work. First, CSSM uses data dispersion to divide uploaded file into several parts, and each part is called fragment. When users upload files into cloud, files will be divided and stored as fragments into different storage nodes. Compared with undivided way, our mechanism would contribute to protect data, leading it difficult for attackers to obtain complete data.

On the other hand, although user files are stored in fragments, it is still possible for attackers to recover the fragments to complete user files according to the logical relationship between the contents. Therefore, our mechanism introduces data encryption. The introduction of data encryption technology is helpful for fragmentation cipher-text storage, and it further increases the difficulty for attackers to steal data.

DATA DISPERSION

In order to prevent attackers from stealing complete user data, CSSM first uses data dispersion technology to split data into fragments, and then distribute the fragments to different storage nodes. Due to the uncertainty of the location where data stored, attackers could hardly locate all fragments and recover user files. To further reduce system overhead, CSSM selected DDS—Stripping (1, n,n) [28] as the dispersal model.

However, cloud storage mechanism still needs to restore each fragment when the user downloads files. And CSSM needs to keep the record of the composition relationship between user file and fragment. By stealing the record, attackers have the opportunity to obtain the complete user file. In addition, the code stored in the proxy nodes may be viewed or tampered by malicious entities, and the sensitive information in CSSM may be maliciously spied or stolen during the operation of the system. In response to the above security threats, we believe those can be dealt with by trusted computing technology (such as trusted execution environment and remote attestation technology). The problems that trusted execution environment addresses can be twofold. One is to prevent the code of the proxy service node from being tampered, which usually adopts integrity measurement and remote attestation technology to determine whether the code should be trusted.

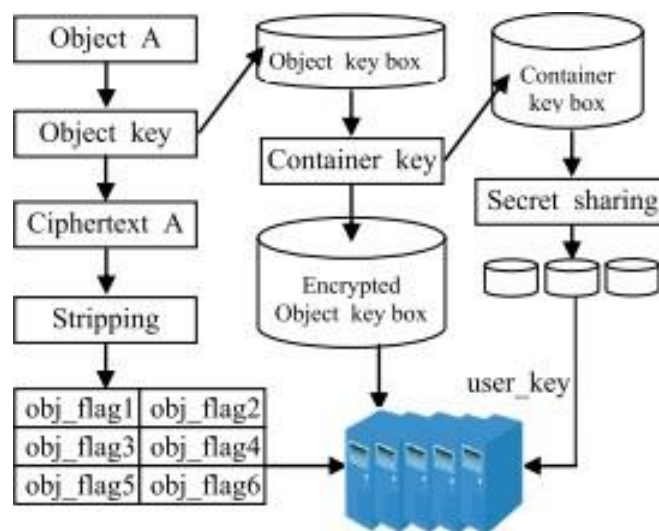


Fig 4.3.1: CSSM System Mechanism

On the other hand, trusted execution environment could ensure that CSSM code runs without interference. We assume CSSM is implemented in trusted execution environment on proxy nodes. Detection mechanism and monitoring mechanism are needed to prevent code tampering and ensure the trusted execution of programs in case of malicious behavior. Nowadays there are many achievements that can be used for reference. CSSM mechanism is based on the research of trusted computing.

DATA ENCRYPTION

Encryption is the most common way to secure user data. In order to reduce time overhead, 128-bit AES symmetric encryption algorithm is selected to implement the encryption and decryption for user data. Key generator should be required in the proxy layer to generate the symmetric key for encryption and decryption. Due to the introduction of data encryption, it brings the necessity and importance of key management. As shown in, we put forward the key hierarchical management method to protect various keys. In cloud storage system, user data is usually regarded as an object and stored in some

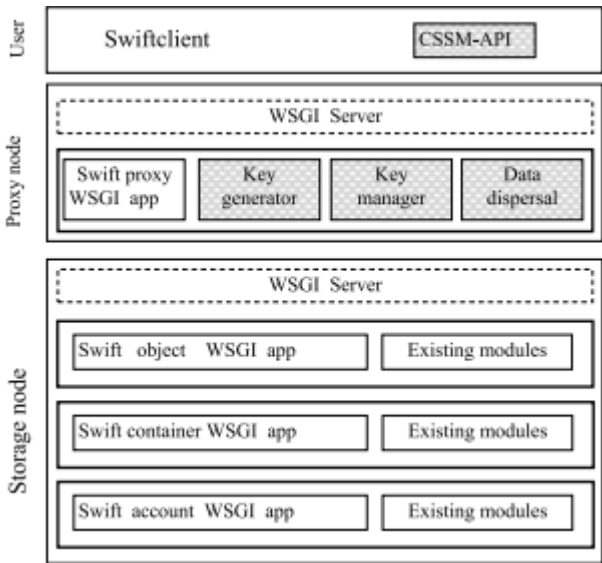


Fig 4.3.2: CSSM System Implementation Architecture

In CSSM, each object and container would be assigned a symmetric key, respectively called object key and container key. Object key would be used for encrypting assigned object. That is to say, the confidential issue of objects will be turned to the same number of object keys.

Besides, we design an object key box for each container, in which the object keys of all object in the container are stored. Then, all the object keys in the object key box are encrypted by their container key and stored in the cloud. Similarly, the confidential issue of object keys will be changed to fewer number of container keys.

In order to guarantee the security of the container key, all the container keys of a user are integrated to form a container key box. And secret sharing algorithm [31] is used to divide container key box into n-block secret sharing blocks whose threshold value is (m, n). To locate these secret sharing blocks, we use the hierarchical key derivation algorithm to establish the index tree whose root node is user_key set by the user. Because the user_key is considered to be a secret only known to the user, it is difficult for attackers to obtain the container key without the user_key, thus ensuring the confidentiality of the container key. In addition, we regard the problem of updating the index tree caused by the change of user_key as a problem to be solved in the future.

CSSM adopts the “dispersion and encryption” strategy for all objects: 1) each object is encrypted to cipher-text first, and then divided into several cipher-text fragments that would be stored in the cloud separately; 2) the object key box consisting of a number of object keys is encrypted by the container key and stored in the cloud; 3) by secret sharing and hierarchical key derivation algorithm, secret sharing blocks are generated and stored in the cloud.

```
[ B0 | B4 | B8 | B12 |
  | B1 | 5 | BS | B13 |
  | B2 | B6 | B10 | B14 |
  | B3 | B7 | B11| B15 ]
```

AES considers each block as a 16 byte (4 byte x 4 byte = 128) grid in a column major arrangement.

Each round comprises of 4 steps:

- SubBytes
- ShiftRows
- MixColumns
- Add Round Key

The last round doesn't have the MixColumns round.

The SubBytes does the substitution and ShiftRows and MixColumns performs the permutation in the algorithm.

SubBytes:

This step implements the substitution.

In this step each byte is substituted by another byte. Its performed using a lookup table also called the S-box. This substitution is done in a way that a byte is never substituted by itself and also not substituted by another byte which is a compliment of the current byte. The result of this step is a 16 byte (4 x 4) matrix like before.

The next two steps implement the permutation.

ShiftRows:

This step is just as it sounds. Each row is shifted a particular number of times.

- The first row is not shifted
- The second row is shifted once to the left.
- The third row is shifted twice to the left.
- The fourth row is shifted thrice to the left.

(A left circular shift is performed.)

MixColumns:

This step is basically a matrix multiplication. Each column is multiplied with a specific matrix and thus the position of each byte in the column is changed as a result.

This step is skipped in the last round.

[c0]	[2 3 1 1]	[b0]
c1	= 1 2 3 1	b1
c2	[1 1 2 3	b2
[c3]	[3 1 1 2]	[b3]

Add Round Keys:

Now the resultant output of the previous stage is XOR-ed with the corresponding round key. Here, the 16 bytes is not considered as a grid but just as 128 bits of data.

DATA DECRYPTION:

The stages in the rounds can be easily undone as these stages have an opposite to it which when performed reverts the changes. Each 128 blocks goes through the 10,12 or 14 rounds depending on the key size.

The stages of each round in decryption is as follows:

- Add round key
- Inverse MixColumns
- ShiftRows
- Inverse SubByte

The decryption process is the encryption process done in reverse so i will explain the steps with notable differences.

Inverse Mix Columns:

This step is similar to the Mix Columns step in encryption, but differs in the matrix used to carry out the operation.

