IJITCE

ISSN 2347- 3657

International Journal of Information Technology & Computer Engineering

Email : ijitce.editor@gmail.com or editor@ijitce.com

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 2, 2025

Secure Data Transfer and Deletion Using Counting Bloom Filter (CBF) in Cloud Computing

P. Somesh[1], Mrs. K. Prashanthi[2], M. Nandini[3], P.J.V. Ramachandra[4], Sk. Mohammad Ali[5]

[2] *Department of Computer Science Engineering, Acharya Nagarjuna University, Guntur, Andhra Pradesh - 522510*

[1,3,4,5] *Students, Department of Computer Science Engineering, Acharya Nagarjuna University, Guntur, Andhra Pradesh - 522510*

Email id: someshpyla30@gmail.com [1], prasanthikola17@gmail.com [2], medanandini5@gmail.com [3], ramchandrapyla89@gmail.com [4], skalikhan07@gmail.com [5].

Abstract: -Memory is the basic requirement to store the data which is why data owners prefer to outsource their data to cloud service providers, as this helps in minimizing the efforts involved in maintaining large systems locally. At the same time, Cloud Storage Platforms are being rapidly developed by offering different quality data storage services like Security, Reliability, Accessibility, Speed, Budget-friendly etc. due to which cloud to cloud data transfer has become a strong necessity for data owners to change from one cloud service platform to the other. Therefore, Migration of data from one cloud to the other and the permanent deletion of migrated data should be done in the most secure way. Because, Security is the most challenging requirement in cloud computing. This abstract explains how to solve the above problem of transferring the data from one cloud to the other and to delete the data in source cloud permanently in most secure way. To solve this problem, we construct a new counting bloom filter-based scheme. The counting bloom filter-based scheme can achieve secure data transfer and also can efficiently work on deletion of data permanently. Also, the proposed scheme satisfies the public verifiability without requirement of any trusted third-party applications. Finally, we develop a simulated implementation which demonstrates the practicality and efficiency of our proposal.

Key Words**: -** Cloud Computing, Counting Bloom Filter, Data Transfer, Security.

1. Introduction

In the early days of cloud computing, verifiability in the context of data transfer between cloud platforms was not a primary concern or focus for most cloud service providers. The concept of verifiable data transfer, particularly in a secure and transparent manner, gained more attention as cloud services evolved, but initially, the focus was mainly on basic infrastructure and functionality rather than ensuring the verifiability of data transfers between cloud platforms. The primary concerns for early adopters were scalability, cost-efficiency, and ease of access to resources. Cloud providers like Amazon, Google, and Microsoft focused heavily on reliability and scalability to attract customers.

Data transfer between cloud service providers was more of a manual process initially, often involving simple methods such as using APIs or physical data transfer methods like shipping hard drives for large datasets rather than seamless, secure, and verifiable automated processes due to lack of widely accepted standards and protocols for verifiable data transfer between cloud platforms, which made it challenging to provide such guarantees at scale. Cloud interoperability and Data portability between different providers were only starting to be explored. But Nowadays, Cloud computing has rapidly transformed how data is stored, accessed, and processed. This emerging computer paradigm interconnects massive-scale storage, processing power and highspeed network bandwidth across geographically distributed data centers. Therefore, Data transfer verifiability became a critical issue, particularly for businesses with sensitive data and compliance needs. Hence, concerns around security, data integrity, and trust have led to the development of advanced techniques, such as verification schemes like Counting Bloom Filters and Blockchain technology, to enable verifiable data transfer between cloud platforms—a concept also referred to as data

portability and cloud interoperability.

1.1. The need for Security and Verifiability in Cloud-to-Cloud Data Transfer

Providing Security is one of the most important and challenging tasks in cloud computing. As Cloud computing platforms increasingly move sensitive data such as financial records, personal information, and proprietary business files etc., to the Cloud Memory, ensuring its protection becomes a top priority. The Virtual and Remote nature of cloud environments introduce unique risks which makes them attractive targets for cyber-attacks like data breaches, ransomware, and denial-of-service attacks as data has become a primary requirement for everyone including Cyber-attackers. Without proper security architecture, cloud systems can become vulnerable, potentially leading to data loss, financial damage, and a loss of public trust. Therefore, implementing strong security to any kind of Cloud platform is essential to maintain confidentiality, integrity, and availability of Data in this modern age computing. Verifiability is one of the most important requirements to ensure the integrity of the data which is being transferred and security while transferring the data. Verifiability mechanisms such as checksums, cryptographic hashes, digital signatures, Counting Bloom Filters, Block Chain Technologies guarantees that the data received at the destination is identical and integral to the data sent from the source without any loss or unauthorized modification. Verifiability is mandatory to achieve customer satisfaction and trust in the respective cloud computing platforms.

