

## Scalable Access Control Models for Large File Storage and Many Users

<sup>1</sup> Mrs. Reena S Sahane; <sup>2</sup> Ms. Abhilasha Bhagat; <sup>3</sup> Ms. Surbhi Pagar;  
<sup>4</sup> Mrs. Sapana Bhirud; <sup>5</sup> Mrs. Priti Malkhede; <sup>6</sup> Ms. Sadhana Kekan

<sup>1,2,3</sup> Department of AI & DS, DYPIEMR, DYPIU, Akurdi Pune

<sup>4,5</sup> Department of AI & ML, PES MCOE, Pune

<sup>6</sup> Department of Information Technology, MCOE Pune

Corresponding Author: **Reena S Sahane**

**Abstract:** The Hidden Identity File Storage Framework (HIFSF) is designed to ensure that no single entity, including the central storage server, can access both the contents of stored files and information about their owners. In this framework, all user files are stored together in a common repository or unified directory, effectively obscuring ownership details and minimizing the risk of privacy breaches or unauthorized access. To implement this concept, the system replaces the traditional file storage structure with a customized architecture developed and tested within an intranet-based environment. When a user uploads a file, they provide a security key that the system uses to generate a unique identifier. This identifier is then applied to rename the file before storage, thereby removing any direct link between the file and its owner. As a result, the server retains no identifiable metadata that could expose ownership or traceability. During retrieval, the same key-based algorithm regenerates the file's unique name, enabling secure and anonymous access by the rightful user without compromising confidentiality.

**Keywords:** Smart Contracts, Role-Based Access Control, Attribute-Based Access Control, Hybrid Models, Auditability, Data Integrity, Access Revocation

### I. Introduction

In most conventional file storage servers, user data is organized into user-specific folders or directories, typically labeled with the corresponding usernames or unique user identifiers. While this arrangement simplifies file management and access control, it also introduces a significant security vulnerability. The folder naming convention makes the data structure predictable and easily searchable, thereby exposing user ownership information. As a result, if an attacker or unauthorized entity gains access to the server configuration, they can potentially identify specific users and

their associated files, leading to ownership recognition attacks and data privacy breaches.

To overcome these limitations, the proposed invention introduces an Invisible Ownership System (IOS) a secure file storage mechanism designed to conceal file ownership information completely. In this system, no single entity, including the file storage server, has complete knowledge about the ownership or structure of the stored files. Instead of allocating separate directories for each user, all files are stored collectively within a common storage area or shared directory, thereby eliminating the direct linkage between users and their stored data. This design ensures that ownership identity and access patterns remain hidden, thus significantly reducing the risk of unauthorized identification or data targeting.

For proof of concept, the file storage service can be reconstructed and deployed over an intranet to demonstrate the mechanism. In operation, the client transmits a file to the file storage service along with a unique security key. The storage service then renames the file using a unique identification code that is dynamically generated based on the user's key. Once stored, the original filename and ownership details are completely removed, ensuring that the system holds no direct record or trace linking the file to its owner.

During subsequent access or retrieval, the system dynamically recalculates or regenerates the filename using the same user key, allowing authorized users to locate and access their data securely without revealing any ownership information to the server or administrators. This approach effectively provides a high level of anonymity, data confidentiality, and resistance to insider attacks, while maintaining efficient file storage and retrieval operations.

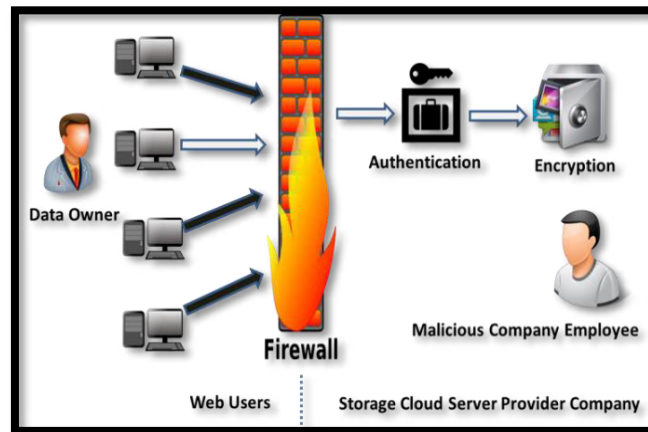
## II. Background

Considering the limitations discussed in the literature review, there is a clear need for a novel and universal approach to address the existing security challenges in cloud-based file storage systems. In modern digital infrastructures, almost every large organization maintains an online presence and operates within a hosting or cloud computing environment, making organizational data a highly critical asset. Although cloud platforms offer advanced protection against external threats such as malware and virus attacks, the data still remains vulnerable to malicious insiders or users with administrative privileges. Individuals possessing superuser or root-level access can potentially exploit their rights to view, modify, or extract sensitive data, posing a significant security risk.

