IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

AN ANALYSIS OF THE ROBUST KEY REVEAL MECHANISM USED IN THE PUBLIC AUDIT MODEL FOR SECURE CLOUD STORAGE.

Akheel Mohammed
Department of Computer Science & Engineering
Dr.VRK Women's of Engineering and technology, Hyderabad, Telangana, India

Ayesha
Department of Computer Science & Engineering
Dr.VRK Women's College of Engineering and technology, Hyderabad, Telangana, India.

ABSTRACT

Cloud storage allows customers to securely store their data remotely and access it using high-quality apps on demand. Data outsourcing is a practice that alleviates users from the burdens associated with data storage and maintenance. By entrusting their large-scale data to cloud platforms, users are relieved of the responsibility of ensuring data integrity, which poses significant challenges. The implementation of public audit for cloud data storage security has significant importance. Individuals have the option to request an external assessment conducted by a third party in order to verify the authenticity and reliability of their externally derived data. Nevertheless, under this particular framework, it is possible for the untrustworthy cloud to potentially generate legitimate authenticators at a later period than the point at which the secret key of the Data Client is revealed, provided that it has acquired such key. This article presents a proposed prototype called Robust Key Revelation for public auditing in secure cloud storage. The objective is to ensure the preservation of cloud storage auditing security both before to and subsequent to the key revelation process. The concept and security model of this novel cloud storage auditing mechanism were formalized, and a specific scheme was devised. The suggested prototype ensures that the security of cloud storage auditing is unaffected by the key reveal in one time period, hence maintaining data integrity across other time periods. The security proof and experimental results provide evidence that the suggested prototype effectively achieves both intended security and efficiency within specified time periods. Keywords: Secure Cloud Storage, Robust Key Revelation, Security, Data Client, and Public Auditing.

I. INTRODUCTION

In the area of cloud computing, cloud storage auditing plays a pivotal role since the fundamental goal of cloud computing is to deliver dependable and quickly accessible data services to clients by making significant use of virtualized infrastructures [1, 2]. [Cloud storage auditing] plays a key part in cloud computing because of this primary goal. The cost-effectiveness and higher performance provided by cloud storage may be a contributing

factor in the growing popularity of using professional cloud service providers (CSPs) for the storage of data by both individuals and businesses. Recent years have seen a substantial acceleration in the development of cloud storage and the accompanying methodologies, which has been considerably aided by this trend. Despite this, there are a number of safety issues with cloud storage, which is emerging as an innovative and creative technology [3]. When it comes to the decision of whether a cloud storage system and its provider effectively satisfy the legal standards that have been outlined by customers in terms of data security, one of the key issues that should be taken into account is [4].

The security of the cloud environment is ensured by the installation of a complete set of protocols, which includes several systems for user authentication, access control, and the preservation of thorough user information logs. This ensures that only authorized users may access the cloud environment. In large-scale cloud settings, the installation of access control systems is a considerable difficulty [5]. When a data customer makes use of storage in the cloud, the space that is made available to them is in the form of a virtual space. Users' access to data that does not belong to them will be limited by the virtual environment, which will enforce these limits. Traditional access techniques often make use of a variety of protection mechanisms to ensure the integrity of the device that is in charge of storing the memory. In contrast, when data is kept in a cloud environment, it is located at a significant distance from the control of the data client. This makes it more difficult for the data client to access the data. The field of cloud infrastructure is distinguished by the absence of individual user control at the level of the user. A cloud computing environment is notable for several additional qualities in addition to its inherent safety, including parallelism, user-friendliness, integrity, and secrecy. These characteristics may be found in a cloud.

