An Optimized Scheme for Assessing Integrity within an Agile Cloud Ecosystem

Sundara Krishnan[1]

Assistant Professor, Department of Computer Science Engineering, Alagappa Chettiar Government College of Engineering and Technology, Karaikudi, India[1]

Abstract: Cloud Computing is regarded as the forthcoming architectural framework for IT enterprises. Unlike conventional solutions, which maintain stringent physical, logical, and personnel controls over IT services, Cloud Computing relocates application software and databases to expansive data centres, where the management of data and services may lack complete reliability. Through cloud computing and storage, users gain the ability to access and share resources provided by cloud service providers at a reduced marginal cost. Furthermore, with cloud computing and storage services, data is not only stored in the cloud but is also frequently shared among numerous users within a group. This project proposes an Operable Proof of Retrievability, a privacy-preserving auditing mechanism for shared data among large groups in the cloud. It employs signatures to generate verification information for shared data, enabling a third-party auditor to assess the accuracy of the shared data without disclosing the identity of the signer for each block. Ultimately, the proposed auditing scheme aims to facilitate efficient public auditing while safeguarding both identity and data privacy within cloud environments.

Keywords: Integrity, auditing, datacentres, cloud computing.

I. INTRODUCTION

Cloud computing represents a modern computing paradigm in which a vast array of systems are interconnected through private or public networks, facilitating a dynamically scalable infrastructure for the storage of applications, data, and files. The emergence of this technology has led to a significant reduction in the costs associated with computation, application hosting, and content storage and delivery. It offers a pragmatic means to achieve direct cost savings and has the potential to revolutionize data centres, transforming them from capital-intensive setups to environments with variable pricing models. The concept of cloud computing is fundamentally rooted in the reusability of IT resources. Unlike traditional models such as "grid computing," "distributed computing," "utility computing," or "autonomic computing," cloud computing expands its reach beyond organizational boundaries. Forrester defines cloud computing as “A pool of abstracted, highly scalable, and managed compute infrastructure capable of hosting end customer applications and billed by consumption”. This technology leverages the internet and centralized remote servers to manage data and applications, enabling consumers and businesses to utilize applications without the need for installation and to access their personal files from any internet-enabled computer. By centralizing data storage, processing, and bandwidth, cloud computing enhances efficiency. Notable examples of cloud computing include services such as Gmail and Hotmail [1, 2].

An individual, organization, or entity that is accountable for providing a service to interested parties is referred to as a Cloud Provider. This entity acquires and oversees the necessary computing infrastructure to deliver these services, operates the cloud software that facilitates the services, and organizes the means to supply cloud services to Cloud Consumers via network access. A Primary Provider is characterized by offering services that are hosted on its own infrastructure. While it may distribute these services to Consumers through a third party, such as a Broker or Intermediary Provider, the key distinction of a Primary Provider is that it does not obtain its service offerings from other Providers. A Cloud Consumer is an individual or organization that engages in a business relationship with Cloud Providers and utilizes their services. The Cloud Consumer explores the service catalogues provided by a cloud provider, requests the necessary service, establishes service contracts, and utilizes the service. Billing for the provisioned service may occur, necessitating the arrangement of payments accordingly. It is important to note that this discussion does not encompass the end user who benefits from the potentially enhanced service provided by the Cloud Consumer. In the context of Software as a Service (SaaS), the Cloud Consumer often coincides with the end user. However, in corporate settings, this is not always the case. For instance, in the case of Gmail, the paying entity, such as the IT department, is recognized as the Cloud Customer, while numerous other employees may utilize the mailing service as end users. A cloud auditor is an entity capable of conducting an independent evaluation of cloud services, information system operations, performance, and the security of the cloud implementation. This party performs an independent examination of cloud service controls with the aim of expressing an opinion on them. Audits are conducted to ensure compliance with standards through the review of objective evidence. A cloud auditor can assess the services offered by a cloud provider concerning security controls, privacy implications, performance, and other relevant factors [3-5].

DOI 10.17148/IARJSET.2017.4824

II. RELATED WORK

Zhu et al. proposed a methodology for verifying the accuracy of business records and financial accounts. An auditor may function as an independent entity, operating autonomously and without affiliation to the company under examination, or may be a captive auditor, with some serving as elected public officials. In the context of cloud computing, the auditor is responsible for assessing the integrity of the cloud environment and monitoring the data stored within it. Given the vast number of users and the potentially enormous volume of data, many users lack the necessary technical expertise and computational resources to conduct a thorough audit of the cloud. Consequently, users may struggle to make informed decisions in the event of data loss. For instance, consider a scenario in 2013 where photographs from a particular event were uploaded to the cloud. By 2030, if an attempt is made to access those images, it may be discovered that only a fraction of the photos is retrievable, with many having been lost. Such losses could result from deletion by the service provider or data corruption [6].

