

Distributed, Concurrent, and Independent Access to Encrypted Cloud Databases

Kandadi Thirupathi Reddy

M.Tech Student,
Department of CSE,
CMR College of Engineering and Technology.

Mr. Shiva Skandha, M.Tech

Associate Professor,
Department of CSE,
CMR College of Engineering and Technology.

ABSTRACT:

Placing critical data in the hands of a cloud provider should come with the guarantee of security and availability for data at rest, in motion, and in use. Several alternatives exist for storage services, while data confidentiality solutions for the database as a service paradigm are still immature. We propose a novel architecture that integrates cloud database services with data confidentiality and the possibility of executing concurrent operations on encrypted data.

This is the first solution supporting geographically distributed clients to connect directly to an encrypted cloud database, and to execute concurrent and independent operations including those modifying the database structure. The proposed architecture has the further advantage of eliminating intermediate proxies that limit the elasticity, availability, and scalability properties that are intrinsic in cloud-based solutions.

The efficacy of the proposed architecture is evaluated through theoretical analyses and extensive experimental results based on a prototype implementation subject to the TPC-C standard benchmark for different numbers of clients and network latencies.

EXISTING SYSTEM:

Original plain data must be accessible only by trusted parties that do not include cloud providers, intermediaries, and Internet; in any untrusted context, data must be encrypted. Satisfying these goals has different levels of complexity depending on the type of cloud service.

There are several solutions ensuring confidentiality for the storage as a service paradigm, while guaranteeing confidentiality in the database as a service (DBaaS) paradigm is still an open research area.

DISADVANTAGES OF EXISTING SYSTEM:

Cannot apply fully homomorphic encryption schemes because of their excessive computational complexity.

PROPOSED SYSTEM:

We propose a novel architecture that integrates cloud database services with data confidentiality and the possibility of executing concurrent operations on encrypted data. This is the first solution supporting geographically distributed clients to connect directly to an encrypted cloud database, and to execute concurrent and independent operations including those modifying the database structure.

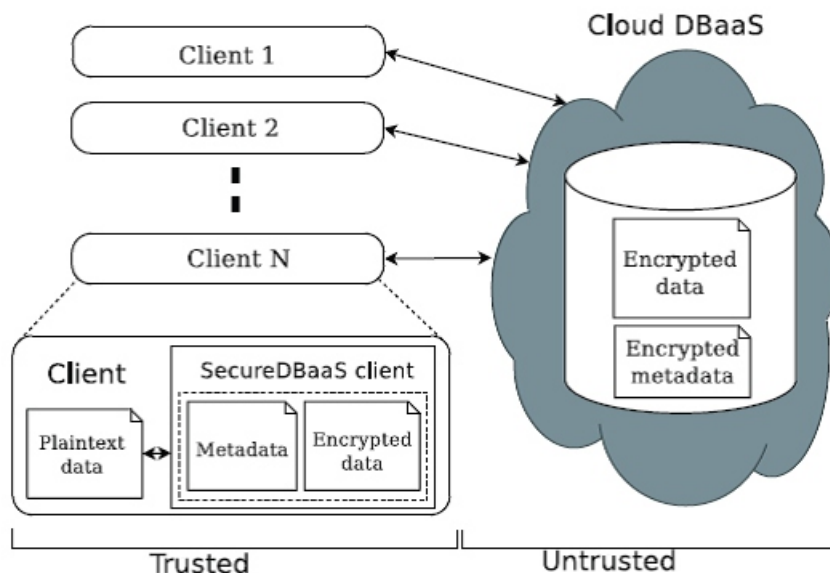
The proposed architecture has the further advantage of eliminating intermediate proxies that limit the elasticity, availability, and scalability properties that are intrinsic in cloud-based solutions. Secure DBaaS provides several original features that differentiate it from previous work in the field of security for remote database services.