The auditing procedures that are associated with cloud storage have attracted a large amount of interest and have been the focus of an in-depth investigation [6]. The processes that are covered in this research concentrate on several areas of analysis, with the major attention being placed on the accomplishment of high data transfer rates and efficient algorithm implementation [7]. As a result, the Homomorphism Linear Authenticator (HLA) process, which provides square-less verification, is the subject of investigation in this work. The goal is to reduce the algorithmic and communication overheads associated with auditing protocols. With the use of this technique, the auditor may ensure that the data stored in the clouds are accurate and trustworthy without having to extract all of the data.

In order to deal with this process, a number of different auditing solutions for cloud storage have been proposed [8]. Auditing a cloud storage service should always include steps to protect the confidentiality of user data. The usage of a third-party auditor (TPA) is implemented to aid the customer in regularly validating the integrity of the data that is kept in the cloud. This is done in order to reduce the amount of computational work that has to be performed by the client. On the other hand, it is not impossible for the Trusted Third Party (TPA) to get the user's information once the auditing procedure has been carried out on a number of separate times. The content provided by the user does not include enough information to be rewritten in an academic style. The major purpose of auditing protocols is to guarantee the highest possible level of safety for any user information that is kept in a cloud-based environment. The investigation of several methods that might enable information-related dynamic actions is another topic that has been discussed in the area of cloud storage auditing [10].

There are several possible causes for key presentations, including the following:

The most important function: The most important service is one in which the consumer takes the lead. In the event that there is any culpability, and the customer is using a programming-oriented key solution that is below standard, then key installation becomes achievable.

Internet-based security assaults are nefarious operations that target computer systems, networks, and data by using the internet as their primary vector of attack. These assaults are distinguished by the fact that they are designed with the goal of compromising the availability, confidentiality, and integrity of digital information. If a client downloads any kind of information or document, and that file happens to have a malicious application on it, then the framework might potentially get corrupted as a result of this. The capacity of programmers to quickly access any secret information is facilitated as a result of this.

The cloud platform also receives incentives by virtue of its participation in the trade process with pertinent hackers. These incentives are acquired via the platform's involvement in transactions with software developers. In this step of the process, the cloud has the power to recover erroneous information or prevent the loss of information in order to collect the data from the client and produce the authenticator. In the context of cloud storage, the management of important presentations is a significant problem, and numerous solutions have been tried to solve this issue. Managing key presentations may be difficult.

This research makes a contribution to improving the safety of cloud storage auditing procedures over all time periods, not only the robust key revelation time period. In this all-encompassing setup, the Third-Party Auditor (TPA) generates an update message on each occasion by using their private key, and then sends the message on to the client. The client updates his secret key by using his private key in conjunction with the update message that he has obtained from the Trusted Third Party (TPA). By using this strategy, the hostile cloud will be unable to get the signing secret keys at unspecified time periods.

The next part of session 2 is dedicated to doing a review of the current body of research, which is also referred to as the literature review in certain circles. During the third session, we will discuss the methodology that will be used by the proposed system. The complete analysis and discussion of the experimental results that were acquired from the installation of the recommended system are the topics that will be covered in the fourth session. The conclusion is going to be discussed in the fifth section of this essay.

.

II. LITERATURE REVIEW

In their study, Wang et al. (2011) discussed the challenges associated with guaranteeing the accuracy of data storage and offered a strategy that is both efficient and secure to tackle these concerns. A secure introduction is made of a third-party auditor (TPA), who will regularly check the integrity of the data kept on a cloud server

per the user's request. Users will not experience any online burdens, and the confidentiality of their data will be maintained since it will not be directly shared with the third-party auditor. A homomorphic encryption strategy is used to encrypt the data via the utilization of Elliptic Curve Cryptography (ECC), with the intention of afterwards sharing the encrypted data with the Trusted Third Party (TPA). ECC offers effective and reliable solutions for cloud storage servers, ensuring both efficiency and security. This results in expedited computational time, hence diminishing the need for processing power, conserving storage capacity, and minimizing bandwidth use. The findings may be further expanded to assist the third-party auditor in doing different auditing responsibilities.

