A Verifiable Deletion Protocol for Enhancing Constraints on Public Clouds

Mr.B.Amarnath Reddy M.Tech,
Assistant Professor
Department of Master of Computer Applications Qis college of engineering and technology Autonomous AP,
Ongole, India

Chittaru Yuvarani
Student,
Department of Master of Computer Applications Qis college of engineering and technology Autonomous Ap Ongole, India
chittaruy@gmail.com

***Abstract*— Public cloud environments facilitate widespread data sharing and multi-user collaboration but simultaneously introduce critical challenges in ensuring secure and verifiable data deletion. Users lack assurance that their deletion requests are executed honestly, as cloud providers may retain deleted data in hidden backups for unauthorized use. To address this issue, a Verifiable Deletion Protocol (VDUP) is introduced, enhancing data control by decoupling deletion requests from credential responses using uncertainty requests and uncertainty roles. This mechanism ensures in-distinguishability between pre- and post-deletion verification, preventing cloud servers from identifying the initiator or linking requests with credentials. The protocol formally defines security properties and demonstrates resilience against backup attacks through concrete instantiations and security proofs. The system introduces a three-step process: Anonymous Check 1 for pre-deletion data fingerprinting, deletion request submission, and Anonymous Check 2 for post-deletion validation. Furthermore, an extension integrates AES encryption for stored data and implements access control policies allowing users to mark files as private or public. Experimental evaluations indicate reduced credential generation overhead compared to existing methods, while also detecting unauthorized retention through signature comparison between check phases, effectively enhancing user trust in cloud-based data deletion.**

Keywords*— *Cloud security, public cloud, verifiable behavior, verifiable deletion.

I. Introduction

Cloud storage has emerged as a cornerstone of modern computing infrastructure, enabling scalable, on-demand access to data across a wide range of applications. Its significance spans critical sectors such as healthcare, finance, government, and education, where vast quantities of sensitive data are routinely stored, retrieved, and exchanged [1]. This paradigm shift from local to cloud-based data management has brought with it compelling benefits in terms of cost reduction, flexibility, collaboration, and business continuity [2][3]. However, these advantages are tempered by persistent concerns surrounding data security, particularly the assurance of complete and irreversible deletion when data is no longer needed.

In public cloud environments, users typically relinquish direct control over their data upon uploading it, placing implicit trust in cloud service providers (CSPs) to manage, retain, and eventually delete information responsibly [4]. Unfortunately, this trust model is inherently flawed. CSPs, considered semi-trusted entities, may fail to execute deletion instructions with full fidelity, either due to technical limitations, internal policies, or even malicious intent [5]. Data marked for deletion may still be retained for system backups, audit logs, machine learning analytics, or commercial purposes—often without explicit user consent. High-profile incidents such as the Facebook-Cambridge Analytica data scandal underscore the tangible risks of unauthorized data retention and highlight the absence of meaningful enforcement mechanisms [6].

The critical challenge lies in the lack of transparency and accountability. Users currently have no concrete way to verify whether their deleted data has been thoroughly and permanently removed from all storage tiers, including replicas and distributed backups [7]. This creates serious implications not only for individual privacy and trust but also for organizational compliance with data protection regulations like the General Data Protection Regulation (GDPR), which mandates the “right to be forgotten” and emphasizes the importance of data minimization [8]. The inability to verify deletion undermines regulatory compliance, introduces legal risks, and may result in reputational damage for entities relying on cloud storage.

Thus, the concept of verifiable deletion is gaining importance, focusing on developing mechanisms that provide cryptographic assurance or third-party auditability to confirm that data has been irrecoverably destroyed. Addressing this concern is essential for fostering trust in cloud infrastructure, ensuring user autonomy over digital assets, and aligning with global data governance standards.

II. Related Work

Ozdemir et al. [6] presented an efficient approach for scaling verifiable computation using set accumulators. Their work focuses on the use of cryptographic accumulators to support verifiable computation systems, making them more scalable and practical for real-world deployment. They highlighted that set accumulators can efficiently verify computations on dynamic data sets without requiring the entire data to be present during verification, a key step toward realizing scalable verification frameworks in cloud environments.