After an extensive study of various mitigation techniques designed to reduce malicious insider attacks, it was observed that the most effective method to secure data is to make it inaccessible to unauthorized users altogether. However, since the data must still reside somewhere for legitimate access, a balance must be achieved between availability and concealment. To address this, the proposed system

introduces a distributed and concealed data storage mechanism that ensures the original data remains hidden from both users and administrators.

In this system, data is first encrypted and intelligently divided into multiple blocks based on the user's request and the system's decision-making logic. These encrypted blocks are then stored across multiple cloud storage locations, ensuring that even if a malicious user gains access to one or more storage units, the data remains unusable and meaningless without the complete set of encrypted fragments and the corresponding decryption logic.

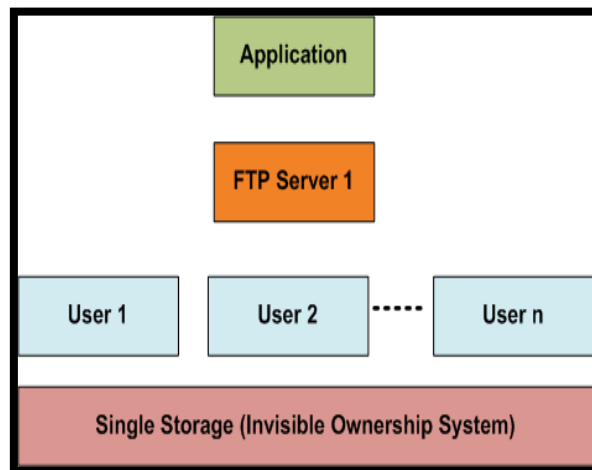


**Figure 1.0 Security breaches**

Conventional file-storage servers organize data into user-specific directories typically named with usernames or identifiers, which makes ownership information trivially discoverable and folders easily searchable by an attacker or a malicious insider. To counter this vulnerability, the proposed Distributed Hidden Ownership Mechanism (D-HOM) with an Invisible Ownership System stores all users' files within a common namespace and removes any persistent owner-linked metadata: clients submit files along with a secret key, the service splits each file into multiple parts ( $F \rightarrow F_1, F_2, F_3 \dots$ ), encrypts each part independently ( $F_{1e}, F_{2e}, F_{3e}$ ), and then renames and stores the encrypted fragments using cryptographic identifiers derived from the client key. Because filenames and storage identifiers are deterministically generated from secret key material and no mapping to user identities is retained on the storage nodes, administrators or cloud insiders cannot determine file ownership or content. For proof of concept, the storage service can be rebuilt and tested over an intranet as a middleware/gateway that implements split/encrypt/rename/push operations; authorized clients re-derive fragment identifiers from their keys to fetch, decrypt, and reassemble files, while lost keys render data unrecoverable unless secure key-escrow or threshold recovery mechanisms are implemented.

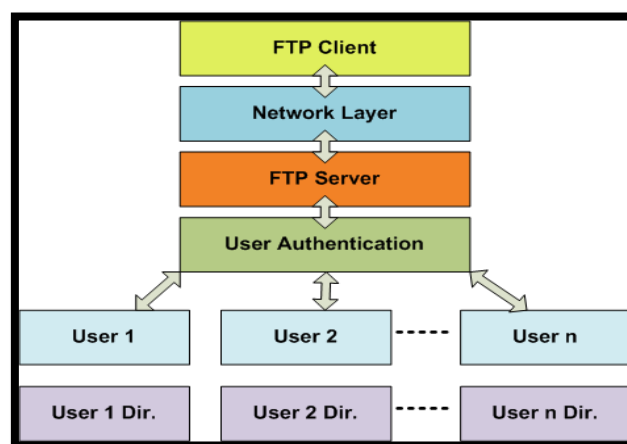
### III. Proposed System

This present invention discloses system and method for secure file storage using hidden owner identity mechanism. Currently with the increase in the creation and use of digital data, enormous amount of data is created on file storage server which is maintained by third party vendor hence there is need to provide secure system which will store all user's data securely on system where only authorized user knows complete information.