ADVANTAGES OF PROPOSED SYSTEM:

The proposed architecture does not require modifications to the cloud database, and it is immediately applicable to existing cloud DBaaS, such as the experimented PostgreSQL Plus Cloud Database, Windows Azure and Xeround. There are no theoretical and practical limits to extend our solution to other platforms and to include new encryption algorithm.

It guarantees data confidentiality by allowing a cloud database server to execute concurrent SQL operations (not only read/write, but also modifications to the database structure) over encrypted data. It provides the same availability, elasticity, and scalability of the original cloud DBaaS because it does not require any intermediate server.

SYSTEM ARCHITECTURE:



SYSTEM STUDY: FEASIBILITY STUDY:

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY:

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY:

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

MODULES:

- 1.Setup Phase
- 2.Meta Data Module
- 3.Sequential SQL Operations
- 4.Concurrent SQL Operations

MODULES DESCRIPTION:

Setup Phase:

*We describe how to initialize a Secure DBaaS architecture from a cloud database service acquired by a tenant from a cloud provider.

*We assume that the DBA creates the metadata storage table that at the beginning contains just the database metadata, and not the table metadata.

*The DBA populates the database metadata through the Secure DBaaS client by using randomly generated encryption keys for any combinations of data types and encryption types, and stores them in the metadata storage table after encryption through the master key.

*Then, the DBA distributes the master key to the legitimate users. User access control policies are administrated by the DBA through some standard data control language as in any unencrypted database. In the following steps, the DBA creates the tables of the encrypted database.

Meta Data Module:

*In this module, we develop Meta data. So our system does not require a trusted broker or a trusted proxy because tenant data and metadata stored by the cloud database are always encrypted.

*In this module, we design such as Tenant data, data structures, and metadata must be encrypted before exiting from the client.

*The information managed by SecureDBaaS includes plaintext data, encrypted data, metadata, and encrypted metadata. Plaintext data consist of information that a tenant wants to store and process remotely in the cloud DBaaS.

*SecureDBaaS clients produce also a set of metadata consisting of information required to encrypt and decrypt data as well as other administration information. Even metadata are encrypted and stored in the cloud DBaaS.

Sequential SQL Operations:

*The first connection of the client with the cloud DBaaS is for authentication purposes. Secure DBaaS relies on standard authentication and authorization mechanisms provided by the original DBMS server. After the authentication, a user interacts with the cloud database through the Secure DBaaS client.

*Secure DBaaS analyzes the original operation to identify which tables are involved and to retrieve their metadata from the cloud database. The metadata are decrypted through the master key and their information is used to translate the original plain SQL into a query that operates on the encrypted database.

*Translated operations contain neither plaintext database (table and column names) nor plaintext tenant data. Nevertheless, they are valid SQL operations that the Secure DBaaS client can issue to the cloud database. Translated operations are then executed by the cloud database over the encrypted tenant data. As there is a one-to-one correspondence between plaintext tables and encrypted tables, it is possible to prevent a trusted database user from accessing or modifying some tenant data by granting limited privileges on some tables.

*User privileges can be managed directly by the untrusted and encrypted cloud database. The results of the translated query that includes encrypted tenant data and metadata are received by the Secure DBaaS client, decrypted, and delivered to the user. The complexity of the translation process depends on the type of SQL statement.

Concurrent SQL Operations:

*The support to concurrent execution of SQL statements issued by multiple independent (and possibly geographically distributed) clients is one of the most important benefits of Secure DBaaS with respect to state-of-the-art solutions.

*Our architecture must guarantee consistency among encrypted tenant data and encrypted metadata because corrupted or out-of-date metadata would prevent clients from decoding encrypted tenant data resulting in permanent data losses.

*A thorough analysis of the possible issues and solutions related to concurrent SQL operations on encrypted tenant data. Here, we remark the importance of distinguishing two classes of statements that are supported by Secure DBaaS: SQL operations not causing modifications to the database structure, such as read, write, and update; operations involving alterations of the database structure through creation, removal, and modification of database tables (data definition layer operators).