Benaloh and de Mare [7] introduced the concept of one-way accumulators as a decentralized alternative to digital signatures. Their pioneering work laid the foundation for cryptographic constructs that allow the secure verification of data membership in a set, which is critical for applications requiring integrity assurances without central trust authorities. This approach is especially useful in the context of verifying whether a data item was properly deleted or excluded from a dataset without revealing the entire data contents.

Papamanthou et al. [8] proposed optimal verification techniques for operations on dynamic sets. Their method ensures that any update or query on a dataset can be verified with minimal overhead, providing both soundness and efficiency. This approach supports dynamic environments such as cloud storage systems, where data is frequently

modified and the correctness of such operations must be assured through verifiable protocols.

Campanelli et al. [9] presented incrementally aggregatable vector commitments (IAVCs) and demonstrated their application to decentralized storage. The innovation in their work lies in combining compact cryptographic commitments with efficient update mechanisms, allowing for secure and verifiable outsourcing of storage with minimal client-side computation. Their solution is highly relevant for public verifiability in cloud systems, ensuring that users can trust the integrity of data stored remotely.

Peikert et al. [10] constructed vector and functional commitments from lattice-based cryptography. These commitments are secure against quantum attacks and allow efficient proof generation for complex queries over datasets. Their model supports strong security guarantees and is compatible with post-quantum cryptographic standards, aligning with long-term data protection requirements in cloud infrastructure.

Wee and Wu [11] further advanced the field by developing succinct vector, polynomial, and functional commitments from lattice assumptions. Their work improves the proof size and verification time significantly, which is beneficial for low-latency applications. These contributions are vital for verifiable storage and computation protocols, especially when clients must validate server-side operations with minimal bandwidth and delay.

Zhang et al. [12] introduced transparent polynomial delegation schemes that are used in zero-knowledge proof systems. Their construction removes the need for a trusted setup and enhances transparency and auditability. This characteristic makes their scheme well-suited for applications in secure cloud storage and blockchain-based verification mechanisms.

Chen et al. [13] proposed publicly verifiable databases that support all efficient updating operations. Their system allows any third party to audit the database without needing access to the internal data, thus enhancing transparency. This capability addresses the pressing issue of trust in outsourced data management, especially for regulatory compliance and tamper-proof recordkeeping.

Wang et al. [14] developed a tag-based verifiable delegated set intersection protocol for outsourced private datasets. Their model ensures that intersecting datasets stored in different cloud domains can be verified securely without disclosing sensitive information, providing practical utility for collaborative analytics while preserving privacy.

Chen et al. [15] introduced proof-carrying data protocols built from arithmetized random oracles. This innovative mechanism supports dynamic computation verification, where clients can check the integrity of outsourced computations alongside the correctness of the data. The model ensures robust protection against dishonest behavior by service providers.

Yang et al. [16] proposed an efficient scheme for verifiable databases of unbounded size using authenticated matrix commitments. Their work allows secure queries over massive datasets without incurring high computational or communication costs. This advancement is especially significant for cloud storage systems handling high-volume dynamic data while requiring strong security and verifiability guarantees.

III. Materials And Methods

The proposed system introduces a secure and verifiable deletion framework, termed Verifiable Deletion Protocol (VDUP), to ensure data deletion integrity in public cloud environments. The core idea is to decouple deletion requests from user credentials using uncertainty-based mechanisms such as uncertainty requests and uncertainty roles, thereby preventing the cloud from associating a specific deletion action with a particular user. The protocol operates in three main stages: (1) Anonymous Check 1, where users collect pre-deletion file signatures; (2) Deletion Request, where selected files are deleted from the cloud; and (3) Anonymous Check 2, where users re-verify file signatures to detect inconsistencies. Additionally, AES encryption is applied to uploaded files to secure data from unauthorized access, even in case of breaches. The system also introduces access control policies, enabling users to classify files as public or private. These enhancements collectively provide confidentiality, integrity, and verifiability of deletion actions in cloud storage.

