Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 7, No. 4, July : 2023

# PRIVACY GUARD: EMPOWERING DATA PRIVACY THROUGH FRAGMENTATION STORAGE MODEL

**Mr. Rajwardhan S. Todkar**, M.Tech Student,
**Dr. Jaydeep B. Patil***, Assistant Professor (M.Tech Guide),
**Miss. Monica S. Todkar**, M.Tech Student,
Dept. of Computer Science and Engineering (AI & ML), D. Y. Patil Agriculture and Technical University, Talsande, 416 112, India. Email address: jaydeep.patil@dyp-atu.org.

**Abstract:**
With the rapid advancement of digital technologies and the increased reliance on cloud storage, ensuring privacy and data security has become a critical concern. The fragmentation storage model is a novel privacy protection technology that aims to enhance the security and confidentiality of data stored in cloud environments. This research paper presents a detailed analysis of the fragmentation storage model, its architecture, benefits, and potential applications. The paper also highlights the efficiency and effectiveness of the model in preserving privacy and mitigating data breaches. Furthermore, it discusses the challenges and future research directions in the field of privacy protection.

**Keywords:** Fragmentation Storage Model, Privacy Protection, Efficient, Data Fragmentation, Distributed Storage, Encryption, Access Control, Authentication, Performance Evaluation, Cloud Storage, Privacy Preservation Mechanisms, Data Security, Privacy Challenges.

## Introduction

The growing popularity of cloud storage has led to an increasing need for robust privacy protection mechanisms. Traditional encryption techniques are often vulnerable to attacks, and they cannot guarantee complete privacy. The fragmentation storage model addresses these limitations by leveraging the concept of data fragmentation and distributed storage. This paper provides an overview of the model's objectives and the structure of the subsequent sections. Users can access a wide range of customizable resources, data, and results thanks to the decentralized data storage and management platform known as cloud computing. However, because the owner no longer has direct access to the data, privacy and security concerns must be addressed. There are issues with confidentiality. Encryption techniques have already been suggested as a way to protect user privacy when outsourcing storage. On the other side, a lot of these encryption techniques are risky, which implies that data security can be compromised just by making an algorithm weaker. As a result, a strategy for guaranteeing that encryption and distribution systems uphold confidentiality, security, and privacy must be developed. Users of the technology known as cloud computing can alter a variety of services. While decentralized data management and storage. However, there have been worries regarding data confidentiality because the owner no longer has authority over the data. Customers who use cloud storage are given an abstraction of limitless storage space that they can use to house data on an as-needed basis. Terabytes of data may be stored on a website that allows sharing of data or videos, for instance. Businesses can now disseminate the storage of a big amount of data instead of maintaining their own data centers. To third-party distributed storage providers, reducing the cost of data management overhead. A simple solution to ensure data redistribution is to encrypt critical data using cryptography and a large number of encryption keys. Keeping track of and protecting such encryption keys, however, raises still another security concern. One such issue is requests to erase files. We want to achieve a big security goal by dividing the data into small parts and storing them on many cloud hubs. Data is broken up using the pseudo random number. Random numbers 1 are those that show up when two conditions are met: (1) There is equal distribution of the numbers inside a specified range

or collection. (2) Random numbers are unrelated to one another; in other words, one cannot predict the next number by knowing the previous ones.

**Objective**
The proposed system will be designed in the following four modules:
1. Designing the distributed database
2. User creation and Assigning access controls based on the roles
3. Metadata Management
4. Implementing security algorithms

The objective is to design a system to fix the below-mentioned issues
1. Security, and availability of data on cloud storage: Cloud data is repeatedly kept in basic text, which can lead to leakage of sensitive data.
2. Avoiding the bottleneck: The simultaneous multiple users accessing the application leads to the bottleneck.
The DD-PLAC (Proxy-less Architecture with Encrypted Metadata in the Cloud and Distributed Database) technique was designed to address the issues with privacy, availability, and bottleneck. When data is spread over the cloud, a distributed cloud database is employed, enabling the databases to fully fulfill the elasticity requirements of cloud computing applications. For some time, databases have been spread among servers with access to fast networks as instances running on those hosts. Depending on their position, access permissions are assigned to each user. The privacy of the data saved in the cloud is enhanced by security algorithms.