CONCLUSION:

We propose an innovative architecture that guarantees confidentiality of data stored in public cloud databases. Unlike state-of-the-art approaches, our solution does not rely on an intermediate proxy that we consider a single point of failure and a bottleneck limiting availability and scalability of typical cloud database services. A large part of the research includes solutions to support concurrent SQL operations (including statements modifying the database structure) on encrypted data issued by heterogeneous and possibly geographically dispersed clients. The proposed architecture does not require modifications to the cloud database, and it is immediately applicable to existing cloud DBaaS, such as the experimented PostgreSQL Plus Cloud Database [23], Windows Azure [24], and Xeround [22]. There are no theoretical and practical limits to extend our solution to other platforms and to include new encryption algorithms. It is worth observing that experimental results based on the TPC-C standard benchmark show that the performance impact of data encryption on response time becomes negligible because it is masked by network latencies that are typical of cloud scenarios. In particular, concurrent read and write operations that do not modify the structure of the encrypted database cause negligible overhead. Dynamic scenarios characterized by (possibly) concurrent modifications of the database structure are supported, but at the price of high computational costs. These performance results open the space to future improvements that we are investigating.

REFERENCES:

- [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.
- [4]J. Li, M. Krohn, D. Mazieres, and D. Shasha, "Secure Untrusted Data Repository (SUNDR)," Proc. Sixth USENIX Conf. Operating 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. 41st Ann. ACM Symp. Theory of Computing, 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.
- [11]E. Mykletun and G. Tsudik, "Aggregation Queries in the Database-as-a-Service Model," Proc. 20th Ann. IFIP WG 11.3 Working Conf. Data and Applications Security, July/Aug. 2006.
- [12]D. Agrawal, A.E. Abbadi, F. Emekci, and A. Metwalley, "Database Management as a Service: Challenges and Opportunities," Proc. 25th IEEE Int'l Conf. Data Eng., Mar.-Apr. 2009.
- [13]V. Ganapathy, D. Thomas, T. Feder, H. Garcia-Molina, and R. Motwani, "Distributing Data for Secure

Database Services,” Proc. Fourth ACM Int’l Workshop Privacy and Anonymity in the Information Soc., Mar. 2011.

[14]A. Shamir, “How to Share a Secret,” Comm. of the ACM, vol. 22, no. 11, pp. 612-613, 1979.

[15]M. Hadavi, E. Damiani, R. Jalili, S. Cimato, and Z. Ganjei, “AS5: A Secure Searchable Secret Sharing Scheme for Privacy Preserving Database Outsourcing,” Proc. Fifth Int’l Workshop Autonomous and Spontaneous Security, Sept. 2013.

[16]“Oracle Advanced Security,” Oracle Corporation, <http://www.oracle.com/technetwork/database/options/advanced-security>, Apr. 2013.

[17]G. Cattaneo, L. Catuogno, A.D. Sorbo, and P. Persiano, “The Design and Implementation of a Transparent Cryptographic File System For Unix,” Proc. FREENIX Track: 2001 USENIX Ann. Technical Conf., Apr. 2001.

[18]E. Damiani, S.D.C. Vimercati, S. Jajodia, S. Paraboschi, and P. Samarati, “Balancing Confidentiality and Efficiency in Untrusted Relational Dbmss,” Proc. Tenth ACM Conf. Computer and Comm. Security, Oct. 2003.

[19]L. Ferretti, M. Colajanni, and M. Marchetti, “Supporting Security and Consistency for Cloud Database,” Proc. Fourth Int’l Symp. Cyberspace Safety and Security, Dec. 2012.

[20]“Transaction Processing Performance Council,” TPC-C, <http://www.tpc.org>, Apr. 2013.

[21]H. Berenson, P. Bernstein, J. Gray, J. Melton, E. O’Neil, and P. O’Neil, “A Critique of Ansi Sql Isolation Levels,” Proc. ACM SIGMOD, June 1995.