Inverse SubBytes:

Inverse S-box is used as a lookup table and using which the bytes are substituted during decryption.

Summary:

AES instruction set is now integrated into the CPU (offers throughput of several GB/s) to improve the speed and security of applications that use AES for encryption and decryption. Even though its been 20 years since its introduction we have failed to break the AES algorithm as it is infeasible even with the current technology. Till date the only vulnerability remains in the implementation of the algorithm.

4.4 PROTOTYPE IMPLEMENTATION

We implemented CSSM prototype based on OpenStack Swift mechanism. The implementation architecture of CSSM is shown in Table 4.1.1. Swift mechanism follows Client/Server

Item	Proxy Server	Storage Server
Server Model	PowerEdge R230	System X3650 M3
Numbers of Node	1	4
CPU	8 Cores	4 Cores
Network Bandwidth	1000 Mbps	1000 Mbps
Storage Capacity	10 TB	292 GB per node
Operating System	Ubuntu 14.04 LTS	Ubuntu 14.04 LTS

Table 4.1.1: Prototype system experimental environment.

The Swift client consists of a swiftclient program. We modify the swiftclient program to provide the user interface (CSSM-API) for CSSM.

In addition, the Swift Server Program provides the core storage service, which is implemented in the WSGI application. Considering the design principles of CSSM, we implement data dispersal, key generator and key manager in the WSGI application on Swift proxy node.

user data needs to be stored; user initiates a storage request containing user_key through the CSSM-API interface. The proxy node's key generator is

responsible for generating the secret keys upon receipt of the request. The output of the key and encryption operations includes: 1) key manager encrypts these generated keys; 2) data dispersal module splits encrypted data.

When user data needs to be downloaded, user initiates a download request containing the user password through the CSSM-API interface. The key generator on proxy node performs the following operations: 1) recover the container key box by using a keyword hierarchy derivation algorithm and user_key; 2) read specific container keys from the container key box; 3) decrypt and read specific object keys from the container key box; 4) perform decryption operations based on the object key, and return the needed data.

CHAPTER 5

EXPERIMENTAL ANALYSIS

5.1 Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES) is a specification for the encryption of electronic data established by the U.S National Institute of Standards and Technology (NIST) in 2001. AES is widely used today as it is a much stronger than DES and triple DES despite being harder to implement.

Points to remember

- AES is a block cipher.
- The key size can be 128/192/256 bits.
- Encrypts data in blocks of 128 bits each.

That means it takes 128 bits as input and outputs 128 bits of encrypted cipher text as output. AES relies on substitution-permutation network principle which means it is performed using a series of linked operations which involves replacing and shuffling of the input data.

Working of the cipher:

AES performs operations on bytes of data rather than in bits. Since the block size is 128 bits, the cipher processes 128 bits (or 16 bytes) of the input data at a time.

The number of rounds depends on the key length as follows:

- 128 bits key – 10 rounds
- 192 bits key – 12 rounds
- 256 bits key – 14 rounds

5.2 Python and Django

PYTHON:

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in

languages such as C++ or Java. It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library. Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming.

Python features sequence unpacking where multiple expressions, each evaluating to anything that can be assigned to (a variable, a writable property, etc.), are associated in the identical manner to that forming tuple literals and, as a whole, are put on the left-hand side of the equal sign in an assignment statement. The statement expects an iterable object on the right-hand side of the equal sign that produces the same number of values as the provided writable expressions when iterated through, and will iterate through it, assigning each of the produced values to the corresponding expression on the left.

The assignment statement (token '=', the equals sign). This operates differently than in traditional imperative programming languages, and this fundamental mechanism (including the nature of Python's version of variables) illuminates many other features of the language. Assignment in C, e.g., `x = 2`, translates to "typed variable name x receives a copy of numeric value 2". The (right-hand) value is copied into an allocated storage location for which the (left-hand) variable name is the symbolic address.

The memory allocated to the variable is large enough (potentially quite large) for the declared type. In the simplest case of Python assignment, using the same example, `x = 2`, translates to "(generic) name x receives a reference to a separate, dynamically allocated object of numeric (int) type of value 2." This is termed binding the name to the object. Since the name's storage location

doesn't contain the indicated value, it is improper to call it a variable. Names may be subsequently rebound at any time to objects of greatly varying types, including strings, procedures, complex objects with data and methods, etc. Successive assignments of a common value to multiple names, e.g., `x = 2 ; y = 2 ; z = 2` result in allocating storage to (at most) three names and one numeric object, to which all three names are bound. Since a name is a generic reference holder it is unreasonable to associate a fixed data type with it. However, at a given time a name will be bound to some object,

Python has array index and array slicing expressions on lists, denoted as

`a[key],`

`a [start: stop]`

or

`a [start: stop: step].`

Slices take elements from the start index up to, but not including, the stop index. The third slice parameter, called step or stride, allows elements to be skipped and reversed. Slice indexes may be omitted, for example `a [:]` returns a copy of the entire list. Each element of a slice is a shallow copy.

In Python, a distinction between expressions and statements is rigidly enforced, in contrast to languages such as Common Lisp, Scheme, or Ruby. This leads to duplicating some functionality.

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 – You 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.

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 cross- 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 – you 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.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking. It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

DJANGO

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection.

5.3 Machine Learning Programming

Advantages of Machine Learning:

Every coin has two faces, each face has its own property and features. It's time to uncover the faces of ML. A very powerful tool that holds the potential to revolutionize the way things work.

1. Easily identifies trends and patterns -

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and

reminders relevant to them. It uses the results to reveal relevant advertisements to them.

2. No human intervention needed -

With ML, we don't need to babysit our project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus software. they learn to filter new threats as they are recognized.

3. Continuous Improvement -

As ML algorithms gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say we need to make a weather forecast model. As the amount of data, we have keeps growing, our algorithms learn to make more accurate predictions faster.

4. Handling multi-dimensional and multi-variety data -

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

5. Wide Applications -

We could be an e-seller or a healthcare provider and make ML work for us. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

Applications of Machine Learning

Applications of Machine Learning include:

Web Search Engine:

One of the reasons why search engines like google, bing etc work so well is because the system has learnt how to rank pages through a complex learning algorithm.

Photo tagging Applications:

Be it facebook or any other photo tagging application, the ability to tagfriends

makes it even more happening. It is all possible because of a face recognition algorithm that runs behind the application.

Spam Detector:

Our mail agent like Gmail or Hotmail does a lot of hard work for us in classifying the mails and moving the spam mails to spam folder. This is again achieved by a spam classifier running in the back end of mail application.

Database Mining for growth of automation:

Typical applications include Web-click data for better UX, medical records for better automation in healthcare, biological data and many more.

Applications that cannot be programmed:

There are some tasks that cannot be programmed as the computers we use are not modelled that way. Examples include Autonomous Driving, Recognition tasks from unordered data (Face Recognition/ Handwriting Recognition), Natural language Processing, computer Vision etc.

Understanding Human Learning:

This is the closest we have understood and mimicked the human brain. It is the start of a new revolution, The real AI. Now, after a brief insight let's come to a more formal definition of Machine Learning

5.4 System Study

SYSTEM TEST

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- | | |
|---------------|--|
| Valid Input | : identified classes of valid input must be accepted. |
| Invalid Input | : identified classes of invalid input must be rejected. |
| Functions | : identified functions must be exercised. |
| Output | : identified classes of application outputs must be exercised. |
| Systems | : interfacing systems or procedures must be invoked. |

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g., components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing: User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

CHAPTER 6

RESULT AND CONCLUSION

For the issue of cloud data leakage caused by management negligence and malicious attack at storage layer, we proposed CSSM, a cloud secure storage mechanism. CSSM adopted a combined approach of data dispersal and encryption technologies, which can improve the data security and prevent attackers from stealing user data. The experimental results show that CSSM can effectively prevent user data leakage at cloud storage layer. In terms of performance, the increased time overhead of CSSM is acceptable to users. This paper provides a feasible approach to solve the storage security problem, especially prevention from user data leakage at cloud storage layer. CSSM could also effectively protect cryptographic materials from storage perspective.