1.2. Problem Statement

As the volume of digital data continues to grow, cloud storage has become a vital solution for data owners seeking scalable and cost-effective storage. However, with the emergence of various cloud service providers offering diverse benefits such as enhanced security, speed, and affordability, the need to migrate data between clouds has become increasingly common. This cloud-to-cloud migration introduces critical challenges, particularly regarding the secure transfer of sensitive data and the permanent deletion of residual data from the source cloud. Most existing solutions either lack robust security mechanisms or depend on trusted third parties to transfer the data from source cloud to destination cloud, which may introduce vulnerabilities and additional costs. Therefore, there is a need for a secure, efficient, and verifiable framework that enables seamless data migration between cloud platforms while guaranteeing complete data deletion in the source cloud without reliance on a trusted third party.

1.3 Objectives of the Project

The main objective of this project is to develop a secure and efficient mechanism using Counting Bloom Filter technology for migrating data between different cloud service providers by using encryption techniques to protect data during transmission from source cloud to destination cloud and cryptographic erasure methods to make the data inaccessible by destroying the cryptographic keys in the process of data deletion in the source cloud while ensuring the permanent deletion of data from the source cloud without the involvement of trusted third party applications, Thereby enhancing transparency and trust in cloud environments.

The key contributions in the project include

1. *Secure data transfer and deletion:* Addressing the challenges of securely transferring and deleting data on cloud platforms, while also ensuring that the process remains publicly verifiable.
2. *Counting Bloom Filter (CBF)-based scheme:* With a CBF-based approach, it's possible to transfer the data between two different cloud platforms securely. If the source cloud server doesn't migrate or delete the data securely, then the data owner or the destination cloud server can spot these unfinished actions by checking the operational evidences.
3. *Eliminating the Need for a Trusted Third Party (TTP):* Moreover, our proposed scheme does not need any engagement of TTP, which is different from the existing solutions. Finally, the simulation results show that our new approach is both efficient and practical.

1.4. Overview of the Project

A crucial concern in today's cloud computing landscape is about providing security to the cloud platforms especially

while the data is being transferred from one cloud to the other. Therefore, Secure data transfer is ensured through encryption and secure communication protocols, Counting Bloom Filters (CBFs) are used in cloud computing for efficient data storage and retrieval, allowing dynamic insertion and deletion, while secure deletion is achieved using cryptographic erasure and compliance with data protection regulations. These techniques help prevent unauthorized access and ensure data privacy in cloud environments.

The key technologies in this project include

1. Counting Bloom Filter (CBF) Basics: CBFs are extension of Bloom Filters with counters used for dynamic insertion and deletion. These are the common applications in cloud environments. But there is a risk of data exposure during transfer and also there are few challenges to face while ensuring permanent and secure deletion in the source cloud

2. Encryption Algorithms: Use of End-to-End Encryption (E2EE) and Homomorphic Encryption for complete data privacy and protection during data transfer. Usageof Attribute-based encryption for access control.

3. Secure Communication protocols: TLS/SSL for encrypted transmission. TLS abbreviates the Transport Layer Security which is responsible for Encrypting data in Transit, authenticating communication parties and ensuring Data Integrity. SSL encrypts the data during transmission to keep it safe from eavesdroppers by using digital certificates to authenticate servers and has been largely replaced by TLS, which is more secure and updated.

4. Authentication Techniques: Due to its strong security and high collision resistance, we use Hash functions (SHA-256) to verify data integrity and Message Authentication Codes (MACs) to prevent tampering.

5. Secure Deletion Techniques in CBF: Counter decrementing technique for logical deletion and to securely erase sensitive data and prevent recovery through memory forensics and counters or underlying data can be overwritten with random values by ensuring no trace remains of the original content.

6. Cryptographic Erasure Methods: Usage of Key Shredding for Irreversible Data Deletion by which makes the data irretrievable, ensuring secure deletion and Blockchain-Based Verification for auditability and to create a tamper-proof and transparent record. This ensures accountability and auditability.

7. Compliance and Regulatory Requirements: GDPR (General Data Protection Regulation) which is applied by the organizations handling personal data of individuals and CCPA (California Consumer Privacy Act) which applies to collecting of personal data for business purpose. ISO 27001 & NIST standards for cloud security.