**2.2 Methodology**
The three architecture kinds listed below are those that are defined to protect privacy.
1. PBA (Proxy Based Architecture).
2. PLA (Proxy-less architectures) that keep client-side metadata.
3. **PLAC (**Proxy-less architectures**)** that use cloud databases to store metadata.

Discussion
**1. Proxy-based architectural designs (PBA)**
The proxy is a bottleneck and a single point of failure that restricts the cloud DBaaS's availability, scalability, and elasticity, hence proxy-based systems cannot meet our design requirements. Due to the need for trust, the proxy cannot be deployed or maintained in the cloud; rather, it must be done locally. Proxy-based systems also are unable to scale simply by adding more proxies. Such a simple solution would imply the replication of metadata across all proxies, but to ensure consistency across all proxies, synchronization methods, and protocols would be necessary.
**2. Architectures without proxies that save metadata on clients (PLA)**
The Proxy-less architectures [4] that keep metadata in the clients don't employ a middle proxy; instead, the metadata is saved on the client side. This design provides availability, scalability, and elasticity so the clients can connect directly to the cloud database. Since each client keeps a local copy of the metadata, each one has its encryption engine. Therefore, this approach can be seen as a sub-case of the proxy-based architecture, where each client has a unique proxy installed. The consistency problems of proxy-based systems would apply to a comparable architecture for cloud accesses.

**3 Cloud-based architecture without proxies and encrypted metadata**
The third architecture, displayed in Fig. 4.3[6], is a proxy-less architecture that stores metadata in a cloud database. Although each client executes the encryption engine, the metadata is saved in a cloud database in this case. There is no need for synchronization because metadata are not shared by all

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 7, No. 4, July : 2023

clients. Client computers run a client software component that enables a user to connect to and submit queries to the cloud DBaaS directly. By using SQL statements, this software component extracts the required metadata from the untrusted database and provides it to the client's encryption engine. The untrusted cloud database is highly available, scalable, and elastic, allowing several clients to access it independently.

