



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 4, April 2025

Auditable and Secure Cloud Data Sharing with Group Management and Access Control

T Shakila¹, E Vishnuvardhanreddy², E Sai Kiran³, Ch Bharath Kumar⁴, CH Lakshmi Narayana⁵

Assistant Professor, Department of CSE Bharath Institute of Higher Education and Research, Selayiur, Chennai,
Tamil Nadu, India

B. Tech Students, Bharath Institute of Higher Education and Research, Selayiur, Chennai, Tamil Nadu, India

ABSTRACT: Cloud storage solutions provide users with the ability to easily store and access their data from anywhere at any time. While these services have many advantages, they also limit direct oversight of your information and increase the chances of data loss. To address these concerns, researchers have developed various testing techniques aimed at protecting the security and integrity of cloud-based data. This project concentrates on the current public testing methods for general data, highlighting critical weaknesses, particularly related to everyday forgery and evidence. These weaknesses can allow cloud servers to show the existence of data even after its deletion. To mitigate these issues, we will introduce new testing procedures focused on boosting data security, ensuring effective identity and dynamic group management, and preventing similar vulnerabilities. Moreover, we will evaluate the computational and communication efficiency of the proposed methods against existing techniques.

KEYWORDS: Public auditing, privacy-preserving, shared data, dynamic group management, third-party auditor.

I. INTRODUCTION

Cloud storage solutions require robust data integrity and security systems that can be confirmed without downloading the entire dataset. These systems generally fall into two categories: private auditing and public auditing. In private auditing, an agreement is formed between the data owner and a private auditor, who conducts data audits only when requested by the owner, thus enhancing privacy. However, this approach can be resource-intensive and does not scale efficiently, particularly for larger datasets with more complex metrics. It relies on advancements in cryptography that secure both the auditing processes and the logs themselves while preventing the disclosure of sensitive information. In contrast, public auditing transfers the verification responsibility from the user to a third-party auditor (TPA), significantly reducing the computational burden on the user. TPAs utilize advanced cryptographic methods that enable data integrity checks while keeping the original data concealed, achieving a balance between efficiency and confidentiality. Consequently, public auditing tends to be more advantageous as it enhances efficiency, accommodates large datasets, and minimizes computational and communication expenses for users. Numerous sophisticated public auditing schemes have emerged in academic literature, aiming to boost security, privacy, and functionality.

Recent auditing designs incorporate privacy-preserving auditing, dynamic data auditing, shared data auditing, and identity traceability. This evolution has led to the formulation of various privacy-preserving auditing techniques that employ elements like Homomorphic Authenticating Tags (HATs), zero-knowledge proofs, and masked file content. These cryptographic methods ensure that the TPA can periodically disclose orders without accessing the actual data, thereby protecting clients from information leaks and unauthorized access. In cloud settings where data is frequently updated, dynamic data auditing is crucial. Secure strategies are established for adding, deleting, and modifying data types, employing block-level verification and authenticated structures like Merkle Hash Trees (MHTs) and skip lists to monitor changes in the data.

When multiple users need to access and edit shared files, auditing shared data becomes essential. Secure and traceable auditing mechanisms provide innovative methods for team members to collaboratively modify and verify data with minimal performance impact. Signature aggregation techniques are also employed to enhance auditing efficiency. Identity-traceable auditing schemes use group-oriented cryptographic methods, such as group signatures, to preserve user anonymity while assigning unique identifiers to each individual, which helps identify malicious activities and prevents unauthorized alterations.

Despite their effectiveness, public auditing systems can present challenges. One such concern is collusion attacks, where a malicious user or a revoked member teams up with untrustworthy Cloud Service Providers (CSPs) to create fraudulent audit proofs. This risk is mitigated through the implementation of provable revocation protocols and randomized audit challenges. Additionally, large-scale auditing can lead to considerable computational and storage demands. To address this issue, batch auditing techniques that allow for the simultaneous verification of multiple aggregate proofs using optimized cryptographic techniques can provide relief. The multi-tenant nature of cloud infrastructure—where numerous users share the same environment—also introduces security challenges. To safeguard user data from being accessed by other users in a multi-tenant setting, strong encryption and configuration policies must be enforced for proper data isolation.