In their study, Deshmukh et al. (2012) created a system that utilizes a distributed method to enhance data security. The proposed structure comprises a primary server and a configuration of subordinate servers. There is no direct connection between clients and slave servers. The master server is responsible for managing the preparation of clients' appeals, while the slave server is responsible for executing the chunking procedure. The chunking procedure involves the storage of duplicate records in order to provide data backup for eventual document recovery. Clients also have the capability to execute robust and dynamic data operations. The document belonging to the client is stored on the main server in the form of tokens, while the records are divided into chunks and stored on slave servers to facilitate file recovery. The suggested approach subsequently included mechanisms to ensure the integrity and accessibility of stored data. This was achieved via the use of token creation and merging algorithms.

The study conducted by Wang et al. [13] The proposed technology enables customers to conduct inspections of cloud data storage. This approach employs the use of homomorphic tokens in conjunction with the Reed-Solomon erasure correcting coding technique. This combination ensures the provision of correctness assurance and facilitates the identification of any misbehaving servers. Furthermore, this concept was further expanded to include support for block-level dynamic operations. If individuals are lacking in enough resources and time to effectively handle data, they have the option to assign this responsibility to a Third-Party Agent (TPA). This technology enables users to securely store data at distant locations and facilitates dynamic actions, including insertion, updating, and deletion.

The authors Zheng et al. (2014) introduced a protocol for safe and sustainable storage auditing in cloud computing, which includes the capability to facilitate key upgrades for clients. In order to mitigate the significant costs associated with key updates at the local level, the responsibility of performing partial key update activities is delegated to the Trusted Third-Party Administrator (TPA). Furthermore, the clients have the ability to authenticate the legitimacy of the recently updated keys via the use of BLS signature technology. The security analysis demonstrates that the protocol under consideration is capable of providing the security attributes of correctness, verifiability, and accountability.

In their study, Jiang et al. (15) presented a novel ID-based public auditing protocol for cloud data integrity verification. This protocol leverages ID-based signature technology to enhance information authentication and the computational capabilities of the auditor. Furthermore, it has been shown that the suggested protocol exhibits security inside the random oracle model, under the assumption that the Diffie-Hellman issue is computationally challenging. In addition, the study conducts a comparative analysis of the suggested protocol in terms of security characteristics, communication efficiency, and computing cost, in relation to two alternative ID-based auditing protocols. The comparative analysis revealed that the suggested protocol effectively fulfills a greater number of security requirements while requiring a reduced computational burden.

In their study, Yang et al. (2016) introduced a public auditing strategy that addresses data confidentiality concerns by using a Third-party Auditor (TPA) for the purpose of conducting audits. The proposed approach involves the creation of a unique log referred to as "attestation," whereby the pseudonym of the user is hashed to ensure the preservation of user anonymity. This approach introduces attestation-based data access identification, which does not introduce any new risks to data confidentiality and does not impose additional online load on the user. The use of user pseudonyms serves to reinforce the responsibility of individuals for data leaking. Our method has been subjected to thorough security and performance examination, which includes comparing it with current auditing schemes.

In their study, Thokchom et al. (2017) introduced a robust auditing method designed to verify the integrity of dynamic data that is shared among a fixed number of users and stored in an untrusted cloud storage environment. The system has been developed using a ring signature scheme based on the Computational Diffie-Hellman problem. The proposed scheme facilitates the auditing of a client's data by a third-party auditor, while ensuring that the auditor remains unaware of the actual content. Additionally, the method preserves the privacy of the group member who signs the data, both from the auditor and the cloud server. The disclosure of the individual responsible for signing the data block inside the group may only be made by the authorized party, if deemed necessary.