2. Literature review

Previous methods for secure deletion and transfer of data from one cloud to the other lacks the data integrity proof. The main disadvantages in existing systems are that they experience performance inefficiency due to weak encryption standards and their inability to detect incomplete deletion or falsified transfers. Traditional methods often rely on trusted third parties or inefficient mechanisms, prompting the exploration of counting Bloom filters as a lightweight, probabilistic data structure to address these issues. The literature review of this project included all the existing systems which either lack data integrity proof or strong encryption techniques and few of them provide the proof of data transfer from one cloud to the other but they couldn't guarantee secure deletion in the source cloud from where the data has been migrated.

2.1 The key projects of the literature review include

- *Xue et al. [19]:* Xue and his team focused on secure data deletion which make sure the data is not only deleted but also unrecoverable when no longer needed. Here in this project, they proposed a method using Key-Policy Attribute-Based Encryption (KP-ABE) which allows fine-grained access control in which only users with the right attributes can decrypt data and they guaranteed assured deletion which means once an attribute is removed, then the associated data becomes inaccessible and effectively deleted. Also, they used Merkle Hash Tree (MHT) which ensures that the data has been truly deleted and helps to verify the integrity of deletion

actions. However, their scheme requires a trusted third-party application, which is a drawback we aim to rectify in this project.

- *Du et al. [20]:* Du and his team designed a scheme called Associated Deletion Scheme (ADM) which ensures data integrity and provable deletion in cloud storage with multiple copies and it uses a pre-deleting sequence and Merkle Hash Trees (MHT) to verify data before deletion. However, their scheme also requires a Trusted Third-Party application (TTP) to manage the data keys.
- *Yang et al. [21]:* In 2018, Yang and his team presented a Blockchain based cloud data deletion scheme in which the cloud executes deletion operation and publishes the corresponding deletion evidence on Blockchain so that the verifier can check the deletion result by verifying the deletion proof. Besides, they solve the bottleneck of requiring a TTP but the main limitation in their project is that there is no guarantee for secure data transfer from one cloud to another cloud.

- *IEEE Transactions on information forensics and security (2019)*, Vol. 14, NO. 2, February 2019. The project paper from IEEE titled "*Enabling Identity-Based Integrity Auditing and Data Sharing with Sensitive Information Hiding for Secure Cloud Storage*" proposes a cloud storage auditing scheme that allows data sharing while hiding sensitive information. It uses a sanitizer to remove sensitive parts and update their signatures so integrity can still be verified. Identity-based cryptography is used to simplify security management. However, their project relies on a Trusted Third Party (TTP) application to transfer the data which we are rectifying in this project.

3. Proposed System

In our proposed system, the project studies the problems of secure data transfer and deletion in cloud-to-cloud data operations and focuses on realizing the public verifiability. Then our project proposes a Counting Bloom Filter-based scheme, with a CBF-based approach, it's possible to transfer the data between two different cloud platforms securely. If the source cloud server doesn't migrate or delete the data securely, then the data owner or the destination cloud server can spot these unfinished actions by checking the operational evidences. Furthermore, we prove that our proposal is able satisfy the desired goals through verified security analysis. These goals include integrity, verifiability, non-repudiation and resilience against insider threats. The structure of the Counting Bloom Filter allows the system to track the existence and removal of data items efficiently without revealing their content. By eliminating the involvement of TTPs, we also improve the Privacy and control that data owners have over their information. Hence, the simulation experiments show that our new proposal is efficient and practical. We evaluated system's performance under various scenarios, including large-scale data transfers and frequent deletion requests and finally found that it operates with minimal computational and communication overhead. These results demonstrate the feasibility of our approach for real-world cloud environments.

3.1. Advantages of our Proposed System

- *Data confidentiality:* When users store their files in the cloud, those files often contain private or sensitive information that must be kept confidential. To ensure that their data stays secure, it's important for data owners to encrypt the files before uploading them to the cloud. By doing this, even if someone gains unauthorized access to the storage system, they won't be able to read or use the actual data. Depending on the situation, users can choose between different types of encryption methods, like symmetric encryption or asymmetric encryption. Encrypting files also gives users more control over who can access their information, especially in shared or untrusted cloud environments. Hence, strong encryption plays a key role in keeping outsourced data safe and secure.
- *Data Integrity:* When data is moved from one cloud to another, it is important to make sure that everything arrives just the way it was sent. Sometimes, Cloud A might only send part of the data or accidentally include files that don't belong. On top of that, data can get corrupted during the transfer just like how a file might get damaged when copied between devices. That's why both the person who owns the data and the receiving cloud (Cloud B) need a way to check that the data is complete and hasn't been changed. Verifying data

integrity gives everyone peace of mind that the information is still accurate, reliable and safe to use after the transfer.