II. LITERATURE REVIEW

Cloud storage has completely changed the way data is stored and accessed, offering more flexibility and lower costs. Yet transferring data to cloud service providers (CSPs) continues to create fundamental concerns related to security and privacy. Cloud auditing provides users with the ability to verify the correctness and completeness of their data without incurring the cost of downloading it in its entirety; thus, several cloud auditing techniques have been introduced in the research literature to address these issues. This report details progress in this space, showcases previous successes, delineates current shortcomings, and explains the impetus for this project's novel approach.

Cloud auditing emerged as the remote verification of data integrity. Ateniese et al. [7] presented the use of RSA-based math to prove correctness rapidly through the Provable Data Possession (PDP) scheme. Simultaneously, Juels et al. (2007) introduced the Proof of Retrievability (POR) scheme, which was based on using hidden markers to verify data presence and retrievability, but the scheme was not fast enough. Later, Shacham et al... and Mallad et al. (2008) enhanced POR performance by introducing BLS signatures, which allowed for quicker and more effective third-party verification. These influential works introduced the concepts of private (user-performed) and public (third-party-auditor-performed) auditing, with public auditing becoming more popular over time as it is more convenient and scalable.

With the rising cloud adoption also came the need for dynamic data operations with better privacy. Users wanted the ability to safely add, delete, or modify stored data. Erway et al. (2009) attempted to enhance PDP's efficiency using list structures to track changes; however, this approach remained expensive and slow. Wang et al. (2010) proposed Merkle Hash Trees (MHTs) to facilitate auditing, but they lacked the protection of individual privacy. Later upgrades, such as random masking, improved privacy risk but were challenging to realize. Zhu et al. (2011) used index hash tables (IHTs) to make updates more flexible but were not efficient. Shen et al. (2017) achieved higher performance using a combination of linked tables and arrays, and Tian and Ding (2019) relied on Dynamic Hash Tables (DHT) to provide better performance on data update speeds; however, it was still a partial advancement in reaching flexibility and privacy, as well as making progress by partially addressing challenges related to this.

Auditing shared data in collaborative cloud environments became a focus. Wang et al. and Knox [1] introduced a system that provided privacy at users' end by operating on a group signature scheme, but no efficient way of user revocation was included. In 2013, Wang et al. proposed Oruta based on ring signatures, but user revocation still required entire keys and tags to be recreated.

In 2014, their proposal involved using a proxy helper to update tokens for removed users, but they did not consider the collusion between a Compound Service Provider (CSP) and a removed user. Jiang et al. (2016) used mathematical tools to handle revocations but were limited by slow performance. Yu et al. (2017, 2018) decreased the communication overhead, but if the user is de-registered, the old keys can invalidate the security devices.

Other researchers suggested different fixes. Wu et al. (2018) proposed a rapid privacy-protecting scheme that circumvented signatures, but there was no user traceability. Chang et al. (2019) proposed a novel user removal protocol, but data leakage was still possible. Shen et al. (2019) employed editable signatures that needed secure links that could be exposed, and Xu et al. (2020) introduced key generation with a secure user data structure that lacked these extra links but had no removals or edits. These challenges demonstrate that shared auditing remains a challenging problem since the group changes frequently, and the set of security threats for the system can evolve.

Tian et al. (2019) tried to design a complete system that satisfied privacy, flexibility, and auditability while employing a DHT and using a "wait-and-see" method for user removal. They trusted that older tags only changed in monitored group members. But this approach had fundamental limitations: attackers could create tags without the secret information, and CSPs could still create proofs even after data deletion because of revealed secrets in updates. Due to these vulnerabilities, major shortcomings were exposed regarding the proposed model, which showed the requirement of a better auditing solution with more security assurances.

III. METHODOLOGY

Existing System

Ateniese et al. and Jiang et al. [2] proposed a PDP scheme for provable data possession and provided two provably secure PDP schemes based on RSA-based homomorphic authenticators. This allows public verification at a lower cost of communication and computation. At the same time, Juels et al. [3] introduced the idea and a formal security model of proof of retrievability (POR) together with a concrete properties-matched sentinel-driven POR scheme. Later, Shacham et al. improved the POR scheme and introduced a new public auditing scheme based on resilient BLS signatures in the random oracle model [4]. Cloud storage has seen many research works in recent years.