REFERENCES: -

- [1] A. Bhardwaj, F. Al-Turjman, M. Kumar, T. Stephan, and L. Mostarda, "Capturing-the-invisible (CTI): Behavior-based attacks recognition in IoT-oriented industrial control systems," *IEEE Access*, vol. 8, pp. 104956–104966, 2020.
- [2] M. Kumar, A. Rani, and S. Srivastava, "Image forensics based on lighting estimation," *Int. J. Image Graph.*, vol. 19, no. 3, Jul. 2019, Art. no. 1950014.
- [3] M. Kumar, S. Srivastava, and N. Uddin, "Image forensic based on lighting estimation," *Austral. J. Forensic Sci.*, vol. 51, no. 3, pp. 243–250, Aug. 2017.
- [4] J. Li, Y. Zhang, X. Chen, and Y. Xiang, "Secure attribute-based data sharing for resource-limited users in cloud computing," *Comput. Secur.*, vol. 72, pp. 1–12, Jan. 2018.
- [5] Y. Zhang, X. Chen, J. Li, D. S. Wong, H. Li, and I. You, "Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing," *Inf. Sci.*, vol. 379, pp. 42–61, Feb. 2017.
- [6] The OpenStack Project. OSSA-2015-006: Unauthorized Delete of Versioned Swift Object. Accessed: Apr. 14, 2015.
- [7] The OpenStack Project. OSSA-2015-016: Information Leak Via Swift Tempurls. Accessed: Aug. 26, 2015.
- [8] The OpenStack Project. Possible Glance Image Exposure Via Swift. Accessed: Feb. 23, 2015.
- [9] Cloud Security Alliance. Top Threats to Cloud Computing: Deep Dive. Accessed: Aug. 8, 2018.
- [10] The OpenStack Project. OpenStack Security Advisories. Accessed: Feb. 2, 2015
- [11] Common Vulnerabilities and Exposures. CVE-2015-5223. Accessed: Jul. 1, 2015.
- [12] Common Vulnerabilities and Exposures. CVE-2016-9590. Accessed: Nov. 23, 2016.
- [13] S. Y. Shah, B. Paulovicks, and P. Zerfos, "Data-at-rest security for spark," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Washington DC, USA, Dec. 2016, pp. 1464–1473.
- [14] Z. Liu, Y. Huang, J. Li, X. Cheng, and C. Shen, "DivORAM: Towards a practical oblivious RAM with variable block size," *Inf. Sci.*, vol. 447, pp. 1–11, Jun. 2018.