Yuan et al. proposed a model for provable data possession (PDP) that enables a client to confirm the retention of original data by an untrusted server without the need to retrieve the data itself. This model produces probabilistic proofs of possession by randomly sampling sets of blocks from the server, significantly lowering input/output costs. The client is required to maintain a fixed number of metadata to validate the proof. The challenge/response protocol transmits a minimal, constant volume of data, thereby reducing network communication. Consequently, the PDP model facilitates remote data verification for extensive datasets within widely distributed storage systems. This serves as a foundation for related research and the detailed explanation of our methodologies. A PDP protocol ensures that an outsourced storage location holds a file composed of a collection of n blocks [7].

Wang introduced a proof-of-retrievability (POR) scheme, which requires a storage server to demonstrate to a verifier that all data belonging to a client is accurately stored. Although current POR schemes provide satisfactory solutions to various practical challenges, they either exhibit significant communication complexity (linear or quadratic) or are limited to private verification, meaning only the data owner can confirm the integrity of the remotely stored data. The challenge remains to create a POR scheme that simultaneously achieves both public verifiability and constant communication costs. In this article, we address this challenge and present the first POR scheme that offers public verifiability alongside constant communication costs. In our proposed scheme, the communication between the proofer and the verifier consists of a fixed number of group elements. Unlike existing private POR constructions, our scheme facilitates public verification, relieving data owners from the necessity of remaining online. This is accomplished through the innovative combination of techniques such as constant size polynomial commitment and homomorphic linear authenticators. A comprehensive analysis indicates that our proposed scheme is both efficient and practical [8].

Shah developed a highly effective public integrity auditing framework for cloud data sharing that accommodates multiple authors. This innovative approach utilizes polynomial-based authentication tags, enabling the cloud to consolidate authentication tags from various authors into a single entity when transmitting integrity proof information to the verifier. Consequently, the verifier requires only a fixed size of integrity proof information and a consistent number of computational operations, regardless of the size of the audited file or the number of authors linked to the data blocks. Furthermore, the introduction of a novel proxy authentication tag update method permits secure delegation of user revocation tasks to the cloud, effectively countering impersonation threats from unauthorized users. By integrating Shamir’s Secret Sharing scheme, the reliability of our design is further bolstered during the user revocation process. Finally, the proposed framework facilitates the aggregation of integrity auditing tasks for multiple files through our batch integrity auditing technique, thereby enhancing both auditing efficiency and the data corruption [9].

Zheng has developed a novel ring signature scheme that is tailored for public auditing purposes. This work will demonstrate the construction of a privacy-preserving public auditing mechanism for shared cloud data utilizing this innovative ring signature scheme. The objective is to employ ring signatures to obscure the identity of the signer for each block, thereby ensuring that the private and sensitive information of the group remains confidential from the Third Party Auditor (TPA). However, conventional ring signatures are not directly applicable to public auditing mechanisms due to their lack of support for block less verification. In the absence of block less verification, the TPA is required to download the entire data file to confirm the accuracy of the shared data, leading to significant bandwidth consumption and prolonged verification times. Consequently, a new homomorphic authenticable ring signature (HARS) scheme has been constructed, which builds upon a traditional ring signature framework. The ring signatures produced by HARS not only maintain identity privacy but also facilitate block less verification. This proposed approach employs a public key-based homomorphic linear authenticator (HLA), allowing the TPA to conduct audits without necessitating a local copy of the data, thereby significantly reducing both communication and computational overhead compared to conventional data auditing methods [10].

III. PROPOSED METHOD

The system architecture outlined in this project comprises three key entities: the cloud server, a collective of users, and a public verifier, it is shown in Figure 1. Within the user group, there are two distinct categories: the original user and several group users. The original user is responsible for the initial creation of shared data in the cloud and subsequently shares this data with the group users. Both the original user and the group users are considered members of the same group, with each member granted the ability to access and modify the shared data. The shared data, along with its verification metadata, including signatures, is stored on the cloud server. A public verifier, which may be a third-party auditor (TPA) offering specialized data auditing services or an external data user seeking to utilize the shared data, has the capability to publicly verify the integrity of the data stored on the cloud server. When the public verifier intends to assess the integrity of the shared data, it initiates the process by sending an auditing challenge to the cloud server. Upon receipt of this challenge, the cloud server provides the public verifier with an auditing proof that demonstrates its possession of the shared data. The public verifier then verifies the accuracy of the entire dataset by evaluating the correctness of the auditing proof. In essence, the public auditing process functions as a challenge-and-response protocol between the public verifier and the cloud server.