Integrated auditing enables data privacy preservation, data dynamics, and shared data. Erway et al. [10] presented the first PDP scheme based on a rank-based approach to support data dynamics, where the skip list is authenticated. Nevertheless, the proposed scheme suffers from high computational and communication overheads, which led Wang et al. [11] to implement an auditing scheme that utilises a Merkle Hash Tree (MHT), hence one of the simplest ever.

Although Wang et al. proposed recent work, Mingxuan et al. [5] presented a privacy-preserving public auditing scheme; their approach brings considerable communication and computation costs for auditing and data updating. Zhu et al. [12] introduced a different approach based on an authenticated scheme that provides data dynamics propagated to a data structure known as an index hash table. In this scheme, the communication and computation requirements have been reduced, but it still does not help with the inefficiency in lookup and updating operations. Shen et al., based on the idea of using a doubly linked list [13], pioneered a tabular-based efficient scheme for computing dependent expressions: a two-dimensional information table plus a 1-D array start location. Tian et al. recently expanded their scheme by exploiting a DHT, which is more efficient than IHT [12]. In the context of privacy, Wang et al. [5] proposed random data masking, which allows for privacy-preserving public auditing, and many data privacy prediction schemes [6]–[9] have also been investigated.

Wang et al. [14] proposed a Knox scheme, which is a lightweight scheme for jointly managed data public auditing. However, this scheme does not support user revocation. A ring signature was proposed by Oruta [15] to hide the identities of individual users; however, it has the problem that all user keys and block tags are required to be renewed to make a user void. Wang et al. [16] also proposed a scheme that is able to achieve user revocation via a proxy re-signature. The scheme employs a resigning proxy update to evade the tag created by the revoked user; however, it is vulnerable to a collusion attack between an invalid user and the cloud server.

In addition, Jiang et al. also proposed a new public auditing scheme in [17], which utilises a vector commitment [37] with the verifier-local-revocation group signature [38]. However, in the revocation phase, the cost of the calculation is relatively high because it is necessary to first determine the revocation; all that is required is to regenerate the tags generated by the revoked user. Yu et al. [18], [19] also introduced polynomial authentication tags and proxy re-signatures. Despite the advantage of alleviating the communication overhead in the verification phase, this scheme is vulnerable to collusion attacks because the revoked user holds its correct private key and can collude with the CSP.

Proposed system

Two attacks are identified: tag forgery and proof forgery. In the case of tag forgery, we demonstrate a malicious client being able to produce a tag for the modified message without knowing the secret value. In the case of proof forgery, given a challenged message m , the attacker is able to create a valid proof for the message m , despite some files stored in the cloud having been erased.

Let us elaborate further on our new public auditing scheme, which is secure against the above attacks and maintains the same features, such as privacy preservation, data dynamics, data sharing, and identity traceability. We changed the tag-generating scheme so that the malleable property was removed and enhanced the data proof generation method to ensure individual privacy. Additionally, we modified the tendency to use lazy revocation mechanisms to ensure that the CSP cannot learn secret information, and we proposed enabling an active revocation mechanism to adapt to various steady states.

We provide a formal security proof for the scheme we propose. The theorems state that without knowledge of secret values, an attacker would not be able to generate both a valid tag and its proof.

Raw messages, as well as computation and time complexity comparison results with other schemes, are also presented, along with communication costs.