**Figure 1.0 System Protocol Architecture**

Since file storage server stores all Users data in various user directories, creates threat of user ownership recognition where system administrator attack is possible. We have developed secure file storage system where file owner information user is hidden hence nobody including user, administrator and file storage server has complete information about stored data. Our system reduces indexing overheads and cumbersome security key management.



**Figure 2.0 Existing FTP process**

### Invisible Ownership System

- Proposed New Method /Algorithm for improving FTP services
- Header Deletion function will remove ownership details of file chunks to provide OS level security
- Proposed system will dynamically create naming convention ambiguity by renaming each file
- New name is generated using Filename, Server name, user key and cloud private key

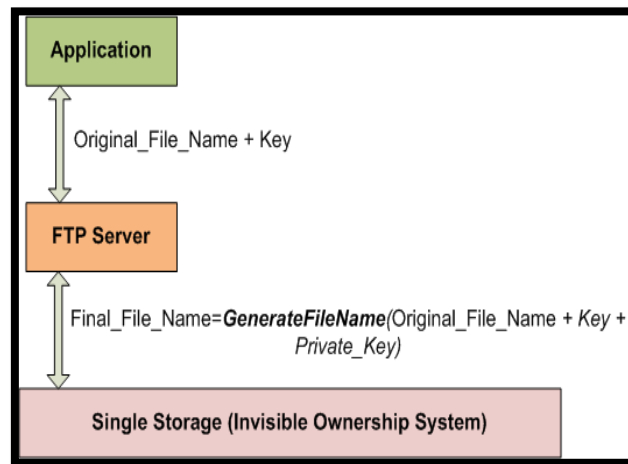


Figure 3.0 System process

This present disclosure relates to the novel work carried for implementation of a Hidden Owner Identity mechanism for securely storing file on storage server. Moreover, it relates to unique, efficient and secure method for storing a file on any type of storage server either local or cloud-based server. Furthermore, present disclosure relates to computer program product consisting of computer -readable instructions stored on computer -readable storage media. These instructions being executed by computing system consisting of processing hardware to execute programs. File storage server is a server which stores various types of critical user data files and privacy -sensitive information using file systems therefore they are main targets for various types of security attacks.

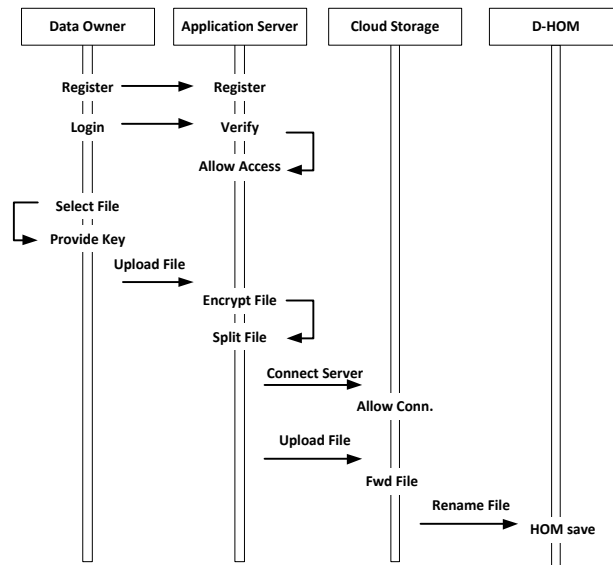
### IV. Implementation

The sequence diagram illustrates the timeline of operations involved in the file processing workflow within the proposed implementation. The process begins with the registration of the client on the application, during which login credentials are generated to enable subsequent verification and authentication. Once the client is successfully authenticated, they are granted permission to initiate file processing within the secured environment.

The client then selects a file for upload and generates a secure encryption key, which is used to create an authenticated upload request to the designated cloud storage.

Following this, the system invokes the Encryption and Splitting APIs, which are responsible for encrypting the selected file, dividing it into multiple secure fragments, and distributing those fragments across the available cloud infrastructure in a protected manner.

This entire sequence can be executed in reverse during the file retrieval process, where the client initiates a download request. The system then reassembles and decrypts the distributed file fragments, ensuring the successful recovery of the original file and maintaining client satisfaction through secure and reliable access.