In their study, Wu et al. (2018) focused on the preservation of identity privacy in CLCA systems. This study presents an analysis of the security models pertaining to privacy preserving CLCA (Confidentiality, Integrity, Availability) schemes, specifically focusing on uncreatability and anonymity. Additionally, an efficient CLCA scheme is provided, which has been shown to be safe within the aforementioned security models. One notable characteristic of this system is the compactness of the message tag, which is composed just of a single group element. The concept of uncreatability is grounded on the many forms of the bilinear Diffie-Hellman assumption within the framework of the random oracle model. The confidentiality of user identification is assured against third-party auditors by information-theoretic means.

In their study, Han et al. (2019) introduced a public cloud auditing system designed specifically for smart cities. This scheme is characterized by its lightweight nature and its ability to maintain anonymity, all without the need for bilinear pairings. The proposed technique is characterized by its pairing-free nature, which enables a third-party auditor to construct authentication meta sets on behalf of users. In addition, it serves to safeguard the confidentiality of data from both third-party auditors and cloud service providers. Furthermore, this novel approach may be seamlessly and organically expanded to include batch auditing within a multi-user context. Comprehensive evaluations of security and performance indicate that the suggested technique exhibits superior levels of security and efficiency when compared with current public cloud auditing schemes.

The authors Zhao et al. (2020) provide a scholarly proposal for a technique that ensures privacy preservation and unforgeability in a searchable encrypted audit log. This scheme is built upon the concept of public-key encryption with keyword search (PEKS). The generation of encrypted audit logs with access rights for users is restricted to the trusted data owner. The server with semi-honest behavior employs a searchable encryption technique to authenticate the audit logs prior to authorizing users' operation permissions and securely storing the said logs. The individual or entity responsible for the data may do a detailed conjunctive query on the archived audit logs and selectively approve just those logs that are deemed legitimate. The proposed technique demonstrates resistance against collusion, tampering, or falsification attempts carried out by both the server and the user. Detailed proposals for the practical execution of the program are presented. The proof of the scheme's rectification is shown, and an analysis is conducted on its security aspects, including privacy preservation, searchability, verifiability, and unforgeability.

III. ROBUST KEY REVELATION PUBLIC AUDITING PROTOTYPE

This study presents a contribution to the security of cloud storage auditing schemes throughout time intervals other than the robust key revelation time period. In this comprehensive structure, the Third-Party Auditor (TPA) creates an update message using their secret key on each occasion, thereafter, transmitting it to the client. The client modifies his signing secret key by using his private key and the update message received from the Trusted Third Party (TPA). This approach renders the malevolent cloud unable of acquiring the signing secret keys during periods of time when they are not exposed.

The system under consideration consists of three distinct entities, namely the cloud, the client, and the third-party auditor (TPA). Cloud computing provides storage services to clients. The client transfers client documents and their accompanying authenticators to the cloud, thereby removing this data from the client's storage space. The user has the ability to access and recover the data from the cloud at their convenience. The Third-Party Administrator (TPA) assumes a supervisory role and is responsible for overseeing two critical functions. One of the primary functions is to provide an auditing service, which involves conducting regular assessments to ensure the integrity of the documents stored in the cloud on behalf of the customer. Ultimately, the purpose of this process is to assist the customer in refreshing their confidential keys by providing them with new messages at various intervals.