- *Public Verifiability:* While trusting a cloud provider to move or delete your data, there's always a risk that they might not follow through as promised. For example, Cloud A might not actually migrate the data to Cloud B, or it might pretend to delete the data but still keep a copy. That's why it is so important for the data owner to have a way to check on their own that everything was done correctly. This is what we call public verifiability as the ability to confirm that the data was transferred and deleted properly, without having to fully trust the cloud provider's word. It gives users more confidence and control, especially while dealing with sensitive information or switching between cloud services.

4. Methodology: System Design and Implementation

The proposed system is designed to ensure secure data transfer between two different cloud service providers and to enable permanent and verifiable data deletion without the need for a trusted third party. The core of the system is built around a Counting Bloom Filter (CBF), which provides an efficient and lightweight method for tracking and managing data integrity during migration and deletion processes.

4.1. System Requirement Specification

- *Hardware Requirement*
  1. Processor Type: Pentium -IV
  2. ROM: 512 MB
  3. RAM: 4 GB
  4. Hard disk: 20 GB
  5.Monitor: SVG

- *Software Requirement*
  1. Operating System: Windows XP
  2. Tools: Eclipse IDE
  3. Server: Apache tomcat
  4. Database: MySQL
  5.Coding language: Java (JSP, Servlet)
  6. Front-end: J2EE

4.2. Requirement Analysis

Requirement analysis, which is also known as requirement engineering is the process of understanding and defining user expectations out of a new or updated product. It involves identifying, validating and managing the software or system requirements. These requirements should be clear and directly tied to business needs or opportunities. They must also be detailed enough to guide the system's design accurately.

4.2.1. Functional Requirements:

It is a requirement for the software products and marks the beginning of requirement analysis process. It involves identifying and listing the key requirements of a specific software system. Functional requirements define the specific actions the system must be capable of performing.

1. User Authentication: Secure login mechanism for authorized users and two-factor authentication for enhanced security.

2. Data Encryption & Integrity Verification: Encrypts user data using AES (Advanced Encryption Standard) before transfer and generates SHA hash of data for integrity verification.

3. Data Upload to Cloud: Transfers encrypted data to cloud storage and stores corresponding metadata (e.g., SHA

hash) for verification.

4. Counting Bloom Filter (CBF) Implementation: Maintains CBF to track stored data and update counters when data is added or removed.

5. Secure Deletion Request: Allows users to request deletion of specific files and updates the CBF counters upon deletion request.

6. Irreversible Data Deletion: If all CBF counters for a file reach zero, then permanently remove encrypted data from cloud storage.

7. Logging & Auditing: Maintain logs of all upload, access, and deletion activities and provides audit trails for security compliance.

4.2.2. Non- Functional Requirements:

Non-functional requirements define how the system should perform

1. Security: Uses strong encryption (AES-256) for data protection and implements secure key management to prevent unauthorized access.

2. Performance: Ensures encryption and hashing processes do not cause significant delays and optimize CBF operations for minimal memory and CPU usage.

3. Scalability: Supports large-scale data uploads and deletions efficiently and also the system should handle multiple concurrent users.

4. Reliability: Ensures high availability of stored data and prevents accidental data loss through secure deletion mechanisms.

5. Usability: Provides a user-friendly interface for uploading, verifying, and deleting data and also ensures deletion requests are simple and efficient.

6. Compliance & Privacy: Adheres to GDPR, HIPAA and other data privacy regulations and also ensures irreversible deletion meets compliance requirements.

7. Error Handling: Implements robust error detection and logging mechanisms and also provides meaningful error messages in case of failures.

4.3 Code Design

Code Design refers to the process of planning and structuring code before actual implementation.

