ISSN: 2366-1313

SECURE ACCESS TO CLOUD DATABASES WITH DISTRIBUTED, CONCURRENT FUNCTIONALITY, ENSURING INDEPENDENCE FROM THIRD PARTIES

[#1]Ch. Balakrishna, Assistant Professor,
[#2]V.Sai Rama Krishna, Assistant Professor,
[#3]B. Santhosh Kumar, Assistant Professor,
Department of Computer Science and Engineering,
SAI SPURTHI INSTITUTE OF TECHNOLOGY, SATHUPALLY, KHAMMAM.

ABSTRACT: When you give your sensitive data to a cloud provider, they must ensure the security and accessibility of your data while it is stored, being transferred, and being utilized. Although there are many storage services available, the development of data secrecy solutions for the database as a service paradigm is still in its early stages. Our unique design integrates cloud database services with data privacy and the capability to execute several operations on encrypted data concurrently. This is the inaugural solution that enables remote clients to establish a secure connection with a cloud database and execute simultaneous, autonomous operations, including those that modify the database's structure. An additional benefit of the proposed method is the lack of intermediate proxies, which restrict the inherent flexibility, accessibility, and scalability of cloud-based solutions. The efficacy of the proposed architecture is evaluated by comprehensive empirical evidence obtained from a prototype implementation that underwent the TPC-C standard test, encompassing diverse client quantities and network latencies.

Keywords: TPC-C, SQL,DBMS.

1. INTRODUCTION

The architecture was designed with three primary goals: facilitating concurrent operations on encrypted data by multiple, independent, and geographically dispersed clients, including SQL statements that alter the database structure; guaranteeing data confidentiality and consistency at both the client and cloud levels; and removing any intermediary servers between the cloud client and cloud provider. The Secure DBaaS prototype demonstrates the capability to execute concurrent and independent operations on a distant encrypted database from several clients situated in diverse geographical areas. This configuration, like an unencrypted Database-as-a-Service (DBaaS), emphasizes the possibility of combining the ease of access, adaptability, and scalability of a typical cloud DBaaS with the protection of data privacy. Secur DBaaS employs well-established cryptographic techniques, isolation measures, and innovative management strategies to efficiently maintain encrypted metadata on a cloud database that is not trusted.

This study examines theoretical approaches to tackle the issues of data consistency that arise when many clients view encrypted data concurrently and independently. Implementing completely homomorphic encryption techniques in this specific circumstance is not practical due to their substantial processing overhead.

It eliminates the need for intermediary proxies or broker servers in the communication between the client and the cloud service provider. By eliminating trusted intermediate servers, secure DBaaS may provide equivalent levels of availability, reliability, and elasticity as cloud DBaaS.

The benefits of utilizing a cloud-based database service, such as its ability to easily expand, remain accessible, and offer adaptability, are constrained by the vulnerability of any intermediary server functioning as a proxy, which can serve as a sole point of failure and a bottleneck. Hence, other suggestions that depend on intermediary server(s) were deemed unsuitable for a cloud-based

ISSN: 2366-1313

solution. Systems that rely on a trusted intermediary proxy, as opposed to Secure DBaaS, are unable to facilitate the typical cloud scenario in which clients in different geographic areas can concurrently perform read/write operations and modify the data structure of a cloud database. Empirical experiments carried out on operational cloud platforms provide evidence that Secure DBaaS may be easily implemented with any DBMS, without the need for any alterations to the cloud database services. Further investigation, encompassing the TPC-C standard test and the proposed design for different quantities of clients and network latencies, reveals that the performance of the Secure DBaaS database remains unaffected by concurrent read and write operations.

2. EXISTINGSYSTEM

Access to the original, unencrypted data should only be granted to reliable establishments, whereas cloud providers, middlemen, and the internet should be exempted from this regulation. Encryption of data is necessary in situations when there is a notable absence of confidence. The cloud service you utilize may impact the ease with which you can accomplish these objectives. The level of confidentiality provided by the database as a service (DBaaS) paradigm requires further analysis for complete understanding. Conversely, the security of the storage as a service (SaaS) paradigm can be guaranteed by several methods.

Disadvantages of Existing System

The use of fully homomorphic encryption algorithms is impeded by their substantial computational complexity.

3. PROPOSEDSYSTEM

Our system offers a unique blend of cloud database services, data privacy, and the ability to perform many actions on encrypted data simultaneously, all of which enhance the system's overall competitive edge. This service offers clients in various locations the capability to access a cloud database concurrently and securely, enabling them to perform a range of tasks, including updating the database's structure.

This represents the initial phase in the creation of a more extensive and all-encompassing resolution. The proposed approach also bypasses intermediary proxies, so overcoming the inherent constraints of cloud-based systems in terms of their inherent flexibility, accessibility, and scalability. Secure DBaaS stands apart from previous endeavors in the realm of securing remote database services due to its several unique characteristics.