APPENDIX SCREENSHOTS

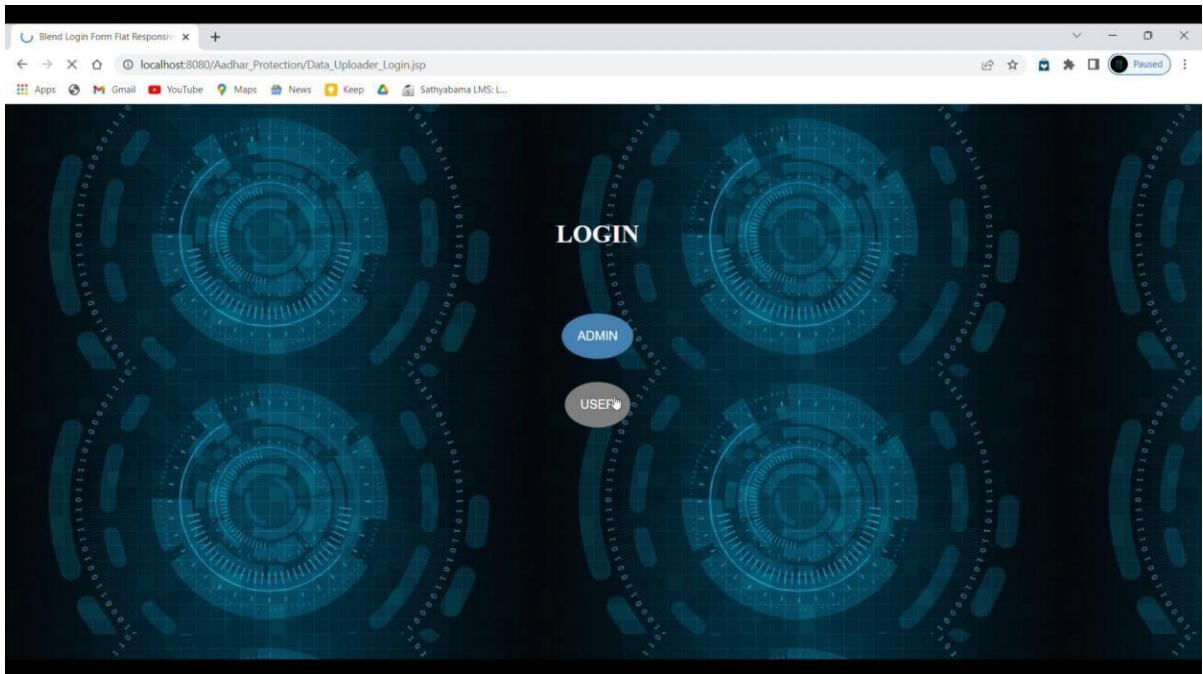


Fig.7.1.1 Home Page in CSSM

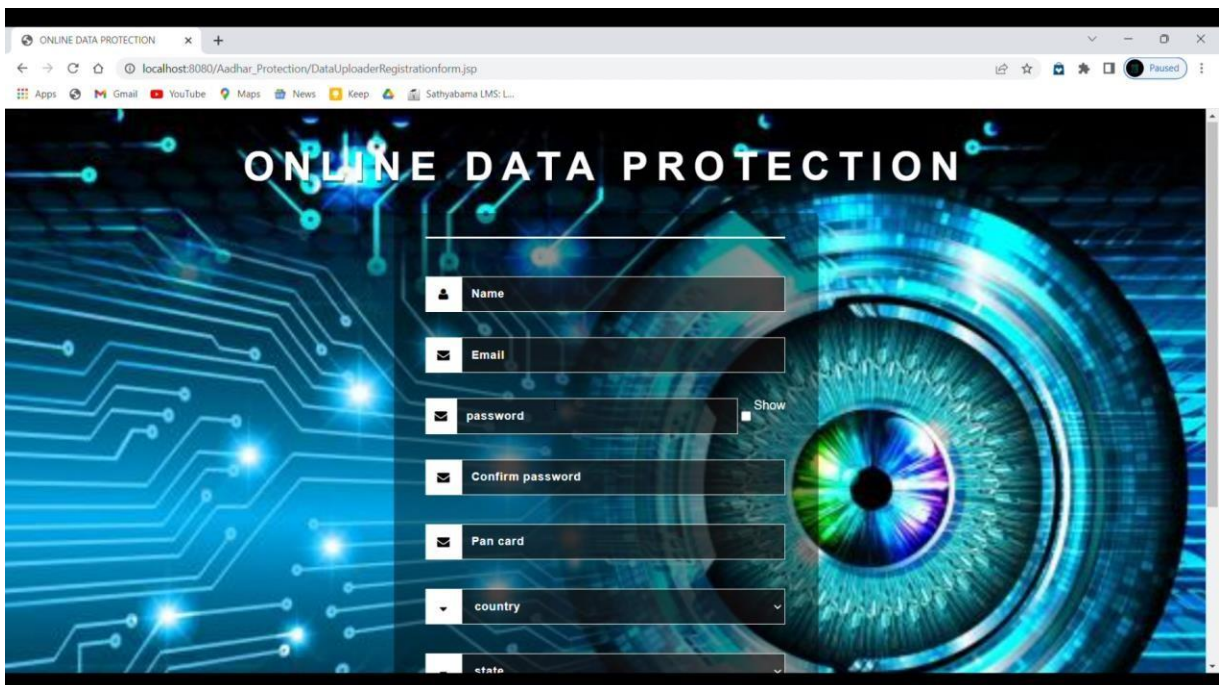


Fig.7.1.2 New Register Page in CSSM



Fig.7.1.3 User Login Page in CSSM

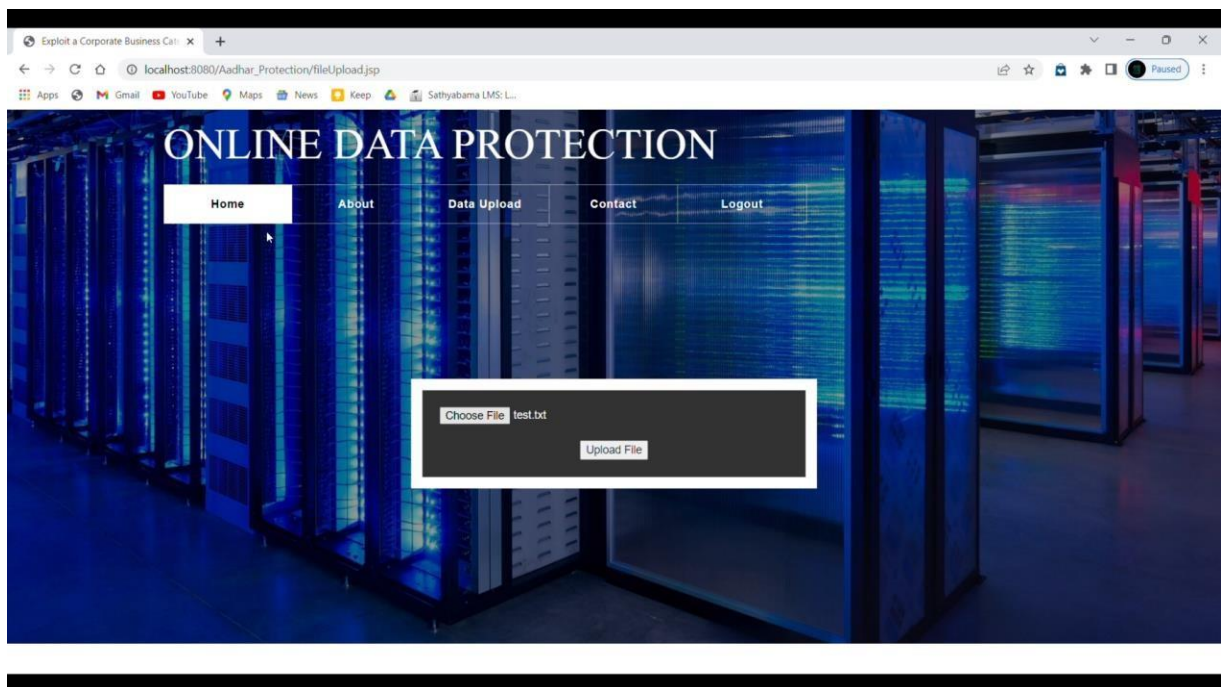


Fig.7.1.4 Data Upload Page in CSSM

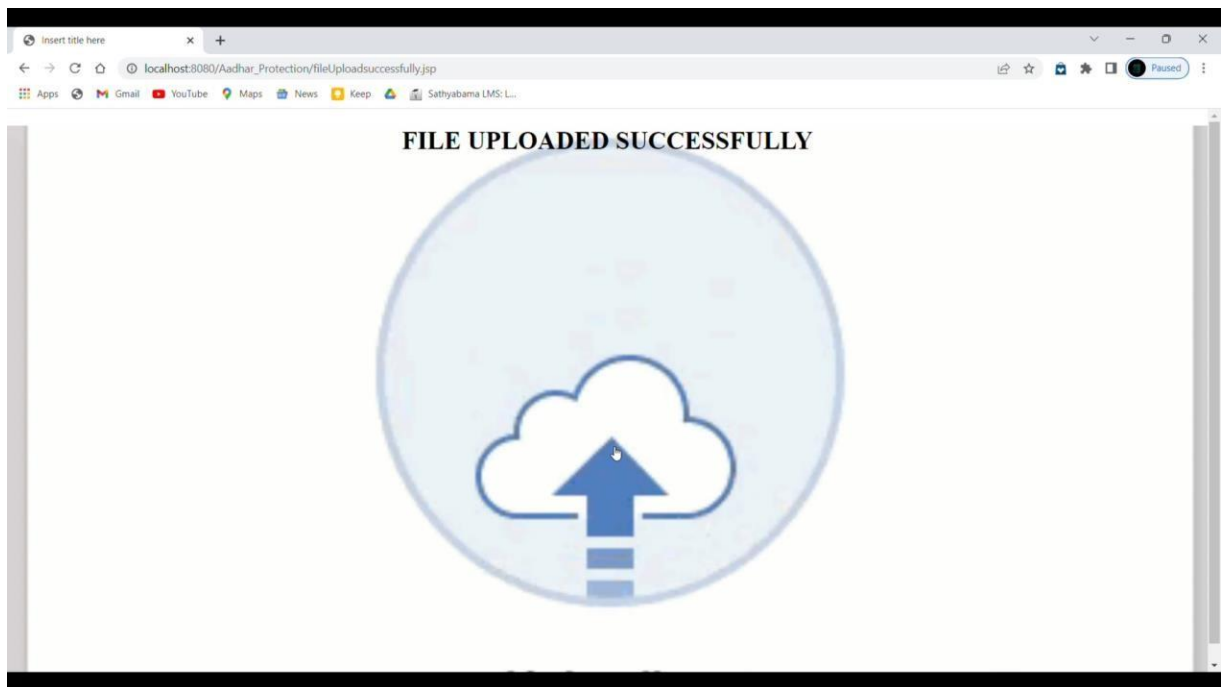


Fig.7.1.5 File Uploaded Successfully Page in CSSM



Fig.7.1.6 Admin Login Page in CSSM



Fig.7.1.7 Enter Port Number Page in CSSM



Fig.7.1.8 File Accept Page in CSSM

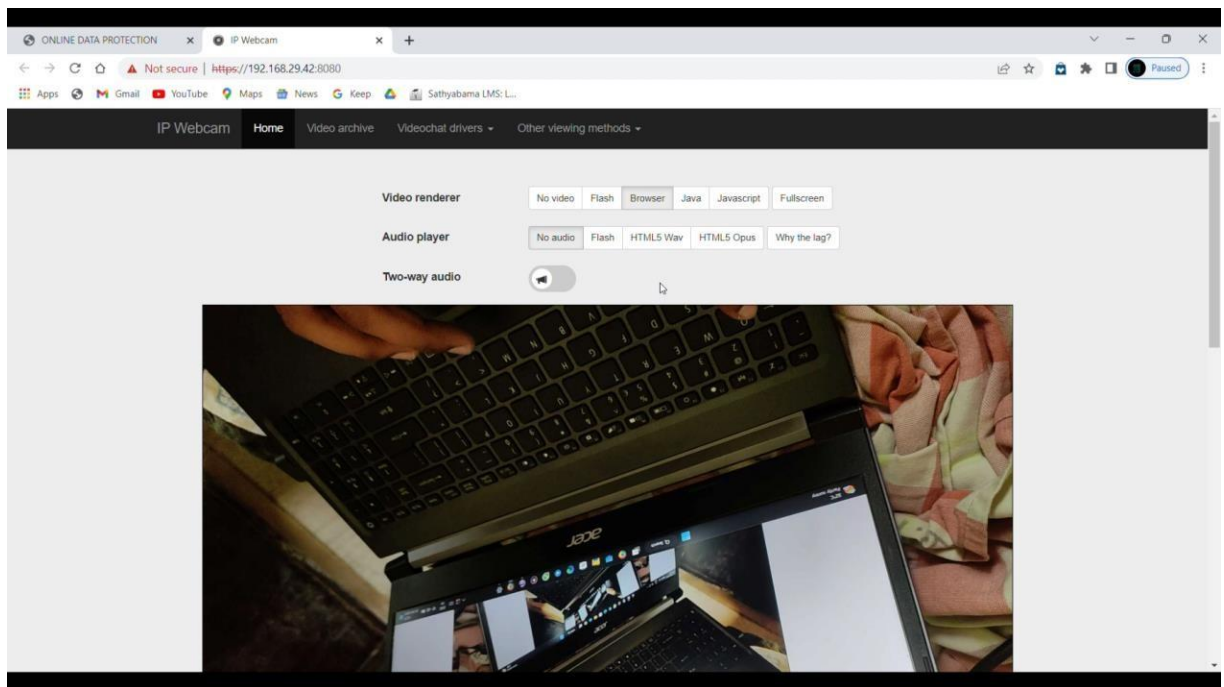


Fig.7.1.9 Verified by IPWeb Cam Page in CSSM

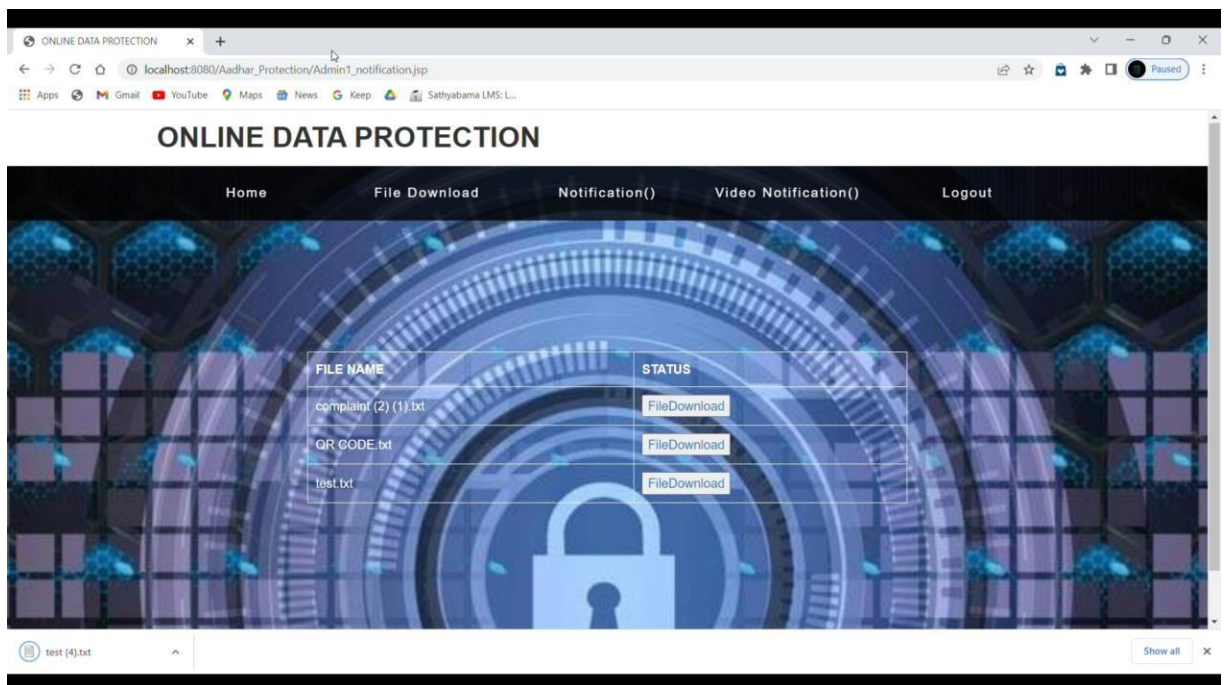


Fig.7.1.10 File Dowload Page in CSSM

Kumaran B
Department Of Information
Technology
Sathyabama Institute of
Science and Technology
Chennai 119, India
sbckumaran@gmail.com

Lokesh N
Department Of Information
Technology
Sathyabama Institute of
Science and Technology
Chennai 119, India
lokeshnithyakumar@gmail.com

Dr. Ajitha
Department Of Information
Technology
Sathyabama Institute of
Science and Technology
Chennai 119, India
Ajitha.it@sathyabama.ac.in

Abstract:

The popularity of cloud storage services is due to their convenience and fast access to data. However, security incidents like data leakage continue to occur, often caused by negligence or malicious attacks. To address this problem, a CSSM has been proposed. The main objective of CSSM is to safeguard the confidentiality of data by utilizing data dispersion and distributed storage. This approach involves encrypting, breaking down, and distributing data, which helps reduce the likelihood of data breaches at the storage layer. This method helps prevent the leakage of cryptographic materials and enhances the overall security of the cloud storage system. Experimental results have demonstrated the effectiveness of CSSM in safeguarding data security. CSSM is a reliable and efficient solution for maintaining cloud data confidentiality and preventing data leakage at the storage layer. It can store a large amount of data in the cloud without significant time overhead. For instance, uploading or downloading a 5GB file with CSSM takes only 646 and 269 seconds, respectively, which is acceptable for users. Overall, CSSM offers a dependable and efficient method for securing cloud data.

Keywords: Cloud computing, data dispersion, data encryption, key management, storage security.

I. INTRODUCTION

Cloud computing has witnessed tremendous growth in recent years, with storage service occupying a prominent position as the backbone of numerous applications, including image forensic, forgery detection, and pattern recognition. As a result, the amount of data being processed in the cloud is increasing rapidly. Among cloud service providers, Amazon Web Service (AWS) is widely regarded as the industry standard. Swift, which is a core component of Open Stack and adheres to this standard, has become widely adopted as a popular cloud storage mechanism.

Despite its convenient services, the Open Stack Swift mechanism is susceptible to several security threats. The Cloud Security Alliance's report in 2018 revealed that in two-thirds of the cases analyzed, user data was leaked due to malicious attacks. Open Stack Swift, a popular cloud storage mechanism, stores data in plaintext by default, which increases the likelihood of unauthorized access to user data stored at the storage layer. Additionally, the Open Stack Vulnerability Management Team's Security Report points out the mechanism's vulnerability to security threats that may result in the leakage of user data or configuration information.