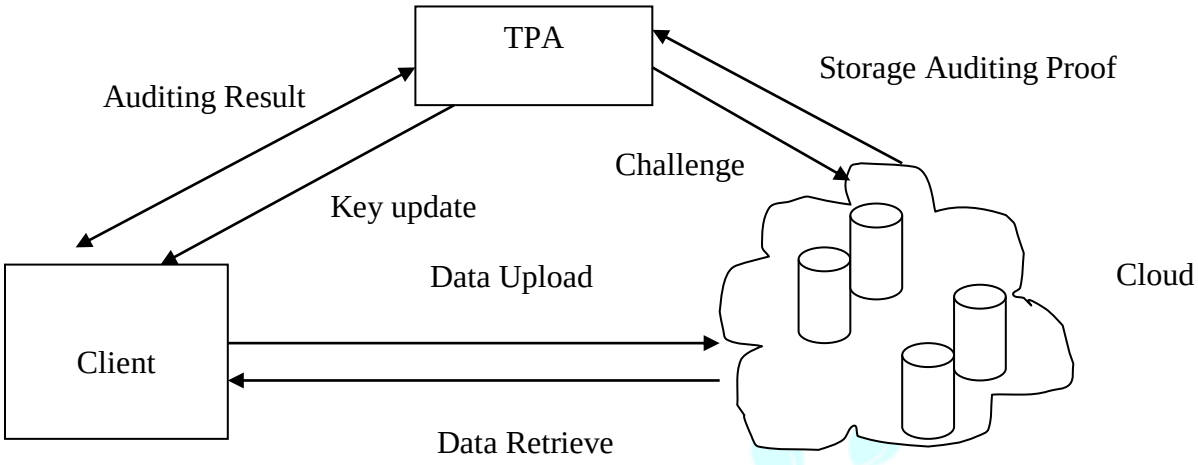


Fig.1. Proposed System Architecture

Similar to other public integrity auditing methods, the Trusted Third Party (TPA) demonstrates honesty in conducting integrity audits on behalf of cloud customers. However, the reliability of important updates in this system paradigm is not totally guaranteed. The system model is shown in Figure 1. Let us consider a scenario where document D is kept in a cloud environment and is partitioned into y chunks, denoted as xi (where i ranges from 1 to y). In contrast to the system model presented in reference [21], the temporal duration of document D inside this particular system model is not required to be predetermined. This implies that the quantity of time periods is infinite, a characteristic that aligns more closely with reality.

3.1. SECURITY ALGORITHM TYPES

A collection of five algorithms (Sy Setup, KeyUpdate, AuthGen, ProofGen, Proof Verify) is used to establish an auditing protocol that offers flexibility in key presentation.

The setup procedure, denoted as Sy Setup (1k, T) → (OPK, EK0), is a probabilistic algorithm that requires a security parameter k and the total number of eras T as input. It outputs an open key OPK and the secret key EK0 associated with the underlying users. The control of this algorithm is managed by the client.

The key update method (OPK, j, EK j) → (EK j+1) is a probabilistic process that utilizes the general key OPK, the current time frame j, and a user's secret key EK j as input to generate a new secret key EK j+1 for the subsequent time frame j + 1. The control of this algorithm is managed by the client.

The AuthGen algorithm is a probabilistic method that generates a set of authenticators _ for a given document D and time frame j. It accepts as input the public key OPK, the current time frame j, the user's secret key EK j, and the document D. The algorithm is also executed by the user.

The proof generation algorithm is a probabilistic algorithm that constructs a proof P, implying the presence of document D in the cloud. This algorithm uses the general population key OPK, an era j, a test Chal representing the challenge phase, and the arrangement of authenticators _. In this scenario, the reviewer issues the combined operation between variables j and Chal, which is then employed by cloud. The operation of this algorithm is governed by cloud computing infrastructure.

To verify the statement (OPK, j, Chal, P), we need to provide proof. ("Correct" or "Incorrect"): The evidence confirming algorithm is a deterministic algorithm that accepts inputs such as the general public's key OPK, a specific period j, a test Chal, and a proof P. It then produces an output of either "Right" or "Wrong". The control of this algorithm is executed by the client.

The aforementioned security model encompasses the notion that if a challenger is unable to guess all the missing blocks, it is unable to provide sufficient evidence during a duration during which the secret key remains undisclosed until it has all the blocks associated with a certain challenge. Each individual block of authenticators has the ability to perform a query operation whenever necessary. In all time periods save the challenged time, the attacker has the ability to query the secret keys. The objective of the opponent is to provide legitimate evidence of ownership P for the blocks presented by Chal within the time period t*. The aforementioned definition demonstrates the existence of a knowledge extractor capable of extracting the challenged data blocks under the condition that the adversary may provide a valid proof of possession P within the time frame t*.