4.3.1 Input Design

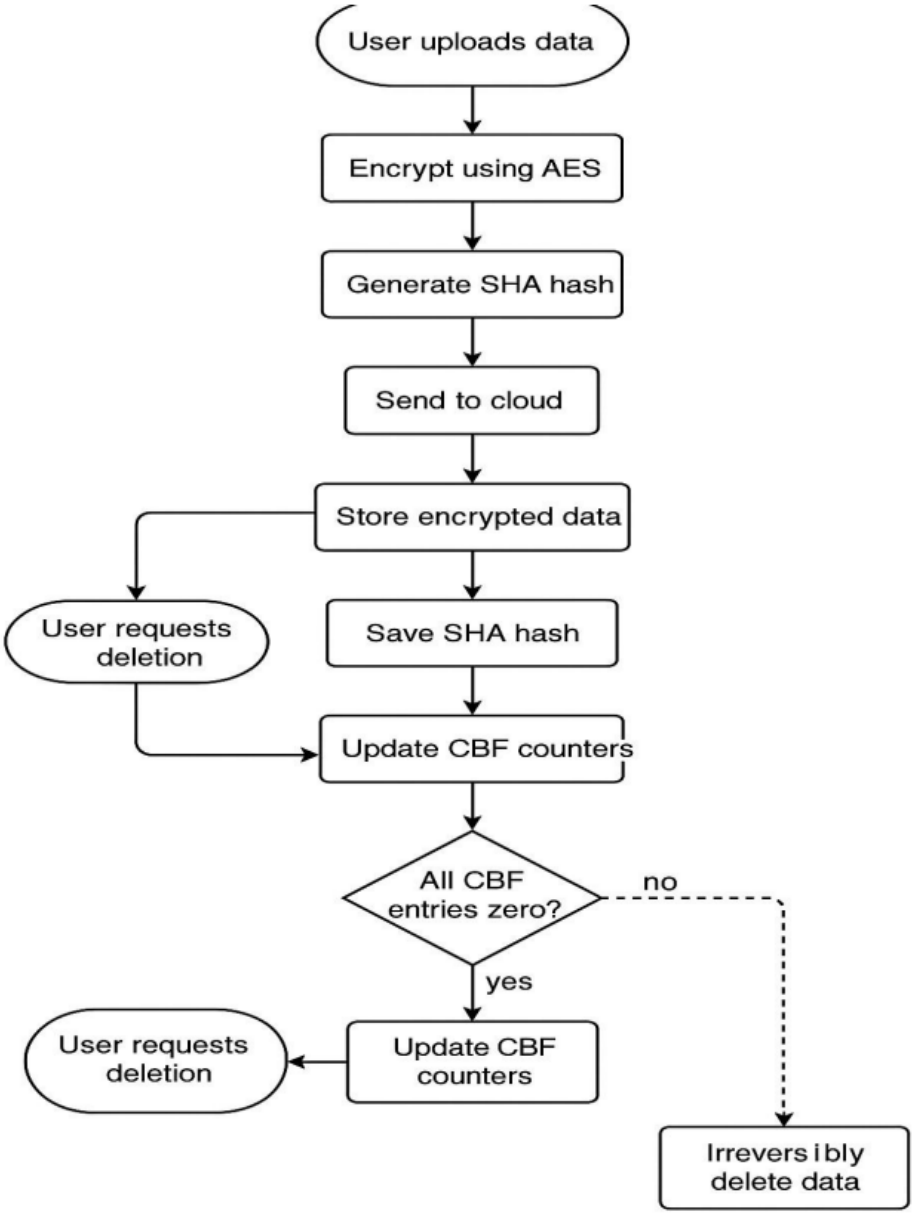
Input design is a critical phase in the software development lifecycle and is directly connected to the performance of the system. The input design ensures that the data entered into an application is executable with maximum accuracy. Therefore, inputs must be designed efficiently to reduce the chances of errors during application execution. Input forms or screens are set up with validation checks to make sure that the given data follows necessary rules. This project has input screens in almost all the modules. Error messages are designed to alert users whenever they make a mistake and guides them to enter the executable data. The errors in the input are controlled by the input design. This application has been developed in a user-friendly manner. The forms are designed to place the cursor automatically in the correct field for input during the executional process. In some cases, users are given the option to choose a set of predefined inputs that are relevant to the application. Validations are required for each data entered. Hence, whenever a user enters an erroneous data, error message is displayed and the user can move on to the subsequent pages after completing all the entries in the current page.