A proposal has been put forth to improve the security of Apache Spark and prevent data breaches. This involves the implementation of a cloud-focused storage system designed to protect sensitive data. This mechanism has been developed within the Apache Spark framework and employs various encryption schemes to protect user data in the cloud during the machine learning process. However, to ensure the overall security of the system, a secure key management mechanism is essential to prevent the exposure of cryptographic materials.

Various methods have been created to enhance the safety and confidentiality of storage systems. In order to prevent the theft of encryption keys, this approach necessitates the use of a key cache server, must be maintained solely in memory. Other have introduced third parties to manage the keys, but this assumes third parties are trustworthy, which may not be the case in real cloud storage environments. A different approach to improving cloud storage security is a scheme that preserves data privacy in sensor-cloud systems. This method utilizes differential storage and edge computing to divide and store user data in local, edge, and cloud layers. However, it requires users with specialized skills to manage the edge servers. A cloud data deduplication scheme has also been proposed to improve efficiency and reduce redundancy, but multiple copies should be sent to different regions to prevent data loss due to disasters.

The paper proposes a solution for safeguarding cloud data from storage layer leaks, by utilizing the CSSM. CSSM is a secure cloud storage mechanism that uses chunked cipher texts to store large-scale cloud data and keys. It achieves this by utilizing data dispersion and encryption techniques. To strengthen key security, user passwords and secret sharing are utilized. The proposed mechanism has been implemented based on the Open Stack Swift mechanism and has undergone multiple tests.

II. LITERATURE SURVEY

The research article titled "Capturing-the-Invisible : method for identifying attacks on Industrial Control Systems that are connected to the Internet of Things (IoT) by analyzing their behavior." discusses the vulnerability of industrial control systems to cyber attacks and proposes a behavior-based detection method called CTI. The proposed method uses process analytics to detect and recognize hidden processes in real-time, making it more effective than traditional signature-based detection methods.

The research article titled "Image Forensics Based on Lighting Estimation" proposes a solution to detect tampered objects in digital images by using physics-based techniques to address the problem of image forgery and manipulation. The proposed method is less dependent on human input and can be applied to any object present in the image, not just limited to human faces. By using lighting parameters, the method can identify the tampered objects. The proposed method accurately detects tampered objects in digital images by determining the angle of incidence relative to the direction of the light source, resulting in a high forgery recognition rate of 92%.

The paper concentrates on attribute-based encryption (ABE) for cloud computing data sharing, but current solutions are problematic due to their high computation overhead and weak data security, limiting their usefulness for mobile users with limited resources. To address these issues, the paper introduces a new ABE scheme that adds system public parameters and conducts a public ciphertext test to reduce computation overhead.

However, current schemes are inefficient and unsuitable from resource constrained mobile users. To overcome these issues, the paper proposes a new approach called match-then-decrypt, to test the attribute key against the hidden access policy. The proposed method generates special components of attribute keys to speed up the decryption process. The proposed technique improves decryption efficiency while preserving attribute privacy and is formally verified for security through performance comparisons.