An algorithm is a step-by-step procedure or set of rules for solving a problem or completing.

a) The letter P represents the prototype.
b) The set P consists of the elements X, O, F, K, T, Success, and Failure.
c) Let X be the set of input values.
d) Let X be a set consisting of three elements, namely X1, X2, and X3.
e) The variable X1 represents the user's login user ID.
f) The equation X2 represents the login password.
g) The variable X3 is assigned the value "File".
h) The variable K represents the collection of keys, which includes both the Secret key and the Open key.
i) The set K is defined as {(SK1, OP1), (SK2, OP3), ..., (Ski, OPi)}.
j) The set of outputs, denoted as O, refers to the collection of all possible results or outcomes in a given system or process.
k) The set O is defined as {O1, O2, O3, O4}.
l) The topic of discussion is the authentication message, denoted as O1.
m) The file labeled "O2" is encrypted.
n) Ozone (O3) as a Means of Detecting Attacks
o) The periodic key, denoted as O4, is a fundamental element in the context of periodicity.
p) The original data file is denoted as O5.
q) The variable T represents the time period required for the production of cryptographic keys.
r) Let F denote the set of functions.
s) The set F is defined as {F1, F2, F3, F4, F5}.

In the context of this discussion, the term "were." The first function, denoted as F1, pertains to the process of authentication. The output of this function, denoted as O1, is determined by the

The encryption process, denoted as F2, is represented by the equation O2 = F2(X3, K).

The equation F3 represents the concept of attack detection. The variable O3 is assigned the value of F3 multiplied by K.

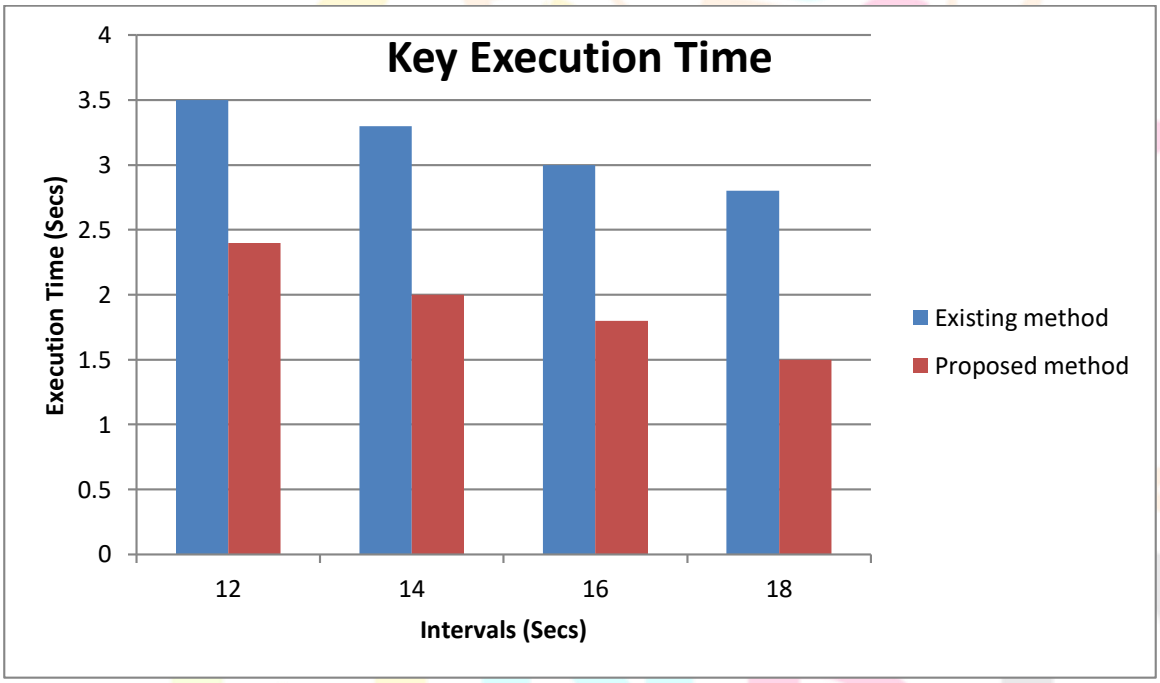
The equation F4 represents the process of periodic key generation. The output O4 is obtained by applying the function F4 to the inputs O3 and T.