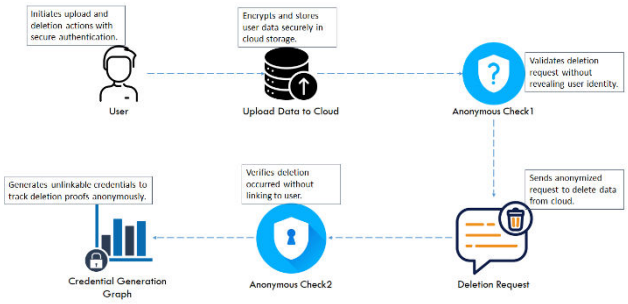


Fig.1. System Arcitecture

The system architecture ensures verifiable deletion in cloud environments while preserving user anonymity. The user securely uploads encrypted data to the cloud. When a deletion is initiated, Anonymous Check1 validates the request without exposing identity. A Deletion Request is then sent anonymously. Anonymous Check2 confirms data deletion without linking to the user. The Credential Generation Graph supports this by creating unlinkable credentials to trace deletion proofs securely. This design guarantees privacy-preserving, authenticated, and verifiable deletion across the cloud.

A) Modules

User Signup: Allows new users or data owners to register by entering their credentials and personal details. This module stores user information securely in the database, enabling account creation and access to system features like data upload, deletion, and verification functionalities after authentication.

User Signin: Enables registered users to log in using valid credentials. Upon successful authentication, the user gains access to the system's dashboard, where they can manage data files, perform anonymous checks, and submit deletion requests while maintaining secure user access.

Upload Data to Cloud: Allows users to upload files to the cloud with added security. Files are encrypted using AES and

associated with verification hashes. Access control can be applied, marking files as public or private to restrict visibility among other users.

Anonymous Check1: Performs a pre-deletion verification by fetching file details and signatures anonymously from the cloud. These results are stored locally to serve as a reference for verifying the effectiveness of the future deletion request, ensuring unbiased initial data capture.

Deletion Request: Allows users to select and delete specific files from the cloud. It triggers the deletion process on the cloud server without revealing the requester's identity, ensuring privacy [17] and preparing the system for post-deletion verification.

Anonymous Check2: Performs a second anonymous verification after deletion. It compares current file signatures with those from Anonymous Check1. If discrepancies are found, deletion is confirmed; identical signatures indicate possible backup retention by the cloud.

Credential Generation Graph: Generates and displays a comparative graph showing computational overhead time for existing systems versus the proposed protocol. It visualizes the efficiency of the VDUP protocol in reducing credential generation time and improving system performance.

Logout: Terminates the current session securely, ensuring that no unauthorized access occurs after the user leaves. It clears all session data and redirects users to the login page, maintaining overall system security and privacy.

B) Methods/Technologies

AES Encryption: AES (Advanced Encryption Standard) is a symmetric encryption algorithm used to secure files before uploading them to the cloud. It ensures that even if cloud [18] storage is compromised, the encrypted files remain unreadable to unauthorized users. This technique strengthens data confidentiality and prevents misuse or theft of sensitive user information stored on cloud servers.

Hashing (File Verification Hashes): Hashing generates unique digital signatures for each uploaded file using cryptographic hash functions. These fixed-size hashes help verify the integrity and authenticity of files during both anonymous checks. Any change in file content results in a different hash, enabling users to detect unauthorized modifications, undeleted files, or cloud backups after a deletion request.

Anonymous Verification (Uncertainty Requests): Anonymous verification involves sending uncertainty-based data requests to the cloud, where the identity of the requester is concealed. This is performed both before and after a deletion operation to ensure unbiased and unlinkable verification. The cloud is unable to determine which request is tied to deletion, enhancing privacy and trust in the verification process [19].

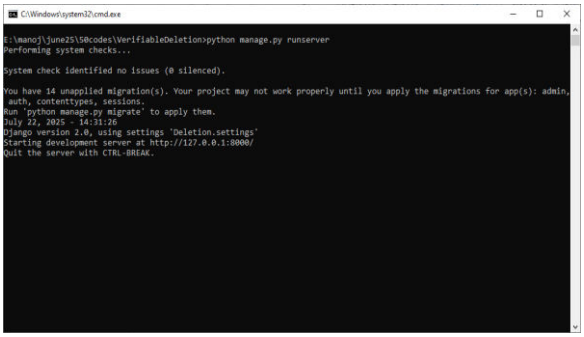
Credential Decoupling Technique: Credential decoupling separates user identity and credential information from deletion actions using uncertainty roles. This technique prevents the cloud from linking deletion requests to any specific user or file. It enhances resistance against targeted data retention and identity-based tracking by the cloud provider, thereby improving data privacy and deletion authenticity.