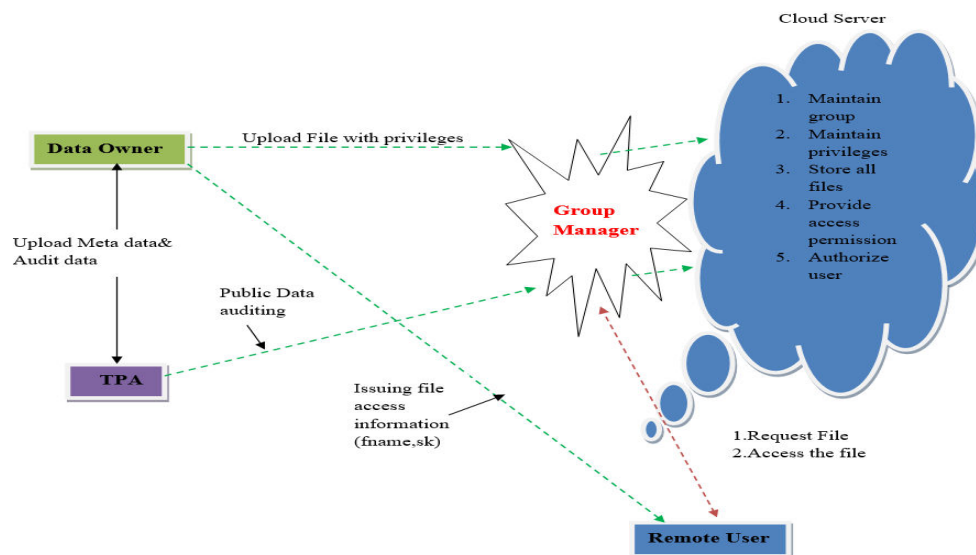


Fig 1: System Architecture

VI. IMPLEMENTATION PROCESS

1. Data Owner

User Registration and Validation: In this module, the data owner must register by providing a username, password, email, and group. Once registered, the owner can log in using a valid username and password. After ensuring the appropriate file types and formats for successful execution, the data owner uploads the data to a cloud server. The cloud's secure storage scheme for the fog node (where the data is split into n number of chunks) directly stores the data matrix; the data provider encrypts the data file, which is then stored on the cloud server via the Group Manager for security purposes. Additionally, the data owner can operate the encrypted data file.

2. Group Manager

According to this system, the Group Manager features a group-orientated design that accesses the cloud repository. The GM serves as an interface between client applications and the cloud. It combines attribute-based access control with proxy re-encryption methods for authentication and authorisation.

3. Cloud Server

A cloud server provides data storage to an end user, as well as file authorisation. The data file is stored on the cloud server with its corresponding tags, including owner, file name, secret key, MAC, and private key. It will also display the registered owner and end users on the cloud server. Based on privilege, the transfer of the data file is executed. If the appropriate privilege is present, the system sends the data to the respective user while also verifying the file name, end

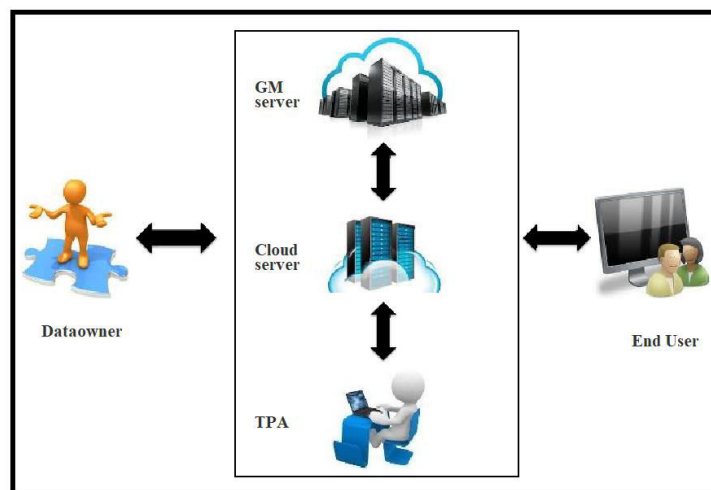
user name, and secret key. If any of these are incorrect, the data will be sent to the respective user; otherwise, the user will be flagged as an attacker.