Advantages of Proposed System

The suggested framework can be readily executed using existing cloud Database-as-a-Service (DBaaS) platforms, such as PostgreSQL Plus Cloud Database, Windows Azure, and Xeround, without necessitating any alterations to the cloud database. There are no theoretical or practical constraints that hinder us from implementing other encryption techniques and extending our solution to include other platforms. This method ensures data secrecy by allowing a cloud database server to perform concurrent SQL operations (such as reading, writing, and altering the database structure) on encrypted data. By removing the requirement for an intermediary server, it provides the same levels of availability, flexibility, and scalability as the original cloud DBaaS.

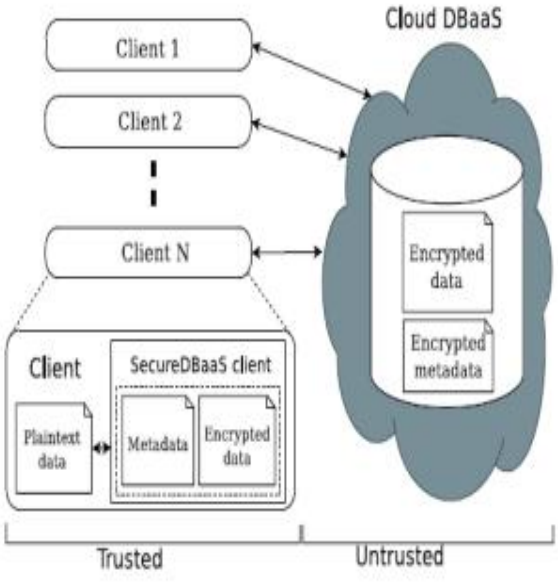

Fig1.SystemArchitecture

4. RELATEDWORK

Secure DBaaS stands out from previous endeavors in the realm of securing remote database services due to its numerous distinctive attributes. The capacity of a cloud database server to execute concurrent SQL operations on encrypted data,

ISSN: 2366-1313

such as reading/writing and modifying the database structure, offers an extra level of safeguarding for the data. Both this server and the intermediate server possess same levels of scalability, accessibility, and adaptability. The network often introduces latencies that obscure the impact of cryptographic overheads on SQL operations.

An advantage of utilizing a cloud database service is its ability to facilitate simultaneous and independent access to the database by several users, regardless of their geographical location. A trustworthy intermediary is unnecessary as the cloud database ensures continuous safeguarding of the data and metadata owned by the database tenants. All selected solutions can be used with various DBMS implementations and are compatible with the most commonly used relational database servers due to their database-agnostic nature.

The initial notable progress in this domain occurred with the creation of encrypted file systems and secure storage technology. Since Sporc, Sundr, and Depot lack the capability to handle encrypted data, we will refrain from discussing the extensive research and diverse range of products offered by these firms.

Through the distribution of data among multiple sources and the transmission of sensitive information, different approaches offer varying levels of privacy.

The installation of these safeguards prohibits a cloud provider from accessing its own data; yet, data retrieval remains possible if cloud providers collaborate. It is recommended to employ a unique approach to enhance the effectiveness of range data searches and increase resilience against collusive providers.

Secure DBaaS distinguishes itself from competing options by not requiring the use of different cloud providers. Furthermore, this solution allows for the execution of most common SQL operations on encrypted data.

Secure Database as a Service (DBaaS) is a software category that employs many encryption methods to safeguard critical data stored in unreliable databases. The primary difficulty in this circumstance arises from the inability of DBMS to directly apply encryption techniques on standard DBaaS, as it restricts the execution of SQL operations on encrypted data. Transparent data encryption is a feature that allows for the encryption of data at the file system level. This feature is accessible in specific Database Management System (DBMS) engines. Due to this attribute, it is feasible to create a reliable database management system even when utilizing volatile storage. Conversely, the database management system is reliable and performs data decryption prior to its utilization. This solution is deemed ineligible for implementation in the DBaaS environment, which is the primary focus of the Secure DBaaS inquiry, due to our lack of confidence in the cloud service provider. The utilization of various approaches facilitates the handling of encrypted data. When the reliability of the DBMS is doubted, several methods might be employed to safeguard the privacy of the data. However, these databases necessitate the use of a specialist DBMS engine and are not compatible with the DBMS software offered by cloud service providers, regardless of whether the software is commercially available or open-source.

Secure DBaaS enables tenants to establish secure cloud databases by leveraging existing Cloud DBaaS services, while ensuring compatibility with traditional DBMS engines. Secure Database-as-a-Service (DBaaS) is similar to services that provide support for popular Database Management System (DBMS) engines, enable the execution of SQL operations on encrypted data, and utilize encryption methods to guarantee data secrecy in DBMSs that are not deemed trustworthy. Nevertheless, the architecture of these systems necessitates a trustworthy intermediary proxy as the crucial element, as it is this element that enables all interactions between each client and the unstable DBMS server.

The proponents of the DBaaS paradigm suggest encrypting data in chunks, also referred to as segments, rather than encrypting individual data elements. To ensure the reliable proxy does its function effectively, it must complete the following tasks: retrieve the entire block, decrypt it, and remove any unnecessary data from the block once a specific data item is needed.