Comparison-Based Signature Validation: This technique involves comparing file signatures obtained during Anonymous Check1 and Check2. If the hashes differ, the file is successfully deleted. If they match, it indicates the cloud retained a backup. It allows users to verify deletion effectively and detect cloud misbehavior by validating whether the file still exists despite a deletion request [20].

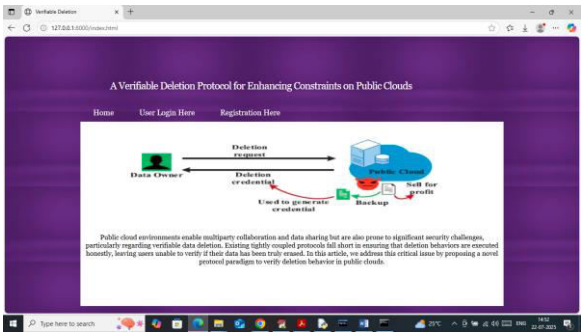
IV. Experimental Results

To run project install python 3.7.2 and then install all packages given in requirements.txt file. Install MYSQL database and then copy content from'database.txt' file and paste in MYSQL console to create database.

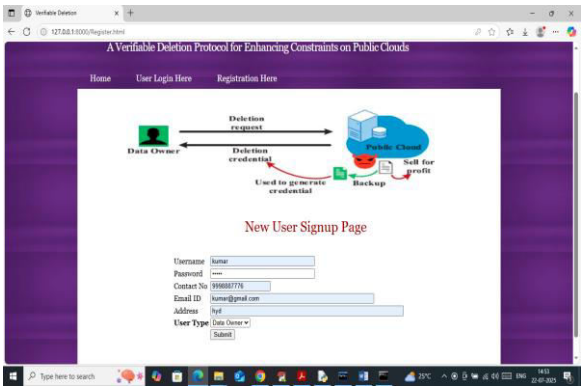
Now double click on 'run.bat' file to start cloud server and then will get below page



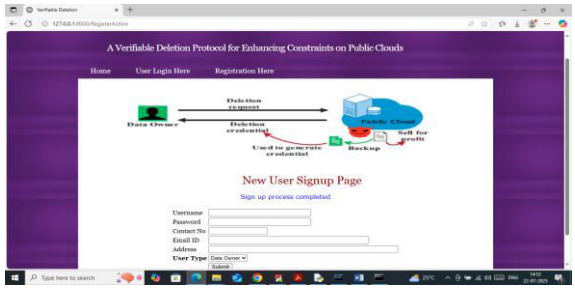
In above screen cloud server started and now open browser and enter URL As http://127.0.0.1:8000/index.html and then press enter key to get below page



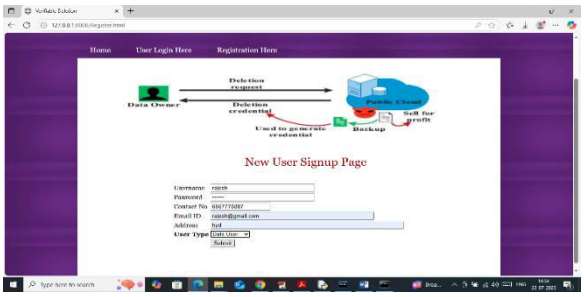
In above screen click on 'Registration Here' link to get below page



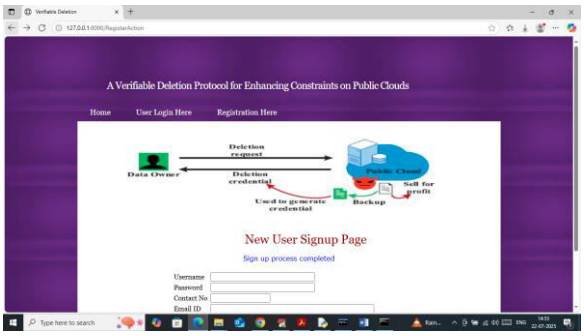
In above screen data owner is getting sign up and then press button to get below page