4. Data Consumer (End User)

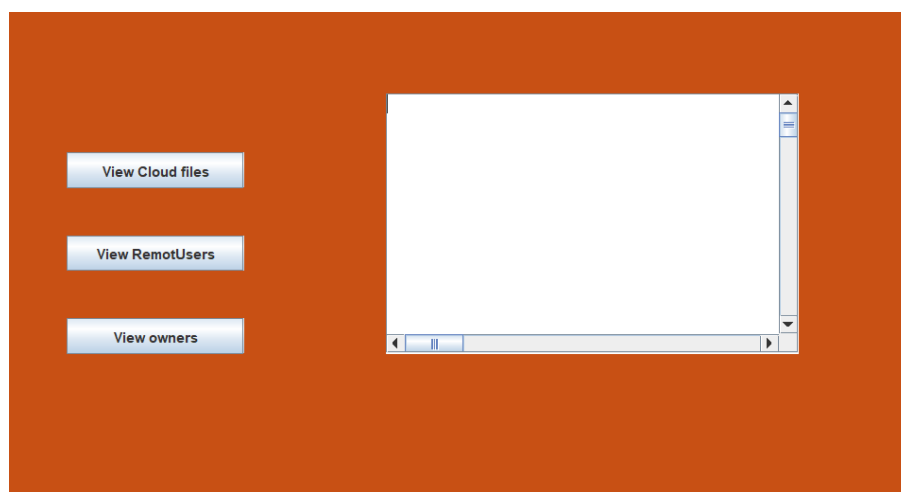
In this context, the end user (the data consumer) makes requests and retrieves the file content in response from the respective cloud servers. If the user has the file name, secret key, and access permissions (.java, .txt, .log), the final step is obtaining the file response from the cloud; those who do not meet these criteria will be treated as attackers and banned from the cloud. For example, if they wish to unblock a file in the cloud, they must execute that action after it has been blocked.

5. Attacker

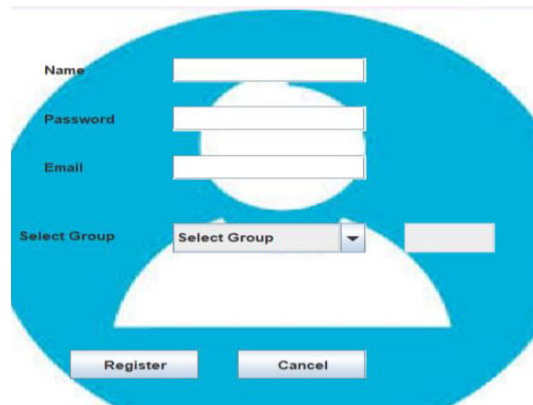
Threat Model Description: The attacker seeks to add a fraudulent key to the cloud file. An attacker can be either inside or outside the the cloud. Accordingly, we categorise attacks as external (originating from outside the cloud) or internal (originating from within the cloud boundary). External Attackers: When an attacker is from outside the cloud, these attackers are recognised as external attackers.



Work flow

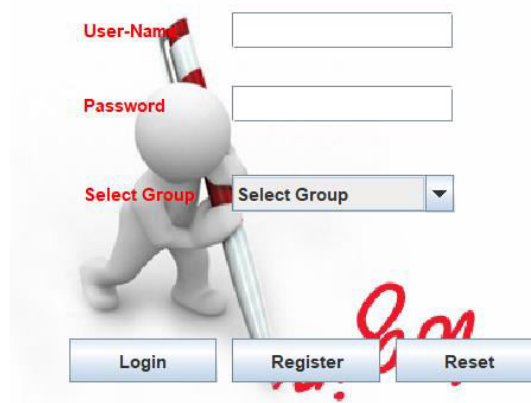


Cloud Server



A registration form with a blue background and a white circular graphic. It contains input fields for Name, Password, and Email. Below these is a 'Select Group' dropdown menu. At the bottom are 'Register' and 'Cancel' buttons.

User Register



A login form with a white background and a 3D character holding a pencil. It contains input fields for User-Name, Password, and a 'Select Group' dropdown menu. At the bottom are 'Login', 'Register', and 'Reset' buttons.

User Login

V. RESULT AND CONCLUSION

1. Security Improvements

As a result, the proposed scheme is free from the two severe vulnerabilities of Tian et al.'s model: tag forging and proof forging. It does so by proving that the tags cannot be forged without the correct secret keys and provides strong tag unforgeability in the random oracle model under the Computational Diffie-Hellman (CDH) assumption. This scheme resists manipulation (even from internal threats), unlike the approach of previous work that relies on malleable tags. Furthermore, only the Cloud Service Provider (CSP) that stores the real data can produce valid proof, ensuring that no one admits to the forgery of proof after the revocation of the user, which is a problem that has existed for a long time. The system is strengthened by the use of Extended Dynamic Hash Tables (EDHT) and a Modification Record Table (MRT), which facilitate fast identity tracing and strong privacy controls, respectively.