The paper discusses Oblivious RAM (ORAM) and the challenges of creating a practical ORAM that supports variable block sizes. The authors propose a solution called "DivORAM. Compared to other ORAM schemes, DivORAM shows significant improvements in network bandwidth usage and response time. The experimental results prove DivORAM to be an efficient and practical solution for applications that require hiding access patterns.

III. PROPOSED METHOD

Proposed System

The paper proposes a CSSM that employs data dispersion and distributed storage to prevent data breaches at layer. CSSM uses hierarchical management and combines user passwords and secret sharing to ensure the encrypted, chunked, and distributed storage of cloud data. The experiments demonstrate that the mechanism is effective in preventing data breaches and can handle large amounts of data without significant time overhead. For example, uploading and downloading a 5GB file with CSSM takes only 646 and 269 seconds, respectively, which is reasonable for users.

Advantage

- Object-based storage offers significant benefits in terms of scalability, allowing for cost-effective, high-capacity storage solutions that can easily expand to meet growing demands. With the ability to scale to petabyte and beyond, object-based storage offers a cost-effective and accessible solution for organizations seeking to store and manage large volumes of data.
- The CSSM mechanism integrates data dispersion and encryption to store large-scale cloud data and keys as chunked cipher texts.
- The use of dispersed storage technology provides a high level of privacy and security in both the transmission and storage of data.

Existing system

To ensure data security, a Chameleon hash function is avoided in generating immediate cipher texts. Instead, offline cipher texts are used to blind the final online cipher texts. The paragraph discusses the match-then-decrypt technique that is used in anonymous ABE to allow fine-grained access control for cloud storage while maintaining the privacy of attributes. It involves computing special components in cipher texts to verify whether decryption is allowed or not. With this technique, the attribute key is first matched with the hidden access policy before decryption to avoid revealing any sensitive information. The proposed method generates special attribute key components to ensure fast decryption and improved efficiency while maintaining security. Data dispersion and encryption are commonly used techniques to achieve privacy-preserving, but they also carry inherent risks.

Disadvantage

- The interquartile range is a measure of dispersion that is commonly used, but it has a major drawback: it cannot be easily manipulated mathematically.
- In SSGK, the shared data is encrypted using a group key, and instead of distributing the group key through a secret sharing scheme, another method is used.

Therefore, current schemes are not well-suited for mobile cloud

computing since they have significant efficiency issues and may not be practical for users with limited computing resources.

SYSTEM REQUIREMENT

Hardware Requirement

The system requirements for the project include a Pentium IV 2.4 GHz processor, a 40 GB hard disk, 1.44 Mb floppy drive, a 14" colour monitor, an optical mouse, and 512 Mb of RAM.

Software Requirement

The technical specifications of a computer system and the software used. It mentions that the operating system is Windows 7 Ultimate and the programming language used is Python. The front-end development is also done in Python.

MODULES

1. LOGIN / LOGOUT: The login module is designed to enable users to access resources by entering a unique username and graphical password. Upon receiving the user-provided credentials, the module verifies if the provided username and password match those stored in the cloud. Access to resources is granted if there is a match, and denied otherwise.

2. The Upload/Download module allows both buyers and sellers to upload and share ads, as well as download files posted by other sellers. This module is designed to handle large data files and facilitate the transfer of data between users.

3. When analyzing data, it's not enough to only consider the average value; understanding the variability or spread of the data is equally important. Dispersion measures, including range, interquartile range, and standard deviation, help to quantify this variability. Two datasets with the same mean can have vastly different spreads, making dispersion measures crucial in fully interpreting and comparing data.

4. ENCRYPTION: In cloud computing, there are two primary encryption algorithms: Symmetric encryption, where the key is used from both encryption and decryption. This approach is typically used for bulk data encryption.

ALGORITHM:

Advanced Encryption Standard (AES)

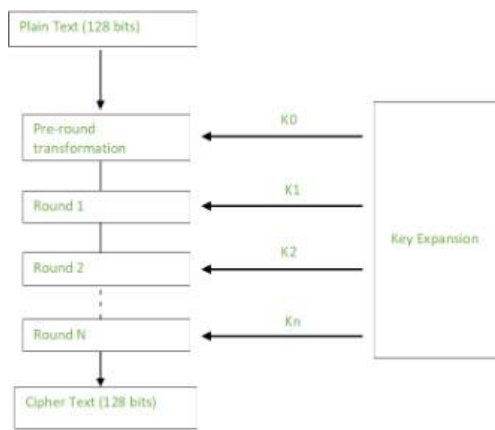
The AES was established in 2001 by the U.S National Institute of Standards and Technology (NIST) as a standard for securing electronic data through encryption. AES is a popular encryption method due to its strength, which surpasses that of DES and triple DES. It operates as a block cipher, where each block of 128 bits is encrypted using a key size of 128, 192, or 256 bits, which involves sequence interconnected operations that substitute and shuffle input data. The result of this encryption is 128 bits of cipher text.

Working of the cipher:

The AES is a block cipher encryption algorithm that operates on data in fixed-size blocks of 128 bits. AES uses a substitution-permutation network principle to replace and shuffle the input data. Despite being more complex than DES and triple DES, AES is considered more robust and secure.

Creation of Round keys:

To encrypt data using AES, the initial key is transformed into multiple round keys using the Key Schedule algorithm. The Key Schedule algorithm is responsible for deriving all the necessary round keys from the initial key. The number of encryption rounds performed by AES varies based on the length of the encryption key. The Key Schedule algorithm plays a crucial role in the AES encryption process by generating the required round keys.



Encryption:

To perform AES encryption, the input data first divided into 128 bits. All block is then represented as a 4x4 grid of bytes in column-major order. Each byte of the grid is treated as a matrix element and the encryption algorithm performs a series of substitutions and permutations on these bytes. This method of representation makes it easier to perform operations on the data using matrix arithmetic.

```
[ b0 | b4 | b8 | b12 |
 | b1 | b5 | b9 | b13 |
 | b2 | b6 | b10 | b14 |
 | b3 | b7 | b11 | b15 ]
```

The AES encryption process consists of multiple rounds, each consisting of four steps. The four steps are: SubBytes, ShiftRows, MixColumns, and Add Round Key. The ShiftRows step shifts the rows of the block by different offsets. The MixColumns step involves performing a matrix multiplication on each column of the block. The Add Round Key step XORs the block on round key derived for original encryption key.

SubBytes:

on ShiftRows and MixColumns steps of the AES algorithm both involve permutation operations that increase the security of the encryption. In ShiftRows, the bytes in each row are shifted by different amounts, while in MixColumns, each column was multiplied by fixed polynomial matrix. These operations work together to spread the influence of each input byte across multiple output bytes, preventing simple algebraic attacks on the encryption.

ShiftRows:

During the ShiftRows step in AES encryption, the 16-byte matrix is manipulated by shifting each of its four rows a certain number of positions. The amount of shift differs for each row, with the first row remaining unchanged. The shift is performed circularly, such that bytes that are shifted out at one end are wrapped back around and shifted in at the other end of the row.

```
[ b0 | b1 | b2 | b3 ]   [ b0 | b1 | b2 | b3 ]
 | b4 | b5 | b6 | b7 | -> | b5 | b6 | b7 | b4 |
 | b8 | b9 | b10 | b11 |  | b10 | b11 | b8 | b9 |
 | b12 | b13 | b14 | b15 | | b15 | b12 | b13 | b14 ]
```

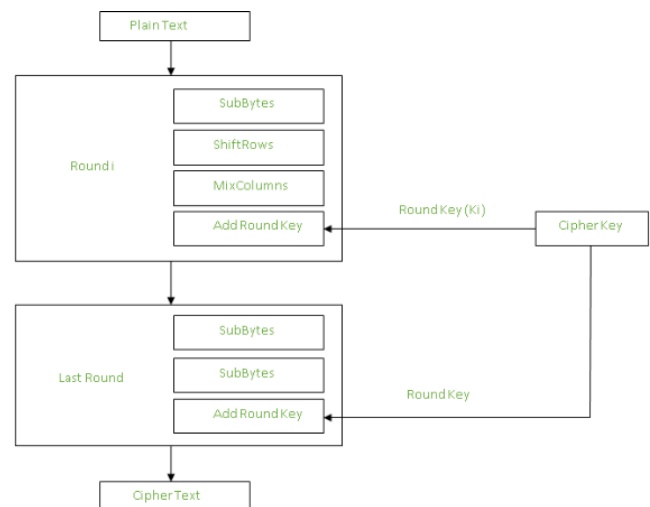
MixColumns:

The MixColumns step in AES involves multiplying all column of the state matrix with a fixed matrix, which results in a new column. This operation is applied to all columns, resulting in a transposition and permutation of the columns in the matrix. It's important to note that this step is not carried out in the final round of the encryption process.

```
[ c0 ]      [ 2 3 1 1 ] [ b0 ]
 | c1 | =   | 1 2 3 1 |   | b1 |
 | c2 |     | 1 1 2 3 |   | b2 |
 [ c3 ]     [ 3 1 1 2 ]   [ b3 ]
```

Add Round Keys:

In the Add Round Key step of AES encryption, the round key derived from the key schedule algorithm is combined with the output of the previous encryption stage using an XOR operation. This step ensures that the output is dependent on both the input data and the encryption key. It is performed in every round of the AES encryption algorithm. This step is applied to each byte in the block, treating the block as a flat array of 128 bits rather than a 4x4 grid of bytes. This process is repeated for each round, including the last round, which does not have the MixColumns step.



Decryption:

The AES decryption process is essentially the inverse of the encryption process, but with some key differences., the decryption process uses the same set of round keys, but in reverse order. The decryption process also has a slightly different sequence of steps than the encryption process.

The decryption process comprises of four stages that are performed in reverse order to the encryption process:

- Add Round Key
- InvMixColumns
- InvShiftRows
- InvSubBytes

The last round does not have the InvMixColumns step.

In the Add Round Key step, the ciphertext XOR-ed with last round key. In the following InvMixColumns step, the columns of the ciphertext are transformed using an inverse matrix multiplication. The InvShiftRows step undoes the row shifting that was done during the encryption process. Finally, the InvSubBytes step undoes the byte substitution that was done during the encryption process.

This process is repeated for all blocks of data that need to be decrypted.

Summary:

The AES instruction set has been incorporated into CPUs. Despite being introduced 20 years ago, the AES algorithm has yet to be cracked, and even with current technology, it remains infeasible to do so. The only known vulnerability is in the implementation of the algorithm.

System Design

Architecture diagram

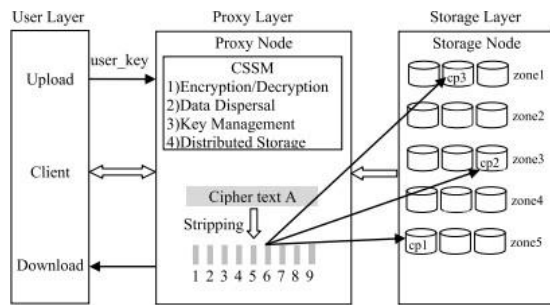


FIGURE 1. CSSM system model architecture

SYSTEM TESTING

PYTHON

Python is popular high-level programming language that interpreted, object-oriented, interactive. with features such as whitespace indentation to delimit code blocks. Python is available on multiple operating systems and has a large community-driven development model, with CPython as its reference implementation .It also includes a comprehensive standard library. One notable feature of Python is its sequence unpacking capability, allowing multiple expressions to be associated in the same way as tuple literals and assigned values from an iterable object on the right-hand side of an assignment statement.

Python is a scripting language that is designed to be easy to read and write. It uses keywords and whites pace indentation to structure code blocks, rather than traditional curly brackets or keywords. This makes it easy to write clear, concise code, even for complex tasks, without the need for a compilation step. Python is also object-oriented, meaning that code is structured around objects, which can contain data and methods. It also comes with a vast standard library that offers various tools and modules for different tasks, such as scientific computing, web development, and data analysis. Python's list data structure provides support for array index and slicing expressions, which allow for efficient access and manipulation of list elements.

Python is popular programming language known for readability, flexibility. It is an interpreted language, meaning that code can be executed without the need for compilation, making it a fast and efficient language to work with. Python's syntax is designed to be easily understood, using English words rather than complex punctuation . It comes with a vast standard library that includes modules for many purposes. The Python Software Foundation oversees Python's development and maintenance, and it has a thriving community of developers who contribute to its growth. Python is a versatile language that can be used for various programming projects, regardless of their scale.

Python is an interpreted and interactive language, which means that you can write and execute code directly

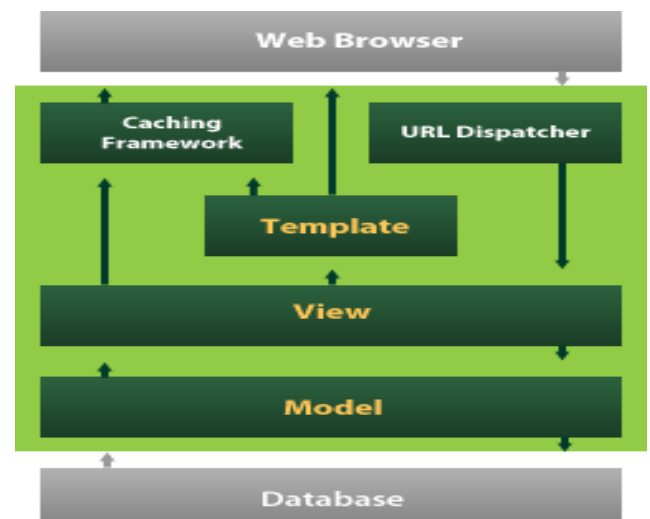
without needing to compile it beforehand. It supports the Object-Oriented programming paradigm and allows you to encapsulate code within objects. Due to its simple syntax and structure, Python is considered a beginner-friendly language.

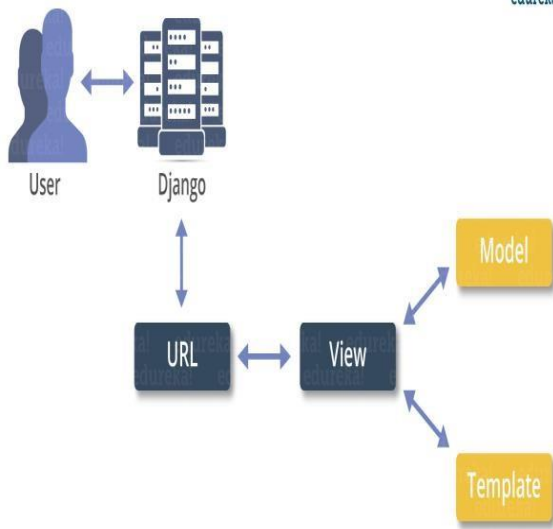
Python's features include –

Python is a user-friendly programming language that offers many benefits, such as ease of use and maintenance, a straightforward syntax, and a comprehensive standard library that works on various platforms. It also provides an interactive mode for testing and debugging, and supports low-level modules for enhanced efficiency. Python supports functional, structured, and object-oriented programming, and. It has dynamic data types, automatic garbage collection, and seamless integration with other programming languages. Additionally, Python supports GUI applications and has interfaces to major commercial databases.

DJANGO

Django is a Python framework designed to facilitate quick development of complex websites by handling many of the common tasks for you. It is created by experienced developers and promotes a pragmatic, clean approach to design. With Django, you can focus on writing your application without having to worry about repetitive tasks. Django emphasizes reusability and modular components, allowing for rapid development and following the principle of avoiding repetition. It is completely free and open source, and Python is the primary language used throughout the framework, including for settings files and data models.





Python – The New Generation Language

Guido van Rossum created Python, which is now maintained by the Python Software Foundation and is a popular programming language. The language features automatic memory management through garbage collection and supports various programming paradigms such as procedural, object-oriented. Python is also known for its large standard library that provides many useful tools and functions to developers.

Features

Python is an interpreted programming language that directly runs source code without a separate compilation and execution process. This flexibility allows developers to choose the programming style that best suits their needs and preferences. For instance, object-oriented programming is suitable for developing large and complex software systems, while functional programming can be used for handling mathematical computations or data analysis. Additionally, aspect-oriented programming can be used for modularizing code and separating cross-cutting concerns, such as logging or error handling, from the core logic of the program. Its simplicity makes it easy to learn, and the emphasis is more on problem-solving than syntax. Python's standard library is extensive and provides support for a variety of tasks, including unit testing, threading, web browsers, cryptography, and GUI development. It is freely available and open-source, allowing users to modify, distribute, and improve the language. The Python community is dedicated to constantly improving and enhancing the language.