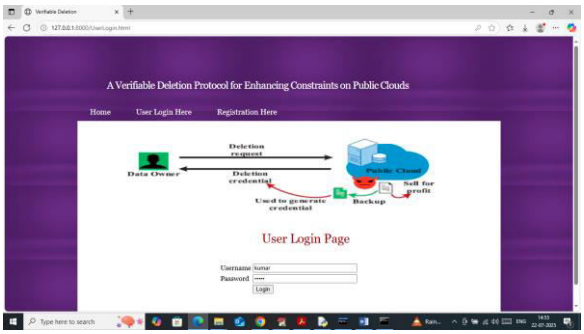
In above screen data owner sign up completed and similarly you can sign up data user also.



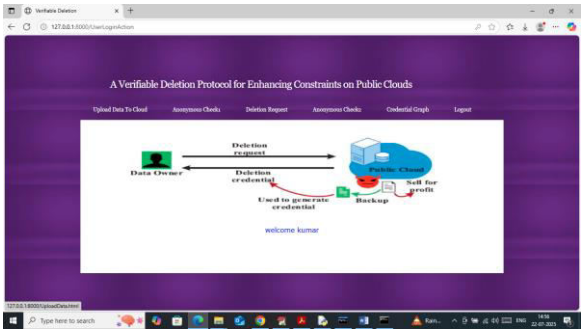
In above screen data user is getting sign up and then press button to get below page



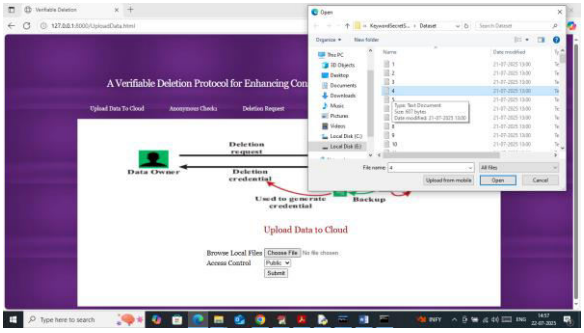
In above screen data user sign up completed and now click on 'User Login' link to get below page



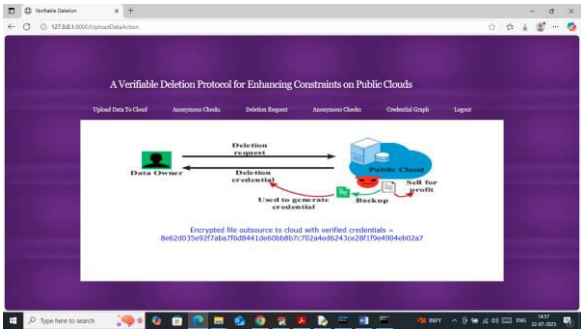
In above screen user is login and after login will get below page



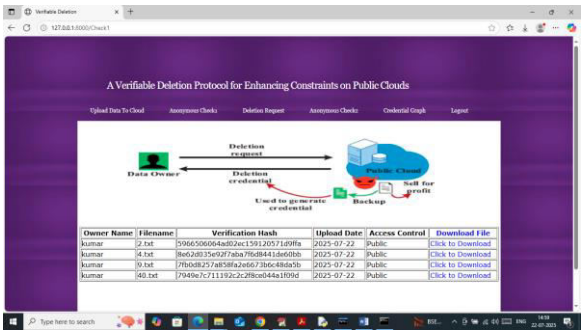
In above screen after login click on 'Upload Data to Cloud' link to get below page



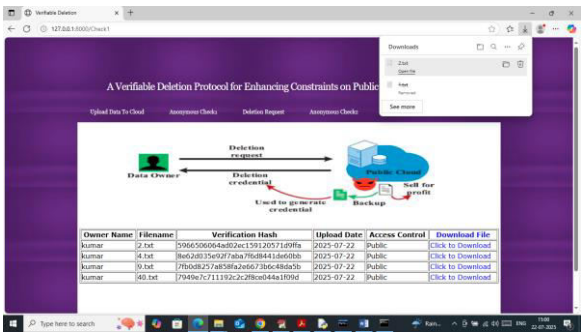
In above screen selecting and uploading file and then choose access control and then press button to uploaded encrypted data to cloud and then will get below page