**Plan of Dissertation Execution**

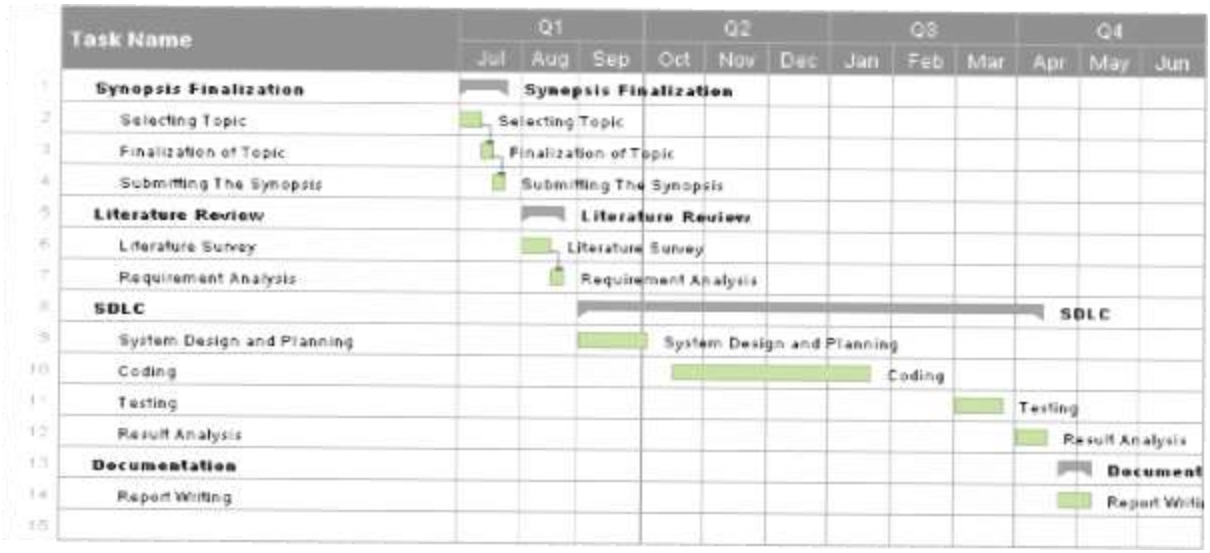


Fig. Plan of Dissertation

**Architecture of System**

The distributed cloud database, where the data is distributed over the cloud, is used in the DD-PLAC architecture, which is the architecture that stores metadata in the cloud database. This allows the databases to truly support the elastic requirements of cloud computing applications. Databases have long been deployed as instances that run on servers with high-speed network connections.

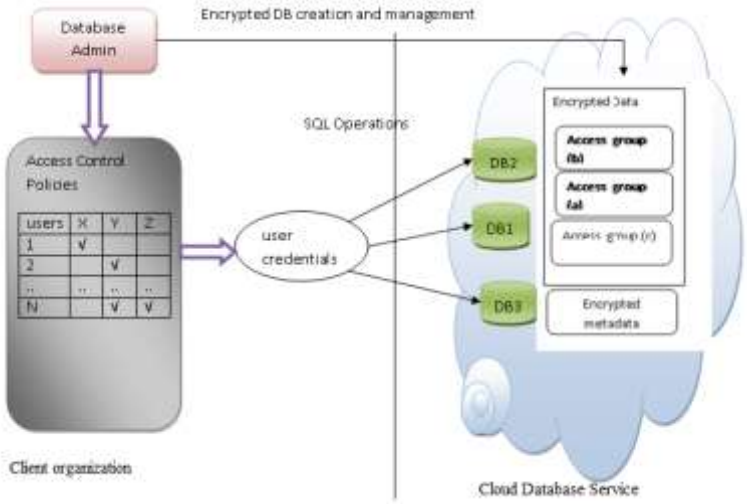


Fig. System architecture

The DD-PLAC client, which is the program used to create and manage the encrypted database, is hosted by a client organization on a trusted Database Admin workstation. By running a DD-PLAC client on their computers, all database users can directly submit SQL operations to the cloud database, even from geographically dispersed places. The cloud database stores the whole collection of data in an encrypted format. The cloud database engine can run queries on encrypted data without having

access to any decryption keys because of the SQL-aware encryption techniques. Even metadata required to maintain encryption schemes are regarded as vital information, thus DD-PLAC keeps them encrypted in the cloud database. The Database Admin and users can effectively obtain metadata through ordinary SQL queries. It is assumed that the database administrator is the only person with root access to the client application and that neither internal nor external attackers can access or decipher the credentials.

**Background and Related Work**

This section discusses the fundamental concepts and related work in the field of privacy protection and cloud storage. It covers topics such as encryption techniques, secure data sharing, and privacy-preserving models. The discussion of related work helps establish the novelty and significance of the fragmentation storage model.

The Architecture of the Fragmentation Storage Model The fragmentation storage model is based on the concept of dividing data into multiple fragments and storing them across different cloud servers. This section presents a detailed description of the architecture, components, and functionalities of the model. It highlights the roles of the client, storage nodes, and the fragmentation algorithm in achieving efficient privacy protection.

Fragmentation Algorithm The fragmentation algorithm plays a crucial role in dividing data into fragments and distributing them across multiple cloud servers. This section presents a comprehensive analysis of the fragmentation algorithm employed in the model. It discusses the factors influencing the fragmentation process, such as data size, security requirements, and storage resources.

Privacy Preservation Mechanisms The fragmentation storage model incorporates several privacy preservation mechanisms to enhance data security. This section explores the various techniques employed, including data encryption, access control mechanisms, and authentication protocols. It discusses how these mechanisms work together to safeguard data privacy throughout the storage and retrieval processes.