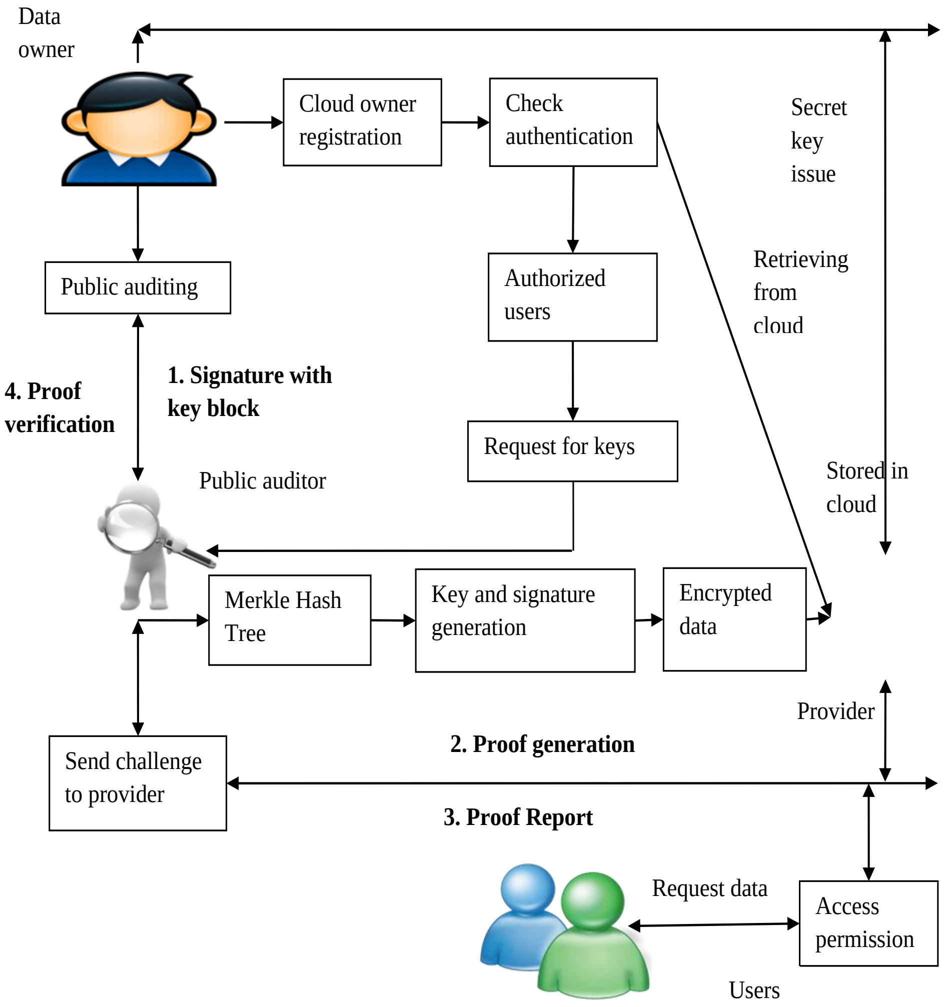


Fig 1: Proposed System Architecture

The current public auditing frameworks can be adapted to ensure the integrity and freshness of shared data. Nonetheless, a notable privacy concern arises when utilizing these existing frameworks for shared data, specifically regarding the potential exposure of identity privacy to public verifiers. To address this privacy challenge, we propose OPOR, an innovative privacy-preserving public auditing mechanism.

DOI 10.17148/IARJSET.2017.4824

This approach employs hash signatures to create homomorphic authenticators within OPOR, enabling a public verifier to confirm the integrity of shared data without needing to access the complete dataset, while simultaneously safeguarding the identity of the signer associated with each block of shared data from public scrutiny. Furthermore, this mechanism is designed to facilitate batch auditing, allowing for the simultaneous execution of multiple auditing tasks, thereby enhancing the efficiency of the verification process. Additionally, OPoR is compatible with random masking techniques; in this initiative, we aim to demonstrate data freshness—ensuring that the cloud retains the most current version of shared data—while maintaining identity privacy. This system guarantees that the retrieved data consistently reflects the latest updates and mitigates the risk of rollback attacks. The Third-Party Auditor (TPA) will be fully automated, effectively overseeing the confidentiality and integrity of the data while seamlessly integrating the random masking technique to establish a privacy-preserving public auditing framework for cloud data storage security, all while adhering to the aforementioned requirements. Comprehensive security and performance evaluations indicate that the proposed mechanisms are both provably secure and highly efficient. Additionally, we illustrate how to extend our primary scheme to accommodate batch auditing for the TPA upon delegation from multiple users.