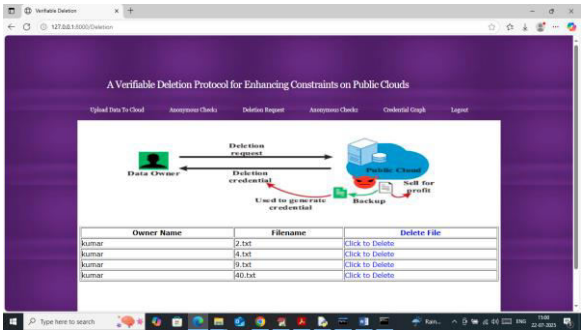
In above screen file successfully uploaded to cloud and can see generated file verification hashes. Similarly you can upload any number of files and now click on 'Anonymous Check1' link to get all file details from cloud and then make a local copy of verification.



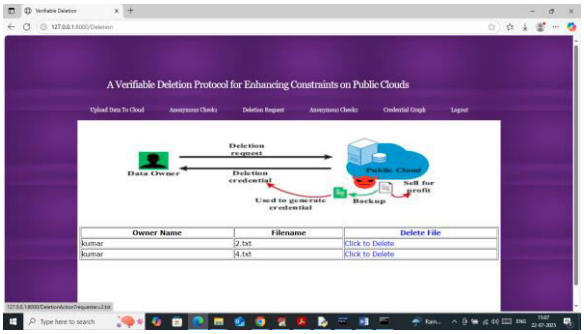
In above screen 'Anonymous Check1' local copy generated with above list of files and now click on 'Download' blue link to download file and then will get below page



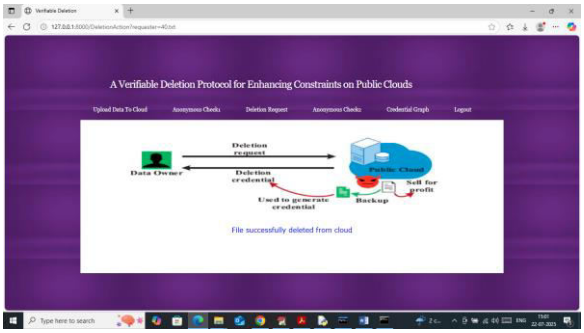
In above screen in browser status bar can see file is downloaded and now click on 'Delete Request' link to get below page



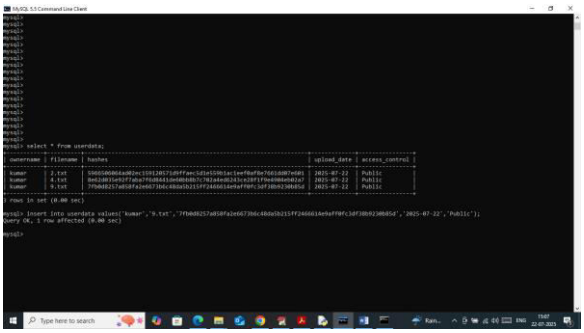
In above screen user can view list of files from cloud and can click on 'Delete' link to send delete request to cloud and then will get below page



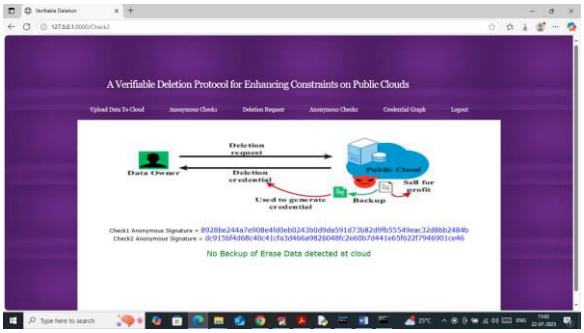
In above screen can see 9.txt file deleted and in below screen we are making a backup copy of same file in MYSQL cloud database