The decryption process, denoted as F5, is performed using function O5, which takes as input the ciphertext (O2) and the decryption key (K).

This programming algorithm demonstrates the process of client identification and authentication, followed by the generation of an encrypted key. It also includes the detection of malicious attacks from potential attackers. Additionally, the algorithm periodically generates new keys and concludes with the decryption of the original data by the client in order to retrieve the documents.

IV. RESULT AND DISCUSSION

In this part, it is important to engage in discussions on the outcomes with data customers who are linked to the cloud service providers. The duration required for verifying the data clients' access to obtain the documents within certain time intervals using the encryption key has to be evaluated in relation to the current approach, in light of the suggested prototype. The observed communication overhead surpasses initial expectations, and the existing user population has managed to maintain the overhead within manageable limits. The shown findings illustrate the impact of implementing a monitoring prototype on the design and the performance of the system. The approach described in reference to [22] is compared with the suggested method that will be presented in the following



sections.

Fig.2. Key Execution Time compares with existing and proposed methods.

Open Stack Time refers to the comprehensive duration required to complete several stages of a process, including receiving a request, retrieving modifications from the user, validating the modifications, and then updating them inside the cloud infrastructure modules. The display of monitoring time serves to illustrate the significance of designing a security model with optimal efficiency, without compromising the performance of the cloud provider. The ping time refers to the overall duration during which a data originator establishes a connection with the monitoring prototype. Figure 2 illustrates a comparison between the execution times of the resilient key in relation to both current and suggested approaches.

CONCLUSION

This paper addresses the issue of key revelation in cloud storage auditing and presents a prototype called Robust Key Revelation for secure cloud storage. The proposed solution ensures the preservation of security in cloud storage auditing both before and after the key revelation. The description and security model of this novel cloud storage auditing approach were formalized, and a specific scheme was devised. The suggested prototype demonstrates that any significant discovery made at a particular moment does not have an impact on the overall security of cloud storage audits throughout different time periods. The suggested prototype demonstrates favorable security and efficiency with time intervals, as shown by both the rigorous security proof and the experimental results.

REFERENCE:

1. Dewan, H. and Hansdah, R.C., 2011, July. A survey of cloud storage facilities. In *2011 IEEE World Congress on Services* (pp. 224-231). IEEE.
2. Wang, C., Wang, Q., Ren, K., Cao, N. and Lou, W., 2012. Toward secure and dependable storage services in cloud computing. *IEEE transactions on Services Computing*, *5*(2), pp.220-232.
3. Ren, K., Wang, C. and Wang, Q., 2012. Security challenges for the public cloud. *IEEE Internet Computing*, *16*(1), pp.69-73.
4. Ryoo, J., Rizvi, S., Aiken, W. and Kissell, J., 2014. Cloud security auditing: challenges and emerging approaches. *IEEE Security & Privacy*, *12*(6), pp.68-74.
5. Massonet, P., Naqvi, S., Ponsard, C., Latanicki, J., Rochwerger, B. and Villari, M., 2011, May. A monitoring and audit logging architecture for data location compliance in federated cloud infrastructures. In *2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum* (p. 1510-1517). IEEE.
6. Wang, C., Ren, K., Lou, W. and Li, J., 2010. Toward publicly auditable secure cloud data storage services. *IEEE network*, *24*(4), pp.19-24.
7. Yang, K. and Jia, X., 2013. An efficient and secure dynamic auditing protocol for data storage in cloud computing. *IEEE transactions on parallel and distributed systems*, *24*(9), pp.1717-1726.
8. Wang, C., Chow, S.S., Wang, Q., Ren, K. and Lou, W., 2013. Privacy-preserving public auditing for secure cloud storage. *IEEE transactions on computers*, *62*(2), pp.362-375.
9. Varalakshmi, N. and Krishna, K.J., 2018. Secure Cloud Storage Auditing Protocol with Resisting Key-Exposure.
10. Yawanikha, T., Meyanand, R., Dhivya, T., Monica, B., Minty, R.R. and Subashini, S.R., 2016. An Efficient Cloud Storage Batch Auditing without Key Exposure Resistance using Public Verifier. In *International conference on system*.
11. Wang, C., Chow, S.S., Wang, Q., Ren, K. and Lou, W., 2013. Privacy-preserving public auditing for secure cloud storage. *IEEE transactions on computers*, *62*(2), pp.362-375.
12. Deshmukh, P.M., Gughane, A.S., Hasija, P.L. and Katpale, S.P., 2012. Maintaining file storage security in cloud computing. *International Journal of Emerging Technology and Advanced Engineering*, *2*(10), pp.2250-2459.
13. Wang, C., Wang, Q., Ren, K., Cao, N. and Lou, W., 2012. Toward secure and dependable storage services in cloud computing. *IEEE transactions on Services Computing*, *5*(2), pp.220-232.
14. Zheng, W., Liu, D., Li, X. and Sangaiah, A.K., 2018. Secure sustainable storage auditing protocol (SSSAP) with efficient key updates for cloud computing. *Sustainable Computing: Informatics and Systems*.

15. Jiang, H., Xie, M., Kang, B., Li, C. and Si, L., 2018. ID-Based Public Auditing Protocol for Cloud Storage Data Integrity Checking with Strengthened Authentication and Security. *Wuhan University Journal of Natural Sciences*, *23*(4), pp.362-368.
16. Yang, Z., Wang, W., Huang, Y. and Li, X., 2019. Privacy-Preserving Public Auditing Scheme for Data Confidentiality and Accountability in Cloud Storage. *Chinese Journal of Electronics*, *28*(1), pp.179-187.
17. Thokchom, S. and Saikia, D.K., 2018, October. Efficient scheme for dynamic Cloud data shared within a static group with privacy preserving auditing and traceability. In *Proceedings of the 2018 International Conference on Cloud Computing and Internet of Things* (pp. 25-32). ACM.
18. Wu, G., Mu, Y., Susilo, W., Guo, F. and Zhang, F., 2019. Privacy-Preserving Certificateless Cloud Auditing with Multiple Users. *Wireless Personal Communications*, pp.1-22.
19. Han, J., Li, Y. and Chen, W., 2019. A Lightweight And privacy-preserving public cloud auditing scheme without bilinear pairings in smart cities. *Computer Standards & Interfaces*, *62*, pp.84-97.
20. Zhao, W., Qiang, L., Zou, H., Zhang, A. and Li, J., 2018, June. Privacy-Preserving and Unforgeable Searchable Encrypted Audit Logs for Cloud Storage. In *2018 5th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2018 4th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)* (pp. 29-34). IEEE.
21. Yu, J., Ren, K., Wang, C. and Varadharajan, V., 2015. Enabling cloud storage auditing with key-exposure resistance. *IEEE Transactions on Information forensics and security*, *10*(6), pp.1167-1179.
22. Mohammed, A. and Vasumathi, D., 2018. Restrictive Ambiguity and Add-On Architecture Prototype for Privacy Preservation in Cloud Auditing. *Journal of Computational and Theoretical Nanoscience*, *15*(8), pp.2565-2571.