Merkle Hash Tree:

In order to facilitate privacy-preserving public auditing, we propose a novel integration of the linear authenticator with the binary tree technique. Within our protocol, the linear combination of sampled blocks in the server's response is obscured using randomness generated by the server itself. This random masking ensures that the Third-Party Auditor (TPA) lacks sufficient information to construct an accurate set of linear equations, thereby preventing the derivation of the user's data content, regardless of the number of linear combinations collected from the same set of file blocks. Furthermore, the validation of the correctness of block-authenticator pairs can still be performed in an innovative manner, which will be detailed shortly, despite the introduction of randomness. Our design incorporates a public key-based Merkle Hash Tree (MHT) to endow the auditing protocol with the capability for public audits.

OPOR (Public Auditing):

The implementation of privacy-preserving public auditing allows the Third-Party Auditor (TPA) to manage multiple audits simultaneously based on the delegations from various users. Conducting individual audits for each task can be labour-intensive and inefficient for the TPA. When faced with K auditing delegations concerning K distinct data files from K different users, it is more beneficial for the TPA to consolidate these tasks and perform the audits collectively. To address this inherent requirement, we make a slight modification to the protocol designed for a single user, enabling the combination of K verification equations (corresponding to K auditing tasks) into a singular equation. Consequently, this leads to the development of a secure batch auditing protocol that facilitates the concurrent auditing of multiple tasks. The suggested system offers an enhanced public auditing framework that operates in a decentralized manner, utilizing an outsourced Proof of Retrievability (POR) method to monitor dynamic data alterations. It comprises five distinct modules essential for the successful execution of the project. A cloud framework is employed to delineate the participants involved in the project and to clarify their respective responsibilities. Additionally, the public audit server manages the keys, ensuring verification of data integrity and monitoring dynamic operations within cloud storage systems. The project can be successfully completed with the inclusion of five essential modules. The modules are enumerated as follows: 1. Cloud Framework 2. Key Management 3. Data Integrity 4. Dynamic Auditing 5. Secure Data Sharing A comprehensive description of each module is provided below:

Cloud Framework:

This module discusses three distinct entities involved in cloud data storage services: the cloud user, who possesses a substantial volume of data files requiring storage; the cloud server, overseen by the cloud service provider, which offers data storage services and is equipped with considerable storage capacity and computational resources; and the third-party auditor, who possesses specialized knowledge and skills that the cloud user may lack, and is entrusted to evaluate the reliability of the cloud storage service on the user's behalf when requested.

Key management:

The Merkle hash tree comprises three primary algorithms: KeyGen, Sign, and Verify. During the KeyGen process, each participant within the group creates their own public and private keys. In the Sign phase, a group member can produce a signature for a block and its corresponding block identifier using their private key along with the public keys of all other group members. The block identifier serves as a unique string that differentiates the specific block from others. In the Ring Verify stage, a verifier can ascertain whether a particular block has been signed by a member of the group. MHT encryption systems facilitate operations on encrypted data without the necessity of accessing the private key, thereby allowing computations to be performed without decryption. The client retains sole possession of the secret key. Upon decrypting the outcome of any operation, the result mirrors that of performing the calculation on the original data. This protocol ensures public verifiability without requiring a third-party auditor, safeguarding privacy by preventing any leakage of sensitive information to external parties.

DOI 10.17148/IARJSET.2017.4824

It achieves efficient performance without reliance on a trusted third party and offers a framework for independent arbitration concerning data retention agreements. However, it incurs unnecessary computational and communication costs. In this protocol, the linear combination of sampled blocks in the server's response is obscured using randomness generated by the server.

Data integrity analysis:

In this module, the Third-Party Auditor (TPA) verifies the accuracy of data storage to ensure that no fraudulent cloud server can successfully pass the TPA's audit without genuinely retaining users' data in its original form. This process guarantees that the TPA cannot extract any user data content from the information gathered during the auditing procedure. Subsequently, a batch auditing scheme is implemented in the OPoR framework, empowering the TPA with a secure and efficient auditing capability to manage multiple auditing requests from a potentially large number of different users concurrently. The auditor oversees the interactions between the data owner and the cloud service provider, receiving the metadata of the data component, the tag generation key, and a random challenge from the data owner. Upon making a request to the cloud server, the auditor retrieves the metadata of the data component and, prior to processing the request, verifies authentication and cross-references it with the metadata received from the data owner. The data owner stores the data on cloud servers.