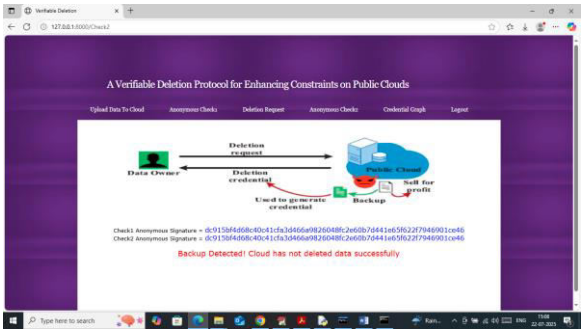
In above screen file successfully deleted from cloud and now logout and login as another user and then click on 'Anonymous Check2' link to verify file deletion and then will get below page



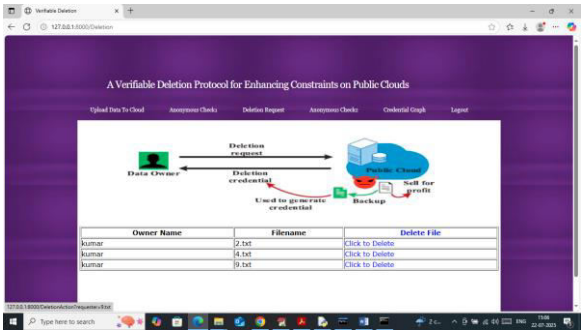
In above screen we are manually making backup copy of 9.txt file and then click on 'Anonymous Check2' link to get below page



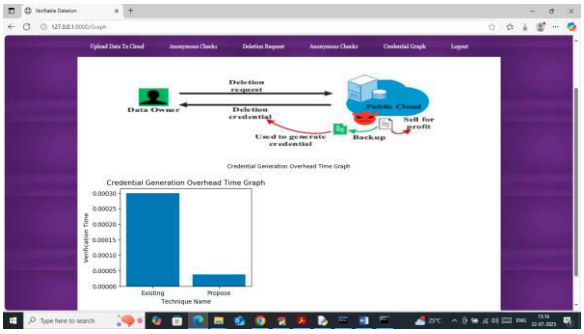
In above screen can see check 1 and check 2 verification signature and can see both signatures are different after deleting file so verifying protocol saying 'Now Backup of Erase data detected as cloud'. Now we can make a copy of deleted file in MYSQL database to simulate of process of cloud data back.



In above screen can see both check1 and check2 signatures contains same values so no changed detected at cloud which means 'cloud' has taken a backup and then propose VDUP protocol alerting user as "Backup detected" in red colour text. Now click on 'Credential Graph' link to get below page



In above screen I am deleting 9.txt file file to get below page



In above graph showing existing and propose credential generation overhead computation time comparison graph where x-axis represents algorithms type and y-axis represents overhead and in both techniques 'Propose VDUP' took less overhead or computation time.

V. Conclusion

The implemented Verifiable Deletion Protocol (VDUP) successfully ensures that data deletion requests in public cloud environments can be independently verified by users, without relying on trust in the cloud service provider. By introducing uncertainty roles and requests, the system breaks the direct link between deletion commands and user credentials, making it significantly harder for the cloud to forge or manipulate deletion evidence. This mechanism enhances transparency and accountability by allowing users to detect unauthorized data retention, even when the cloud attempts to hide such behavior through backup copies.

Integration of AES encryption provides an additional layer of security, ensuring that even if data is compromised or backed up without permission, it remains inaccessible without proper decryption keys. Furthermore, the inclusion of access control mechanisms empowers data owners to distinguish between public and private files, enforcing user-specific visibility and access constraints.

Overall, the work delivers a robust solution that verifies deletion behavior, secures data through encryption, prevents backup misuse, and establishes strict access boundaries—offering a comprehensive security framework for trustworthy cloud data management.

In the future, the protocol can be extended to support real-time alerts for unauthorized access or backup attempts using AI-driven anomaly detection. Integration with blockchain can further enhance transparency by immutably logging deletion activities. Support for multi-cloud environments and interoperability across platforms can improve scalability. Enhancing user interfaces for better usability and extending encryption methods to support hybrid or quantum-safe algorithms will strengthen data protection. These improvements will make the system more resilient, adaptable, and applicable to wider cloud-based security scenarios.