International Journal of Information Technology & Computer Engineering
ISSN 2347-3657
Volume 13, Issue 2, 2025

4.3.2. Output Design

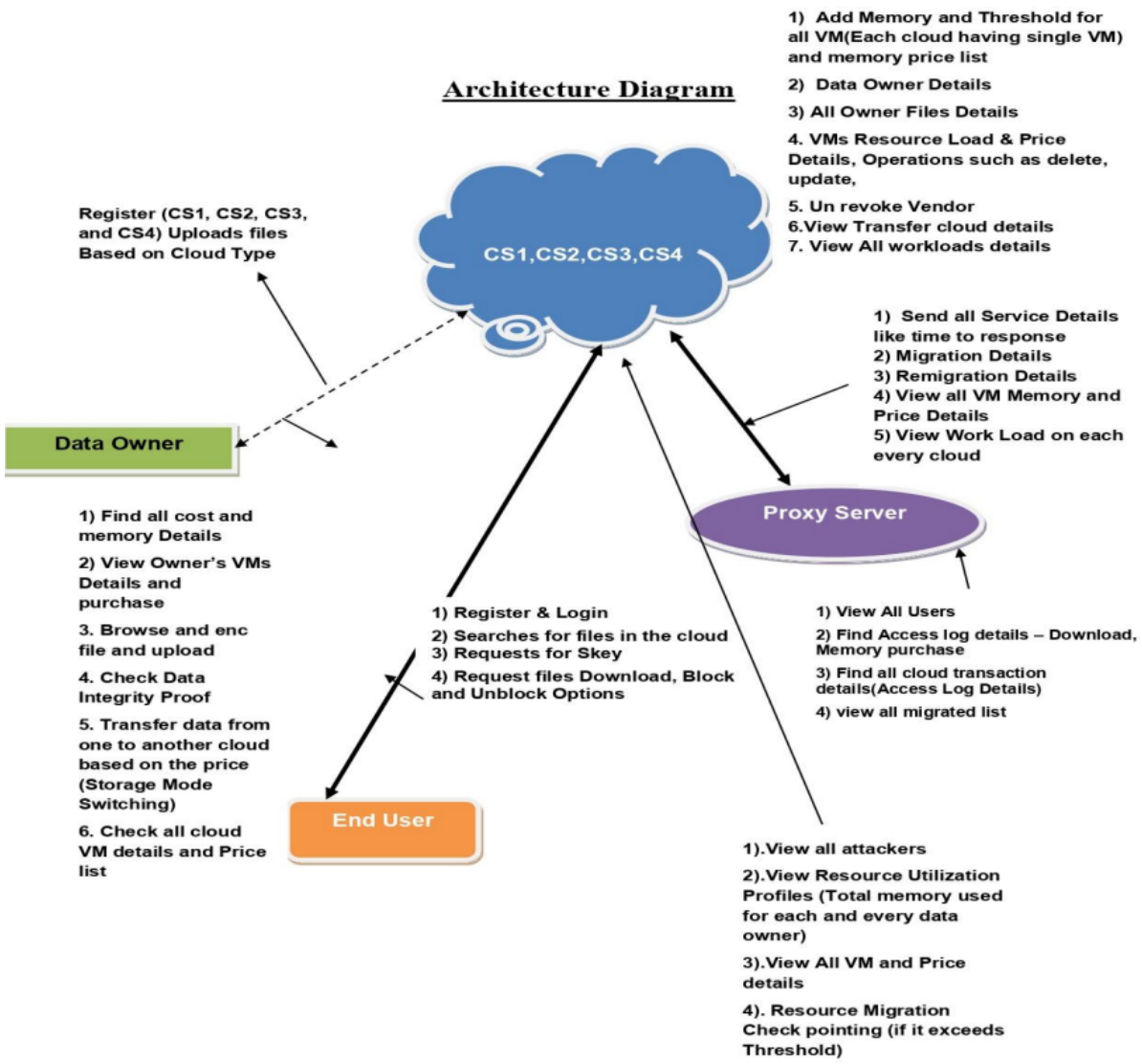
The output from the computer is mainly used to create an efficient way for the company to communicate, especially between the administrator and the clients. The output of VPN is the system which allows the administrator to manage his clients in terms of creating new clients and assigning new projects to them while maintaining a record of the project validity and providing folder level access to each client on the user side depending on the projects allotted to him. After completion of a project, a new project may be assigned to the client. User authentication procedures are maintained at the initial stages itself. A new user can either be created by the administrator or register themselves, but only the administrator has the authority to assign projects and approve new users. The application starts running when it is executed for the first time. The server has to be started and then the internet explorer is used as the browser. The project will run on local area network so the server machine will serve as administrator while the other connected systems can act as clients. The developed system is highly user friendly and can be easily understood by anyone using it even for the first time.

4.4. Activity Diagram



International Journal of Information Technology & Computer Engineering
ISSN 2347-3657
Volume 13, Issue 2, 2025

4.5. General Architecture



Describing the overall features of the software is concerned with defining the requirements and establishing the high level of the system. During architectural design, the various web pages and their interconnections are identified and designed. The main software components are identified and broken down into processing modules and conceptual data structures, along with the connections between the modules.