---

**Algorithm: Save file (HOM)**

---

1. Get File → FL
  2. Get Key → K
  3. Generate GUID → G
  4. Rename FL=Enc(Gk)
  5. CreateFile(FP)
  6. Write FL→ FP
  7. Save FP → CentralStorage
- 

**Conclusion**

In presented proposed system we analysed different aspects of cloud computing and different file storage mechanism including local and distributed file system. Across the development we studied different available solutions and tried to find out what parameters can be considered and re-designed to enhance the performance or minimize the cost factor. We carried out the development of D-HOM (Distributed Hidden Ownership Mechanism) system which will enhance the file security by encrypting and distributing the file parts over multiple cloud and re-join when

accessing. With simulation and implementation, we generated the performance result and resulting parameters are enhanced.

## References

1. A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.
2. G. R. Blakley, "Safeguarding cryptographic keys," in *Proceedings of the National Computer Conference (AFIPS)*, vol. 48, 1979, pp. 313–317.
3. F.-H. Chen, Y.-C. Lin, and S.-H. Chien, "Secure and reliable distributed file system for cloud storage," *Journal of Cloud Computing*, vol. 5, no. 1, pp. 1–12, 2016.
4. K. K. Mar and H. S. Lim, "Multi-cloud based secure virtual diffused file system," in *Proc. International Conference on Cloud Computing and Virtualization*, Singapore, 2019, pp. 45–50.
5. Z. Wu, M. Zhou, and W. Wei, "Research on multi-cloud storage security and data integrity verification," *IEEE Access*, vol. 7, pp. 68936–68948, 2019.
6. J. Wilcox-O’Hearn and B. Warner, "Tahoe: The least-authority file system," in *Proc. 4th ACM International Workshop on Storage Security and Survivability (StorageSS)*, 2008, pp. 21–26.
7. K. R. Krishna and S. Ram, "Enhanced security model for data storage in multi-cloud environment," *International Journal of Computer Applications*, vol. 179, no. 48, pp. 1–6, 2018.
8. R. Buyya, C. Vecchiola, and S. T. Selvi, *Mastering Cloud Computing: Foundations and Applications Programming*, Elsevier, 2013.
9. S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 1–11, 2011.
10. N. Santos, K. P. Gummadi, and R. Rodrigues, "Towards trusted cloud computing," in *Proc. 2009 Conference on Hot Topics in Cloud Computing (Hot Cloud)*, USENIX Association, 2009.
11. J.-M. Bohli, N. Gruschka, M. Jensen, L. Lo Iacono, and N. Marnau, "Security and privacy enhancing multi-cloud architectures," *IEEE Transactions on Dependable and Secure Computing*, Jan. 2013.
12. Z. Xiao and Y. Xiao, "Security and privacy in cloud computing," *IEEE Communications Surveys & Tutorials*, Mar. 2012.
13. A. Malik and M. M. Nazir, "Security framework for cloud computing environment," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 3, no. 3, Mar. 2012.
14. M. Singhal and S. Chandrasekhar, "Collaboration in multicloud computing environments: Framework and security issues," *IEEE Computer Society*, 2013.

15. M. A. AlZain, E. Pardede, B. Soh, and J. A. Thom, "Cloud computing security: From single to multi-clouds," in Proc. International Conference on System Sciences, 2012.
16. K. Yang, X. Jia, B. Zhang, and R. Xie, "DAC-MACS: Effective data access control for multi-authority cloud storage systems," IEEE, 2013.
17. P. Mell and T. Grance, "The NIST definition of cloud computing," National Institute of Standards and Technology, Tech. Rep., Sept. 2011.
18. J.-J. Hwang and H.-K. Chuang, "A business model for cloud computing based on a separate encryption and decryption service," IEEE, 2012.
19. J.-M. Bohli, M. Jensen, N. Gruschka, J. Schwenk, and L. L. Iacono, "Security prospects through cloud computing by adopting multiple clouds," in Proc. IEEE 4th Int'l Conf. Cloud Computing (CLOUD), 2011.
20. M. Jensen, J. Schwenk, N. Gruschka, and L. Lo Iacono, "On technical security issues in cloud computing," in Proc. IEEE Int'l Conf. Cloud Computing (CLOUD-II), 2009.
21. K. Yang and X. Jia, "Attribute-based access control for multi-authority systems in cloud storage," in Proc. 32nd IEEE Int'l Conf. Distributed Computing Systems, 2012.
22. M. A. AlZain, B. Soh, and E. Pardede, "MCDB: Using multi-clouds to ensure security in cloud computing," in Proc. 9th IEEE Int'l Conf. Dependable, Autonomic and Secure Computing, 2011.