2. Efficiency Gains

At the level of performance, the scheme also shows competitive results in terms of computation and communication costs compared to previous schemes such as Panda [16], Yu et al. [19], and Tian et al. [25]. In terms of computation: The tag generation takes 1 group G_1 multiplication and 3 group G_1 exponentiations, plus 1 hash operation, which is equivalent to Tian et al. [25] and outperforms others by skipping excessive pairing operations.

Proof verification is not as efficient as Tian et al. [25] while scaling better (particularly in cases of user revocation). In terms of communication, the same overhead as PASCOD [25] is incurred during the challenge phase, but the response phase has a fixed cost of $7|p|$, meaning its footprint is smaller than that of Panda's [16] dataset-dependent size and is

marginally above PASCD's 6[p]. It is well suited for large dynamic environments, especially if well-tuned with not-so-powerful hardware (for example, Pentium IV with 1 GB RAM), Java, and MySQL.

3. Implementation Overview

The system is structured into five main modules: Data Owner, Group Manager (admin), Cloud Server, Data Consumer, and Attacker. It has secure tag-based access control, immediate tag expiration for revoked users, and strong prevention against attackers. These capabilities serve to enforce access controls over the data, ensure data privacy, and strengthen the system's security posture against both internal collusion and external attacks.

4. Key Advantages

Our proposed scheme surpasses Tian et al. in security and outperforms Panda and Yu et al. [19] in terms of scalability and collusion resilience. It strikes a solid balance between security, dynamic capabilities, and system efficiency. Although solid already, it can become even stronger by using lightweight cryptography methods and improving the trust model.

VI. CONCLUSION

The research paper "Privacy-Preserving Public Auditing for Shared Cloud Data with Secure Group Management" by Dongmin Kim and Kee Sung Kim (IEEE Access and Transaction on Cloud Computing, Volume 10, 22 April 2022) introduces a robust auditing scheme enhancing shared cloud data security. It fixes tag and proof forgery flaws in Tian et al.'s scheme [25], ensuring attackers need secret keys or intact data to manipulate tags or proofs, proven under the Computational Diffie-Hellman assumption. An active revocation process with a refined lazy approach prevents collusion between revoked users and the CSP, securing data integrity and privacy.

Efficiency-wise, it matches Tian et al.'s PASCD [25] in tag generation ($1\text{MulG1} + 3\text{ExpG1} + 1\text{hash1Mul}_{\{G_1\}} + 3\text{Exp}_{\{G_1\}} + 1\text{hash1MulG1} + 3\text{ExpG1} + 1\text{hash}$) and offers a constant response cost ($7|p|7|p|7|p|$), outperforming Panda [16] in scalability despite a slightly higher verification cost. The Extended Dynamic Hash Table (EDHT) and Modification

Record Table (MRT) enable dynamic data management and traceability. Implemented on a Pentium IV (3.5 GHz, 1 GB RAM) with Java and MySQL, it balances security and efficiency, ideal for cloud storage as of April 2, 2025. It surpasses prior schemes in safeguarding shared data while preserving privacy, dynamics, and group management.

VII. FUTURE WORK

1. **Improved Cloud Auditing:** Further the development of stronger auditing methods to ensure data integrity with changing security challenges.
2. **Dynamic Group Management:** Enhance support for any changes in group membership that do not violate system semantics.
3. **Performance and one of the top-scored protocols:** Make it low-cost for operation and easily verifiable.
4. **Cross-Cloud Integration:** Generalize the scheme for use with standards like OAuth to facilitate integration across a range of cloud services.
5. **Integrating AI-Powered Threat Detection:** Leverage systems such as Deep Log to proactively identify and respond to anomalies or threats specifically through logs of the system.

REFERENCES

- [1] A. Fu, S. Yu, Y. Zhang, H. Wang, and C. Huang, "A new privacy-aware public auditing scheme for cloud data sharing with group users", IEEE Trans. Big Data, vol. 8, no. 1, pp. 14_24, Feb. 2023.
- [2] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing", in Proc. IEEE INFOCOM, Mar. 2023.
- [3] DONGMIN KIM AND KEE SUNG KIM, "Privacy-Preserving Public Auditing for Shared Cloud Data with Secure Group Management", IEEE Access, 22. April. 2022.
- [4] Available on[Online]: Cloud Storage-Global Market Trajectory and Analytics.(Apr. 2021). <https://www.researchandmarkets.com/reports/5140992/cloud-storage-global-market-trajectory>.