Dynamic auditing:

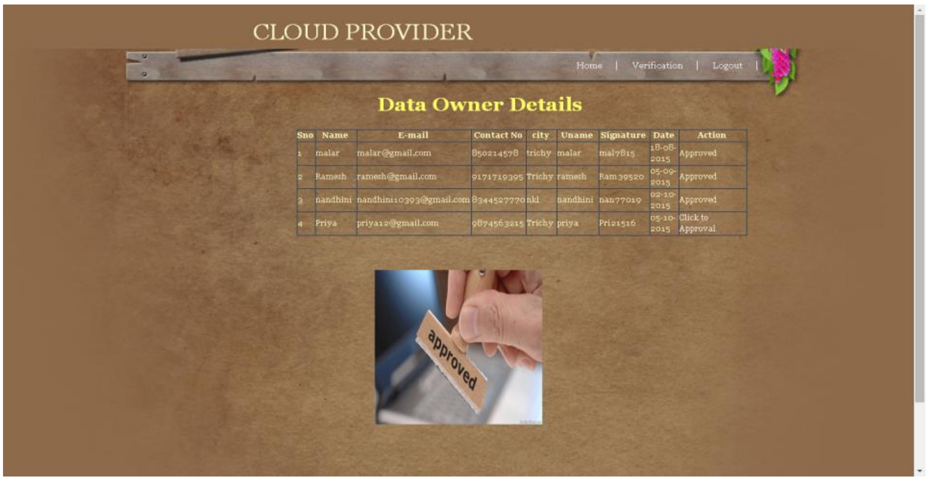
To guarantee data integrity and optimize the computational resources of cloud users while minimizing their online responsibilities, it is essential to implement a public auditing service for cloud data storage. This service enables users to engage an independent third-party auditor (TPA) to assess the outsourced data as required. In this framework, the TPA is empowered to verify the accuracy of the cloud data upon request, without the necessity of retrieving the entire dataset or imposing extra online demands on the cloud users. By facilitating privacy-preserving public auditing in Cloud Computing, the TPA can simultaneously manage multiple auditing requests from various users.

Secure Data sharing:

Every user is designated a data owner from the Provider. Users have the ability to access cipher texts from the server without restriction. To decrypt a cipher text, users must present their secret keys, which are provided by the data owner, along with the corresponding global public key to the server, requesting the generation of a decryption token for a specific cipher text. Once the decryption token is obtained, the user can utilize their global secret key to decrypt the cipher text. Users possessing keys that align with the access policy specified in the cipher text are permitted to access the complete data content.

IV. RESULTS AND DISCUSSION

The results of the proposed are presented in this section. Validation is conducted for each individual field. If any fields are found to be empty, a message box will appear. This validation error can be rectified. An access permission error may occur during the account login process. If the data owner has not obtained permission from the cloud provider, a message will be displayed stating, “Obtain access permission before accessing your account.” Functional testing offers systematic evidence that the tested functions are available as outlined by the business and technical requirements, system documentation, and user manuals. White Box Testing refers to a testing approach where the software tester possesses knowledge of the internal workings, structure, and language of the software, or at least its intended purpose. This method is employed to evaluate areas that cannot be accessed through a black box approach. Conversely, Black Box Testing involves assessing the software without any understanding of its internal workings, structure, or language. Black box tests, like most other testing types, should be derived from a definitive source document, such as a specification or requirements document. Unit testing is typically performed as part of a combined code and unit testing phase within the software lifecycle, although it is not unusual for coding and unit testing to occur as two separate phases.