4.6. Algorithm Design

Cloud computing requires strong security mechanisms to protect data during storage, transmission, and deletion. Therefore, we are using AES, Counting Bloom Filter (CBF), and SHA in Secure Data Transfer and Deletion in Cloud Computing. Here's how AES, SHA, and Counting Bloom Filters (CBF) can be used together to ensure secure data handling.

4.6.1. Secure Data Transfer Using AES and SHA

1. AES for Encryption & Decryption

- Before sending data to the cloud, AES encryption is applied.
- The cloud stores only the encrypted data, preventing unauthorized access.
- On retrieval, data is decrypted using the correct AES key.

ISSN 2347–3657
Volume 13, Issue 2, 2025

2. *Workflow:*

Sender (User)

- Encrypts the data using AES (e.g., AES-256).
- Generates a SHA hash of the plaintext for integrity verification.
- Sends the encrypted data + hash to the cloud.

Cloud Storage

- Stores the AES-encrypted data.
- Stores the SHA hash to verify data integrity during retrieval.

Receiver (User or Service)

- Retrieves encrypted data from the cloud.
- Verifies the integrity using the stored SHA hash.
- Decrypts the data using the AES key.

SHA for Data Integrity Verification

- SHA is used to ensure that the stored or transferred data has not been altered.
- When a user uploads a file, a SHA-256 hash is generated and stored.
- On retrieval, the hash is recomputed and compared to the stored hash.
- If the hashes match, then the data is unchanged. If not, then it may have been tampered with, corrupted, or altered in some way during transmission or storage.

4.6.2. Secure Data Deletion in Cloud Using Counting Bloom Filter (CBF)

- Instead of directly deleting files, CBF is used to track deletions securely.
- A Counting Bloom Filter maintains counters instead of simple bits which allows reversible insertions and deletions.
- Once the count at all hashed positions reaches zero, then the data is permanently removed.

4.6.3. Secure Deletion Workflow

Before Deletion

- Data is indexed in the Counting Bloom Filter using multiple hash functions.
- Each index has a counter that tracks the number of times data has been added.

After Deletion

- Instead of physically deleting, the CBF counters are decremented at the hashed positions.
- If all counters for a particular data entry reaches zero, then the data is marked as irretrievable.

Final Secure Wiping

- If required, AES keys used for encryption can be securely deleted which makes recovery impossible.

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 2, 2025

4.6.4. End-to-End Secure Cloud Data Storage and Deletion System

Combined Approach Using AES, SHA, and CBF

Stage	***Technology used***	***Purpose***
Data Upload	AES Encryption	Data Confidentiality
Integrity Check	SHA Hashing	Ensures zero data tampering
Storage in Cloud	Encrypted AES data + SHA Hash	Secure storage
Data Retrieval	AES Decryption	Decodes data for authorized users
Verification	SHA Hash Check	Confirms Data Integrity
Secure Deletion	Counting Bloom Filter	Ensures permanent data removal

4.7. Module Description

4.7.1. Owner's Module

The Owner Module enables data owners to securely upload their files to the cloud using predefined access policies. The process begins with the owner obtaining a public key specific to the file they intend to upload. After receiving the public key, the owner requests the corresponding secret key which is then used to encrypt and upload the file securely. In addition to uploading files, the owner has access to find all cost and memory details, view owner's VM details and purchase, check data integrity proof and to transfer data between clouds.

4.7.2. User's Module

The Client Module allows users to securely search for and access files stored in the cloud. Clients can search for a file using either the file ID or the file name. If the provided file ID or name is incorrect, the system will deny access. When a valid file is found, the server prompts the user for the secret key to retrieve the encrypted file. To access the decrypted version, the user must supply the correct secret key. In addition to file retrieval, the client can view all attackers, view resource utilization profiles, view all VM and price details, and can also perform resource migration with checkpointing.