- [5] Y. Zhang, C. Chen, D. Zheng, R. Guo, and S. Xu, "Shared dynamic data audit supporting anonymous user revocation in cloud storage", IEEE 2021.
- [6] G. Wu, Y. Mu, W. Susilo, F. Guo, and F. Zhang, Int. J. Inf. Secur., "Threshold Privacy preserving cloud auditing with multiple uploaders", Jun. 2020.
- [7] H. Tian, F. Nan, H. Jiang, C.-C. Chang, J. Ning, and Y. Huang, "Public auditing for shared cloud data with efficient and secure group management" Jan. 2019.
- [8] W. Shen, J. Qin, J. Yu, R. Hao, and J. Hu, "Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage", IEEE Trans. Inf. Forensics Security, vol. 14, no. 2, pp. 331_346, Feb. 2018.
- [9] G. Yang, J. Yu, W. Shen, Q. Su, Z. Fu, and R. Hao, "Enabling public auditing for shared data in cloud storage supporting identity privacy and traceability," J. Syst. Softw., vol. 113, pp. 130_139, Mar. 2017.
- [10] T. Jiang, X. Chen, and J. Ma, "Public integrity auditing for shared dynamic cloud data with group user revocation," IEEE Trans. Comput., vol. 65, no. 8, pp. 2363_2373, Aug. 2017.
- [11] J. Shen, J. Shen, X. Chen, X. Huang, and W. Susilo, "An efficient public auditing protocol with novel dynamic structure for cloud data", IEEE Trans. Inf. Forensics Security, vol. 12, no. 10, pp. 2402_2415, Oct. 2017.
- [12] T. Jiang, X. Chen, and J. Ma, IEEE Trans. Comput., "Public integrity auditing for shared dynamic cloud data with group user revocation," vol. 65, no. 8, pp. 2363_2373, Aug. 2016.
- [13] K. He, C. Huang, K. Yang, and J. Shi, "Identity-preserving public auditing for shared cloud data," in Proc. IEEE 23rd Int. Symp. Quality Service (IWQoS), Jun. 2016, pp. 159_164.
- [14] J. Yuan and S. Yu, "Public integrity auditing for dynamic data sharing with multiuser modification," IEEE Trans. Inf. Forensics Security, vol. 10, no. 8, pp. 1717_1726, Aug. 2015.
- [15] B. Wang, B. Li, and H. Li, "Panda: Public auditing for shared data with efficient user revocation in the cloud," IEEE Trans. Serv. Comput., vol. 8, no. 1, pp. 92_106, Jan./Feb. 2015.
- [16] J. Yuan and S. Yu, "Efficient public integrity checking for cloud data sharing with multi-user modification," in Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM), Apr. 2015, pp. 2121_2129.
- [17] B. Wang, B. Li, and H. Li, "Oruta: Privacy-preserving public auditing for shared data in the cloud," IEEE Trans. Cloud Comput., vol. 2, no. 1, pp. 43_56, Jan./Mar. 2014.
- [18] Y. Zhu, G.-J. Ahn, H. Hu, S. S. Yau, H. G. An, and C.-J. Hu, "Dynamic audit services for outsourced storages in clouds," IEEE Trans. Services Comput., vol. 6, no. 2, pp. 227_238, Apr./Jun. 2014.
- [19] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 9, pp. 1717_1726, Sep. 2014.
- [20] B. Wang, B. Li, and H. Li, "Knox: Privacy-preserving auditing for shared data with large groups in the cloud," in Proc. 10th Interfaces Conf. Appl. Crypto. Netw. Secur., 2013, pp. 507_525.
- [21] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 5, pp. 847_859, May 2012.
- [22] Z. Hao, S. Zhong, and N. Yu, "A privacy-preserving remote data integrity checking protocol with data dynamics and public verifiability," IEEE Trans. Knowl. Data Eng., vol. 23, no. 9, pp. 1432_1437, Sep. 2011.
- [23] C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in Proc. 16th ACM Conf. Comput. Commun. Secur. (CCS), New York, NY, USA, 2011, pp. 213_222.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details