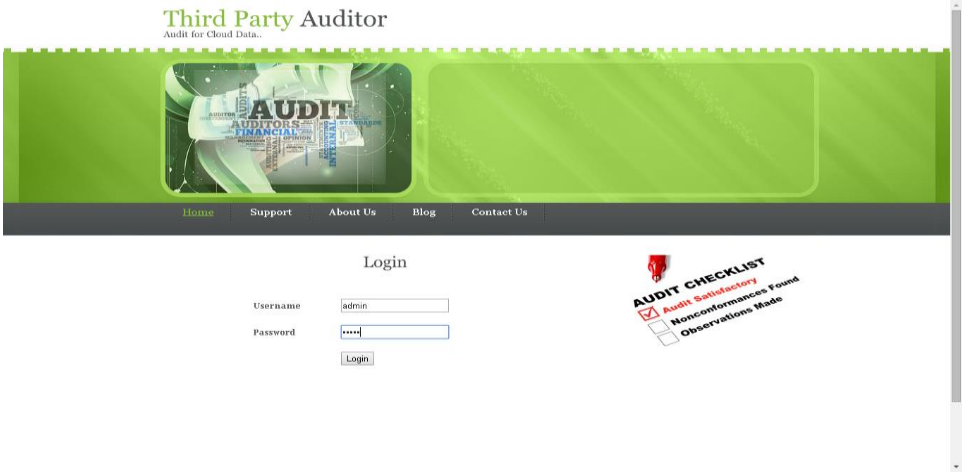


Data Owner Details

Sno	Name	E-mail	Contact No	city	Uname	Signature	Date	Action
1	malar	malar@gmail.com	850214578	trichy	malar	mal7815	18-08-2015	Approved
2	Ramesh	ramesh@gmail.com	9171719395	Trichy	ramesh	Ram39520	05-09-2015	Approved
3	nandhini	nandhini10393@gmail.com	8344527770	nkl	nandhini	nan77019	02-10-2015	Approved
4	Priya	priya12@gmail.com	9874563215	Trichy	priya	Pri21516	05-10-2015	Click to Approval

Figure 2. Data Owner Details

The document includes information regarding the data owner, which encompasses their name, email address, contact number, city, username, date, signature, and the action related to the approval status. The approval status is communicated to the data owner. It is shown in Figure 2.



Login

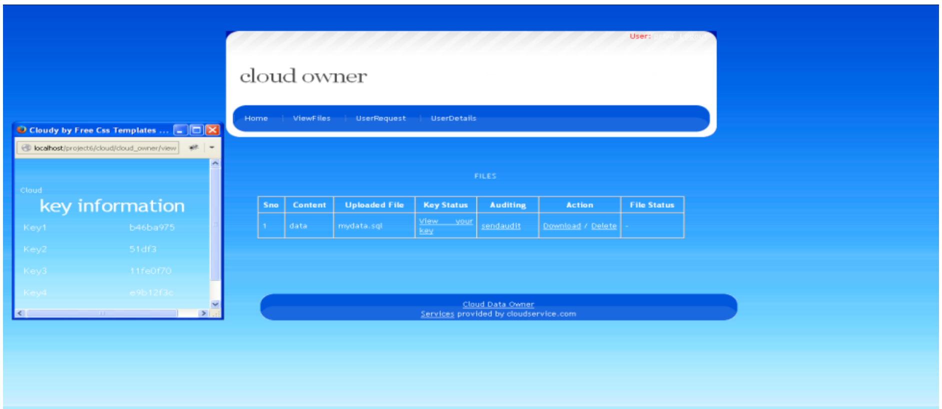
Username: admin

Password: •••••

Login

Figure 3. TPA Login

The cloud owner can review the details of the uploaded file, monitor its key status, conduct audits, and perform actions related to downloading and deleting operations. The final status of the file is updated following the completion of the auditing process. It is shown in figure 4.


Cloudy by Free Css Templates ...
localhost/project6/cloud/cloud_owner/view
cloud owner
User:
Home
ViewFiles
UserRequest
UserDetails
Cloud Data Owner
Services provided by cloudservice.com

Cloud

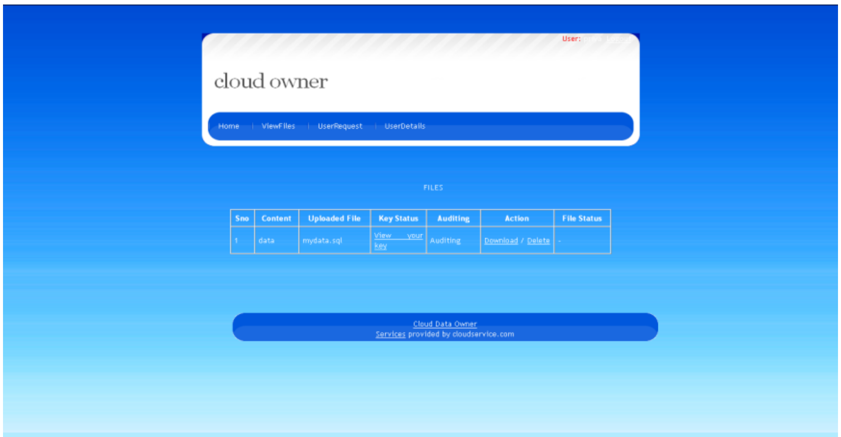
key information

Key1	b46ba975
Key2	51df3
Key3	11fe0f70
Key4	e9b12f3c

FILES

Sno	Content	Uploaded File	Key Status	Auditing	Action	File Status
1	data	mydata.sql	View your key	sendaudit	Download / Delete	-