References

[1] Tian, T., Liu, Z., Gao, T., Zhang, X., Jin, S., & Liu, X. (2025, April). Blockchain-based verifiable data deletion and software management for cloud storage. In *Fifth International Conference on Telecommunications, Optics, and Computer Science (TOCS 2024)* (Vol. 13629, pp. 513-519). SPIE.

[2] Lapmoon, J., & Fugkeaw, S. (2025). A Verifiable and Secure Industrial IoT Data Deduplication Scheme With Real-Time Data Integrity Checking in Fog-Assisted Cloud Environments. *IEEE Access*.

[3] Darwish, M. A., Markatou, E. A., & Smaragdakis, G. (2025, March). Provable Co-Owned Data Deletion with Zero-Residuals and Verifiability in Multi-Cloud Environment. In *Proceedings of the 18th European Workshop on Systems Security* (pp. 77-83).

[4] Ali, M., & Liu, X. (2025). A Novel Approach to Cloud Security: Publicly Verifiable Remote Signcryption Framework. *IEEE Internet of Things Journal*.

[5] Vairamuthu, G., & Sriraam, A. G. (2025, March). Secure cloud storage for health care data: An integrity auditing protocol ensuring privacy and public verifiability. In *2025 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI)* (pp. 1-6). IEEE.

[6] Ozdemir, A., Wahby, R. S., Whitehat, B., & Boneh, D. (2020). Scaling verifiable computation using efficient set accumulators. *Proceedings of the 29th USENIX Security Symposium*, 1–12.

[7] Benaloh, J., & de Mare, M. (1994). One-way accumulators: A decentralized alternative to digital signatures. *Workshop on the Theory and Application of Cryptographic Techniques*, 274–285.

[8] Papamanthou, C., Tamassia, R., & Triandopoulos, N. (2011). Optimal verification of operations on dynamic sets. *Annual International Cryptology Conference*, 91–110.

[9] Campanelli, M., Fiore, D., Greco, N., Kolonelos, D., & Nizzardo, L. (2020). Incrementally aggregatable vector commitments and applications to verifiable decentralized storage. *International Conference on the Theory and Application of Cryptology and Information Security*, 3–35.

[10] Peikert, C., Pepin, Z., & Sharp, C. (2021). Vector and functional commitments from lattices. *Proceedings of the 19th International Conference on Theory of Cryptography*, 480–511.

[11] Wee, H., & Wu, D. J. (2023). Succinct vector, polynomial, and functional commitments from lattices. *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 385–416.

[12] Zhang, J., Xie, T., Zhang, Y., & Song, D. (2020). Transparent polynomial delegation and its applications to zero knowledge proof. *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, 859–876.

[13] Chen, X., et al. (2021). Publicly verifiable databases with all efficient updating operations. *IEEE Transactions on Knowledge and Data Engineering*, 33(12), 3729–3740.

[14] Wang, Q., Zhou, F., Xu, J., & Peng, S. (2022). Tag-based verifiable delegated set intersection over outsourced private datasets. *IEEE Transactions on Cloud Computing*, 10(2), 1201–1214.

[15] Chen, M., Chiesa, A., Gur, T., O'Connor, J., & Spooner, N. (2023). Proof-carrying data from arithmetized random oracles. *Proceedings of the 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 379–404.

[16] Yang, H., Feng, D., & Qin, J. (2023). Efficient verifiable unbounded-size database from authenticated matrix commitment. *IEEE Transactions on Dependable and Secure Computing*, 20(5), 3873–3889.

[17] Xu, R., Li, C., & Joshi, J. (2023). Blockchain-based transparency framework for privacy preserving third-party services. *IEEE Transactions on Dependable and Secure Computing*, 20(3), 2302–2313.

[18] Liu, D., Huang, C., Ni, J., Lin, X., & Shen, X. S. (2022). Blockchain-cloud transparent data marketing: Consortium management and fairness. *IEEE Transactions on Computers*, 71(12), 3322–3335.

[19] Rivinius, M., Reisert, P., Rausch, D., & Küsters, R. (2022). Publicly accountable robust multi-party computation. *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, 2430–2449.

[20] Lu, J., Li, H., Liu, C., Li, L., & Cheng, K. (2022). Detecting missing-permission-check vulnerabilities in distributed cloud systems. *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2145–2158. https://doi.org/10.1145/3548606.3560589