ISSN: 2366-1313

Hence, due to this design decision, substantial modifications must be made to the initial SQL operations generated by every client, hence imposing a considerable burden on both the DBMS server and the trusted proxy. The scope of SQL operators that can be managed is broadened due to further enhancements, such as optimization and generalization. However, these enhancements are still constrained by the inherent limitations associated with proxy-based architecture. SQL-aware encryption techniques can be utilized on encrypted data through Secure Database as a Service (DBaaS), commonly referred to as cloud computing database.

The proposed technique, initially introduced in CryptDB, facilitates the execution of operations on encrypted data that are equivalent to those conducted on plaintext. The DBMS consistently employs the same query approach for acquiring both encrypted and unencrypted data. Their main focus is on establishing multitier web applications, and using a reliable proxy makes the process of creating a secure DBaaS easier and more straightforward. However, there are a handful of unfavorable consequences stemming from it.

Due to the reliable reputation of the proxy, it cannot be entrusted to an untrustworthy cloud provider. Therefore, the responsibility for both the initial setup and continuous administration of the proxy lies with the cloud tenant. The trusted proxy is the sole vulnerability in the secure DBaaS service, hindering the general accessibility, scalability, and flexibility of the service.

The constraint that has been imposed is a substantial obstacle to implementing the cloud database scenario, as high availability, scalability, and elasticity are crucial factors in adopting cloud services. To address this problem, one can resolve it by employing Secure Database-as-a-Service (DBaaS), which facilitates direct connections between clients and the cloud-based DBaaS. This minimizes reliance on intermediate components and decreases the probability of new bottlenecks and potential failure points.

5. CONCLUSSION

Our cutting-edge approach ensures the privacy of data stored in public cloud databases. From our viewpoint, an intermediate proxy is both a vulnerable point that might cause system failure and a limitation that hinders the accessibility and scalability of typical cloud database services.

A substantial section of the study is dedicated to devising solutions that enable concurrent SQL operations on encrypted data from several clients, some of whom may be located far apart geographically. This includes handling statements that affect the database's structure.

The proposed solution can be readily implemented on existing cloud DBaaS platforms, such as PostgreSQL Plus Cloud Database, Windows Azure, and Xeround, without necessitating any alterations to the cloud databases. In theory and in practice, it is possible to integrate more encryption methods and expand our system to include multiple platforms. According to the testing findings obtained from the TPC-C standard benchmark, the typical network delay in cloud environments renders the impact of data encryption on response time insignificant.

Simultaneous read and write operations that do not modify the structure of the encrypted database do not encounter any delay. It is possible to have dynamic scenarios where changes are made to the database structure at the same time, but this comes with a significant computational expense. The performance results provide a foundation for potential future progress that we are now investigating.

REFERENCE

1. M. Armbrust et al., A View of Cloud Computing, Comm. of the ACM, vol. 53, no. 4, pp. 50-58, 2010.
2. W. Jansen and T. Grance, Guidelines on Security and Privacy in Public Cloud Computing, Technical Report Special Publication 800-144, NIST, 2011.
3. A.J. Feldman, W.P. Zeller, M.J. Freedman, and E.W. Felten, SPORC: Group Collaboration Using Untrusted Cloud Resources, Proc. Ninth USENIX Conf. Operating Systems Design and Implementation, Oct. 2010.

ISSN: 2366-1313

4. J. Li, M. Krohn, D. Mazie `res, and D. Shasha, Secure Untrusted Data Repository(SUNDR), Proc. SixthUSENIX Conf. Opearting Systems Design and Implementation, Oct. 2004.
5. P. Mahajan, S. Setty, S. Lee, A. Clement, L. Alvisi, M. Dahlin, and M. Walfish, Depot: Cloud Storage with Minimal Trust, ACM Trans. Computer Systems, vol. 29,no. 4, article 12, 2011.
6. H. Hacigu ¨mu ¨s¸, B. Iyer, and S. Mehrotra, Providing Database as a Service, Proc. 18th IEEE Int'l Conf. Data Eng., Feb. 2002.
7. C. Gentry, Fully Homomorphic Encryption Using Ideal Lattices,Proc.41stAnn. ACMSymp.TheoryofComputing, May 2009.
8. R.A. Popa, C.M.S. Redfield, N. Zeldovich, and H. Balakrishnan, CryptDB: Protecting Confidentiality with Encrypted Query Processing, Proc. 23rd ACM Symp. Operating Systems Principles, Oct. 2011.
9. H. Hacigu ¨mu ¨s¸, B. Iyer, C. Li, and S. Mehrotra, Executing SQL over Encrypted Data in the Database- Service-Provider Model, Proc. ACM SIGMOD Int'l Conf. Management Data, June 2002.
10. J. Li and E. Omiecinski, Efficiency and Security Trade-Off in Supporting Range Queries on Encrypted Databases, Proc. 19th Ann. IFIP WG 11.3 Working Conf. Data and Applications Security, Aug. 2005.