Figure 4. Key information



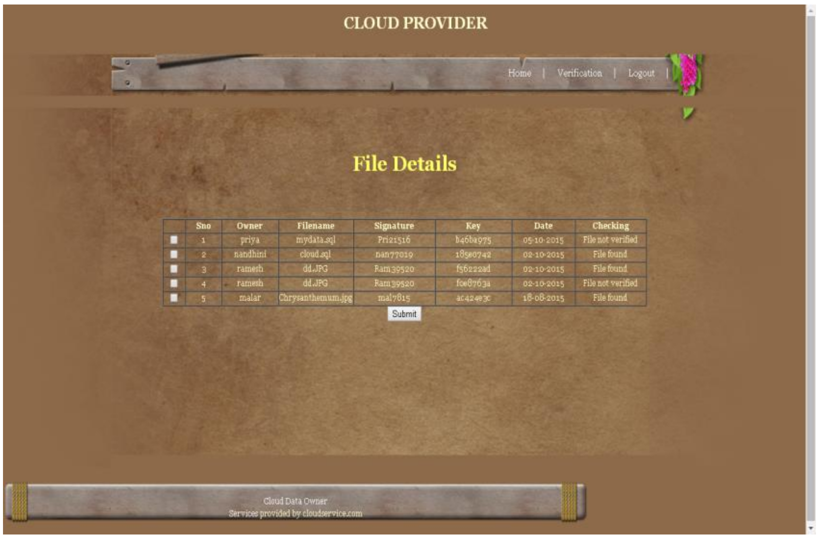
FILES

Sno	Content	Uploaded File	Key Status	Auditing	Action	File Status
1	data	mydata.sql	View your key	Auditing	Download / Delete	-

Cloud Data Owner
Services provided by cloudservice.com

Figure 5. Auditing Status

The text outlines the process of auditing in the Cloud. A hash signature is created utilizing the MHT system. The data owner specifies the key blocks. During the auditing phase, a message containing the signature and key blocks is transmitted to the Third Party Auditor (TPA). It is shown in figure 5.

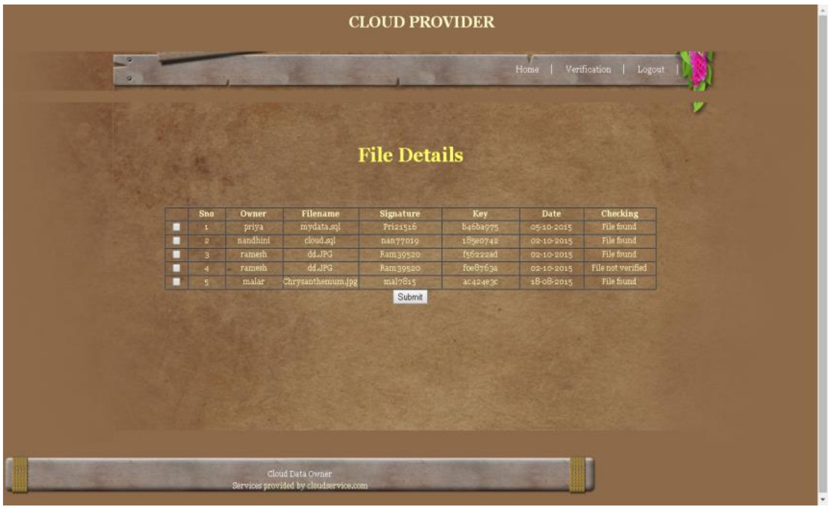


File Details

	Sno	Owner	Filename	Signature	Key	Date	Checking
□	1	priya	mydata.sql	Pri21516	b46ba975	05-10-2015	File not verified
□	2	nandhini	cloud.sql	nan77019	18580742	02-10-2015	File found
□	3	ramesh	dd.JPG	Ram39520	f56222ad	02-10-2015	File found
□	4	ramesh	dd.JPG	Ram39520	f0e8763a	02-10-2015	File not verified
□	5	malar	Chrysanthemum.jpg	mal7815	ac424e3c	18-08-2015	File found

Cloud Data Owner
Services provided by cloudservice.com

Figure 6. Audit Message



File Details

	Sno	Owner	Filename	Signature	Key	Date	Checking
□	1	priya	mydata.sql	Pri21516	b46ba975	05-10-2015	File found
□	2	nandhini	cloud.sql	nan77019	18580742	02-10-2015	File found
□	3	ramesh	dd.JPG	Ram39520	f56222ad	02-10-2015	File found
□	4	ramesh	dd.JPG	Ram39520	f0e8763a	02-10-2015	File not verified
□	5	malar	Chrysanthemum.jpg	mal7815	ac424e3c	18-08-2015	File found

Cloud Data Owner
Services provided by cloudservice.com

Figure 7. File Status

V. CONCLUSION

The security aspects of cloud computing have been thoroughly examined in prior research. This project focuses on identifying various privacy threats and reviewing the techniques available to mitigate them. While certain methods employed conventional cryptographic techniques to ensure privacy, others opted for alternative strategies. Additionally, the discussion includes methods for maintaining privacy during public auditing processes. Ultimately, it is imperative that every cloud user is assured that their data is securely stored, processed, accessed, and audited at all times.