Types of Machine Learning

Machine learning algorithms are categorized into four groups based on their input and output data, problem-solving approach, and approach to data. Algorithms enable faster and more precise identification of potential risks or profitable opportunities by analyzing large data sets. However, training machine learning algorithms can be time-consuming and resource-intensive.

Supervised Learning

Supervised learning is a machine learning technique that involves the use of labeled input-output pairs from a known dataset to apply the learned relationship to new input data. This approach,

depending on the type of output data that is expected.

Unsupervised Learning

Unsupervised Learning is a machine learning where data is clustered based on relationships between variables, without prior knowledge of what the solution should look like. This learning method involves identifying unknown patterns in a dataset without predefined labels, making it a self-organized learning approach that doesn't require labeled data or feedback based on prediction results.

Reinforcement Learning

Reinforcement learning is a machine learning approach that involves an agent learning through its interaction with an environment, which actions to take in different situations through trial and error. This type of learning is often used in complex decision-making tasks where it is difficult or impossible to provide labeled training data.

Semi-Supervised Learning

Semi-supervised learning is a machine learning approach that uses unlabeled data for training. It involves a mix of labeled data along with a large amount of unlabeled data to improve learning accuracy. This approach is typically used when obtaining data is costly or difficult, while unlabeled data is easier to obtain.

Objectives

The primary goals of the training were to acquire knowledge and skills in several areas, including program complexity measurement, Python programming, machine learning libraries such as Scikit, Numpy, Matplotlib, Pandas, Theano, and TensorFlow, statistical mathematics relevant to machine learning algorithms, and problem-solving using statistical and mathematical concepts. Additionally, the training covered supervised and unsupervised learning.

ML Algorithms

Machine Learning Programming and Use Case

Advantages of Machine Learning

Machine learning is a transformative technology that offers significant benefits. It has two distinct characteristics, each with its own set of advantages. Machine learning algorithms are capable of identifying hidden patterns and trends that may be challenging for humans to discern. They can operate autonomously without human intervention and improve their performance over time. These algorithms are capable of handling complex and dynamic data in uncertain environments. The applications of machine learning span a wide range of industries, from e-commerce to healthcare, enabling companies to deliver personalized experiences to their customers while also improving their marketing efforts.

Applications of Machine Learning

Applications of Machine Learning include:

Web Search Engine:

Search engines such as Google and Bing are highly effective because they use complex learning algorithms to rank pages. The system has learned to identify patterns and relationships in data that enable it to deliver relevant and accurate results.

Photo tagging Applications:

Face recognition algorithms are used by applications like Facebook and photo tagging applications to recognize and tag friends in photos. This feature allows for a more interactive and engaging user experience, and it is made possible through the use of complex algorithms that can recognize and identify faces in photos.

Spam Detector:

Mail agents such as Gmail or Hotmail use spam filters to automatically categorize incoming emails as either spam or non-spam. The spam filter is powered by a spam classification algorithm that runs in the background of the mail application. The algorithm identifies patterns and features in the email content and header, then assigns a probability score for each email indicating the likelihood that it is spam. Emails with high scores are then moved to a separate spam folder, while legitimate emails are delivered to the inbox.

Database Mining for growth of automation:

Machine learning has various applications, including improving user experience by analyzing web-click data, automating healthcare by analyzing medical records and more.

Applications that cannot be programmed:

Certain tasks are too complex for traditional programming methods to handle, such as autonomous driving, natural language processing, pattern recognition from unstructured data, and computer vision. These tasks require the use of machine learning algorithms that allow computers to learn from data and make decisions based on that learning.

Understanding Human Learning:

The advancement of neural networks techniques have brought us closer to emulating the complex functions of the human brain, ushering in a new era of true artificial intelligence. With this understanding, we can now move on to a more precise explanation of machine learning.

REQUIREMENT ANALYSIS

The objective of the project was to improve the user-friendliness of some applications by analyzing their design. This involved organizing the navigation between screens and minimizing the need for excessive typing. To increase accessibility, the applications

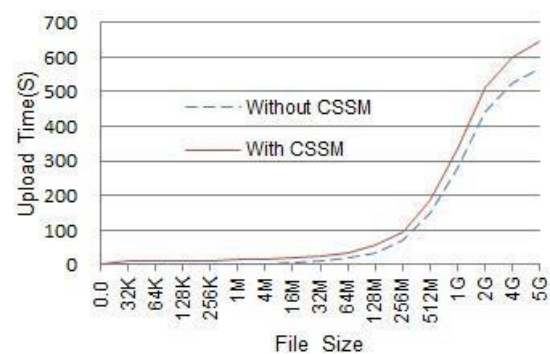
were optimized for use with a variety of browsers.

REQUIREMENT SPECIFICATION

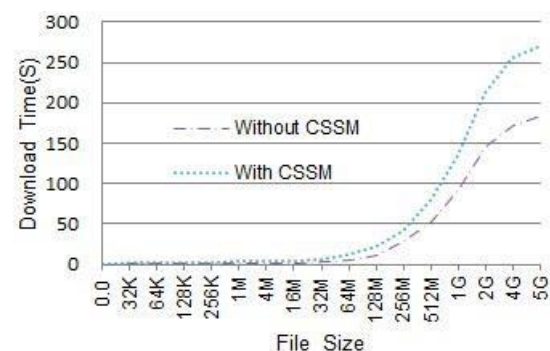
The functional requirements for the project include the need for a graphical user interface to interact with the user. In order to develop the application, certain software requirements need to be fulfilled, which include Python, Django, MySQL, MySQLclient, and WampServer 2.4.

This paragraph lists three operating systems that are supported by a particular software or program. The operating systems listed are Windows 7, Windows XP, and Windows 8.

The project was developed using Python as the main programming language, and any browser (particularly Chrome) for debugging and emulation. The hardware requirements for development include a processor of Pentium IV or higher, at least 256 MB of RAM, and a minimum of 512MB space on the hard disk.



1) File upload time cost comparison.



2) File download time cost comparison.

IV. CONCLUSION

To combat the problem cloud data leakage caused carelessness or deliberate attacks, a secure storage mechanism called CSSM has been suggested. Experiments have demonstrated that CSSM effectively blocks data leakage while maintaining reasonable time overhead. This approach offers a practical solution for storage security concerns, particularly in preventing user data from being leaked through cloud storage. Furthermore, CSSM is capable of securing cryptographic materials at the storage level.

FUTURE SCOPE

We will perform practical tests in an actual cloud environment to evaluate the efficiency of cloud storage when dealing with greater data quantities and access regulations

REFERENCES

- [1] A. Bhardwaj, F. Al-Turjman, M. Kumar, T. Stephan, and L. Mostarda, “Capturing-the-invisible (CTI): Behavior-based attacks recognition in IoT-oriented industrial control systems,” *IEEE Access*, vol. 8, pp. 104956–104966, 2020.
- [2] M. Kumar, A. Rani, and S. Srivastava, “Image forensics based on lighting estimation,” *Int. J. Image Graph.*, vol. 19, no. 3, Jul. 2019, Art. no. 1950014.
- [3] M. Kumar, S. Srivastava, and N. Uddin, “Image forensic based on lighting estimation,” *Austral. J. Forensic Sci.*, vol. 51, no. 3, pp. 243–250, Aug. 2017.
- [4] J. Li, Y. Zhang, X. Chen, and Y. Xiang, “Secure attribute-based data sharing for resource-limited users in cloud computing,” *Comput. Secur.*, vol. 72, pp. 1–12, Jan. 2018.
- [5] Y. Zhang, X. Chen, J. Li, D. S. Wong, H. Li, and I. You, “Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing,” *Inf. Sci.*, vol. 379, pp. 42–61, Feb. 2017.
- [6] The OpenStack Project. OSSA-2015-006: Unauthorized Delete of Versioned Swift Object. Accessed: Apr. 14, 2015.
- [7] The OpenStack Project. OSSA-2015-016: Information Leak Via Swift Tempurls. Accessed: Aug. 26, 2015.