Performance Evaluation To assess the efficiency and effectiveness of the fragmentation storage model, a performance evaluation is conducted. This section presents the experimental setup, metrics, and results obtained from the evaluation. The evaluation demonstrates the model's ability to protect privacy while maintaining acceptable performance levels in terms of storage overhead, response time, and scalability.

Applications of the Fragmentation Storage Model This section explores the potential applications of the fragmentation storage model in different domains. It discusses how the model can be utilized in areas such as healthcare, finance, e-commerce, and government sectors to ensure secure and private data storage. Real-world use cases and implementation scenarios are also presented.

**Conclusion**

This report wraps up by listing the major discoveries and contributions made by the investigation. It emphasizes the importance of the fragmentation storage model as an efficient privacy protection technology for cloud storage environments. The paper also reiterates the significance of ongoing research and development in this field to address emerging privacy challenges.

**References**

1. Aggarwal, G., Bawa, M., Ganesan, P., Garcia-Molina, H., Kenthapadi, K., Mot- wani, R., Srivastava, U., Thomas, D., Xu, Y.: Two can keep a secret: a distributed architecture for secure database services. In: Proc. of the 2nd Conference on Innovative Data Systems Research (CIDR 2005), Asilomar, California, USA (January 2005)
2. California Senate bill SB 1386 (September 2002)

3. Caselli, A., Damiani, E., De Capitani di Vimercati, S., Jajodia, S., Paraboschi, S., Samarati, P.: Modeling and assessing inference exposure in encrypted databases. ACM Transactions on Information and System Security 8(1) (February 2005) 119– 152
4. Ciriani, V., De Capitani di Vimercati, S., Foresti, S., Samarati, P.: k-anonymity. In Yu, T., Jajodia, S., eds.: Security in Decentralized Data Management. Springer, Berlin Heidelberg (2007)
5. Damiani, E., De Capitani di Vimercati, S., Jajodia, S., Paraboschi, S., Samarati, P.: Balancing confidentiality and efficiency in untrusted relational DBMSs. In: Proc. of the 10th ACM Conference on Computer and Communications Security (CCS03), Washington DC, USA (October 2003)
6. Dawson, S., De Capitani di Vimercati, S., Lincoln, P., Samarati, P.: Maximizing sharing of protected information. Journal of Computer and System Sciences 64(3) (May 2002) 496–541
7. Garey, M.R., Johnson, and D.S.: Computers and intractability: a guide to the theory of NP-completeness. W.H. Freeman (1979)
8.Hacigu¨mu¨s, H., Iyer, B., Mehrotra, S.: Providing database as a service. In: Proc. of the 18th International Conference on Data Engineering (ICDE02), San Jose, California, USA (February 2002)
9. Hacigu¨mu¨s, H., Iyer, B., Mehrotra, S., Li, C.: Executing SQL over encrypted data in the database-service-provider model. In: Proc. of the 21st ACM SIGMOD International Conference on Management of Data, Madison, Wisconsin, USA (June 2002)
10. Krivelevich, M., Sudakov, B.: Approximate coloring of uniform hypergraphs. Journal of Algorithms 49(1) (2003) 2–12
11. Navathe, S., Ceri, S., Wiederhold, G., Dou, J.: Vertical partitioning algorithms for database design. ACM Transaction on Database Systems 9(4) (December 1984) 680–710
12. Navathe, S., Ra, M.: Vertical partitioning for database design: a graphical algorithm. In: Proc. of the 1989 ACM SIGMOD International Conference on Management of Data, Portland, Oregon, USA (June 1989)
13. Payment card industry (PCI) data security standard (September 2006)
https://www.pcisecuritystandards.org/pdfs/pci dss v1-1.pdf.
14. Personal data protection code. Legislative Decree No. 196 (June 2003)
15. Samarati, P.: Protecting respondent's privacy in microdata release. IEEE Transactions on Knowledge and Data Engineering 13(6) (November/December 2001) 1010–1017
16. Schneier, B.: Applied Cryptography: protocols, algorithms, and source code in C.
John Wiley & Sons, 2/E, New York (1996)