Ensuring data freshness is crucial to safeguard against intentional misconfigurations or rollbacks, and it can facilitate the development of an authenticated file system that efficiently and transparently transitions an enterprise-class distributed file system to the cloud in a scalable manner. This authentication allows enterprise tenants to verify the freshness of the data retrieved during file system operations.

Users must retain complete control over their published data, and robust security mechanisms should consistently support all cloud applications. Achieving these objectives will bring us closer to realizing the long-cherished vision of secure cloud computing in the near future. The proposed model may serve as a foundation for establishing a secure cloud computing environment, significantly enhancing privacy preservation.

REFERENCES

[1]. Jin Li, Xiao Tan, Xiaofeng Chen, Duncan S. Wong, FatosXhafa, "OPoR: Enabling Proof of Retrievability in Cloud Computing with Resource-Constrained Devices" IEEE TRANSACTIONS ON CLOUD COMPUTING, VOL. 3, NO. 2, APRIL-JUNE 2015.
[2]. Chang E.C and Xu, Forrester,.J, "Remote integrity check with dishonest storage server," in Proceedings of ESORICS 2008, volume 5283of LNCS. Springer-Verlag, 2008, pp. 223–237.
[3]. Tao Jiang, Xiaofeng Chen, and Jianfeng Ma, "Public Integrity Auditing for Shared Dynamic Cloud Data with Group User Revocation"IEEETrans. Parallel Distrib. Syst., vol. PP, no. 99, 2015.
[4]. Zhu.Y, Wang.H, Hu .Z, Ahn. G.J., Hu.H, and Yau. S. S., "Dynamic audit services for integrity verification of outsourced storages in clouds," in SAC, 2011, pp. 1550–1557.
[5]. Li.J and Kim.K, "Hidden attribute-based signatures without anonymity revocation" Information Sciences, vol. 180, no. 9, pp. 1681–1689, 2010.
[6]. Zhu. Y, Hu.H, Ahn G.J, and Yu.M., "Cooperative provable data possession for integrity verification in multicloud storage," IEEETrans. Parallel Distrib. Syst., vol. 23, no. 12, pp. 2231–2244, 2012.
[7]. Jiawei Yuan and Shucheng Yu "Public Integrity Auditing for Dynamic Data Sharing With Multiuser Modification" IEEETrans. Information Forensic and security. Syst., vol. 10, no.8, 2015.
[8]. Wang. C, Wang. Q, Ren. K, and Lou. W, "Privacy-preserving public auditing for data storage security in cloud computing," in INFOCOM, 2010, pp. 525–533.
[9]. Shah.M.A, Swaminathan.R, and Baker.M, "Privacy-preserving audit and extraction of digital contents," Cryptology ePrint Archive, Report 2008/186, 2008.
[10]. Zheng .Q and Xu. S, "Fair and dynamic proofs of retrievability," in CODASPY, 2011, pp. 237–248.
[11]. J. Li, X. Chen, J. Li, C. Jia, J. Ma, and W. Lou, "Fine-grained access control system based on attribute-based encryption," ESORICS, 2013.
[12]. J. Li and K. Kim, "Hidden attribute-based signatures without anonymity revocation," Information Sciences, vol. 180, no. 9, pp. 1681– 1689, 2010.
[13]. J. Li, C. Jia, J. Li, and X. Chen, "Outsourcing encryption of attribute-based encryption with mapreduce," ICICS, 2012.
[14]. X. Chen, J. Li, J. Ma, Q. Tang, and W. Lou, "New algorithms of outsourcing modular exponentiations," ESORICS, pp. 541–556, 2012.
[15]. Y. Zhu, H. Hu, G.-J. Ahn, and M. Yu, "Cooperative provable data possession for integrity verification in multicloud storage," IEEETrans. Parallel Distrib. Syst., vol. 23, no. 12, pp. 2231– 2244, 2012.
[16]. H. Xiong, X. Zhang, D. Yao, X. Wu, and Y. Wen, "Towards end-to-end secure content storage and delivery with public cloud," in CODASPY, 2012, pp. 257–266.
[17]. C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," Cryptology ePrint Archive, Report 2008/432, 2008.