International Journal of Information Technology & Computer Engineering
ISSN 2347-3657
Volume 13, Issue 2, 2025

5. System Testing

SI NO	Test Name	Input	Output	Expected Result	Status
1	Owner browsing the file	File	Data stored	Data stored in encrypted form	PASS
	Owner browsing data	data	Data is not Present in directory	Display Data	FAIL
2	Owner Migrate file	Provide details of from and to cloud details	Data migration	Delete details of files in existing cloud	PASS
	Owner Migrate file	no details of from and to cloud details	Data not migration	Delete details of files in existing cloud	FAIL
3	End-user File Search	File name	No. of files from cloud	Got file with same name	PASS
	End-user File Search	No name	File not found	Got file with same name	FAIL
4	End-user File request	File name	sent request success	File Request sent	PASS
	End-user File request	No name	Sent request failed	File Request sent	FAIL
5	End-user File Download	File name	Get Mac and SK	Download file content	PASS
6	End-user File Download	No name	File not found	Download file content	PASS

6. Conclusion

With the rapid growth of cloud computing and storage technologies, data outsourcing has become a common practice among individuals and organizations. However, this paradigm shift has also introduced serious concerns regarding data security, integrity, and trust. Data owners often face the challenge of ensuring that cloud servers perform critical operations such as data transfer and deletion honestly and transparently. Given the lack of complete trust between the data owner and the cloud service providers, it becomes crucial to design mechanisms that enable verification of these operations without relying solely on the cloud's integrity.

To address this challenge, we have proposed a Counting Bloom Filter (CBF) based secure data transfer and deletion verification scheme. This scheme is designed to enhance trustworthiness in cloud environments by allowing data owners to verify both the integrity of transferred data and the authenticity of data deletion. In our approach, when data is transferred from Cloud A to Cloud B, Cloud B is responsible for verifying the integrity of the received data. This verification ensures that the transferred data has not been tampered with or partially lost during migration. Cloud B performs this check using cryptographic proofs and hash values, ensuring the data has been completely and accurately migrated from the source.

Simultaneously, Cloud A is required to provide verifiable evidence of deletion. This is accomplished by using a

Counting Bloom Filter, a probabilistic data structure that can track the presence or deletion of data items efficiently. After deletion, Cloud A generates a deletion proof based on CBF and sends it to the data owner. The data owner can then verify this proof independently to confirm that the deleted data no longer exists on Cloud A. This process ensures that Cloud A cannot falsely claim that the data has been deleted when it is still retained or misused.

Our scheme significantly reduces the possibility of malicious behavior by cloud providers, as they are compelled to provide cryptographically verifiable evidence for both data transfer and deletion. This makes it difficult for any single cloud to act dishonestly without detection. To validate the effectiveness and robustness of our proposal, we conducted extensive security analysis and simulations. The results demonstrate that the proposed scheme is both secure and practical for real world cloud environments. It introduces only minimal overhead while providing a high level of assurance to data owners regarding the integrity and deletion of their outsourced data.

7. References

[1] C. Yang and J. Ye, “Secure and efficient fine-grained data access control scheme in cloud computing”, Journal of High-Speed Networks, Vol.21, No.4, pp.259–271, 2015.

[2] X. Chen, J. Li, J. Ma, et al., “New algorithms for secure outsourcing of modular exponentiations”, IEEE Transactions on Parallel and Distributed Systems, Vol.25, No.9, pp.2386–2396, 2014.

[3] P. Li, J. Li, Z. Huang, et al., “Privacy-preserving outsourced classification in cloud computing”, Cluster Computing, Vol.21, No.1, pp.277–286, 2018.

[4] B. Varghese and R. Buyya, “Next generation cloud computing: New trends and research directions”, Future Generation Computer Systems, Vol.79, pp.849–861, 2018.

[5] W. Shen, J. Qin, J. Yu, et al., “Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage”, IEEE Transactions on Information Forensics and Security, Vol.14, No.2, pp.331–346,2019.

[6] R. Kaur, I. Chana and J. Bhattacharya J, “Data deduplication techniques for efficient cloud storage management: A systematic review”, The Journal of Supercomputing, Vol.74, No.5, pp.2035–2085, 2018.

[7] Cisco, “Cisco global cloud index: Forecast and methodology,2014–2019”, available at: https://www.cisco.com/c/en/us-/solutions/collateral/service-provider/global-cloud-index-gci/white- paper-c11-738085.pdf, 2019-5-5.

[8] Cloudsfer, “Migrate & backup your files from any cloud to any cloud”, available at: https://www.cloudsfer.com/, 2019-5-5.

[9] Y. Liu, S. Xiao, H. Wang, et al., “New provable data transfer from provable data possession and deletion for secure cloud storage”, International Journal of Distributed Sensor Networks, Vol.15, No.4, pp.1–12, 2019.

[10] Y. Wang, X. Tao, J. Ni, et al., “Data integrity checking with reliable data transfer for secure cloud storage”, International Journal of Web and Grid Services, Vol.14, No.1, pp.106–121,2018.