REVIEW ON SECURED DATABASE AS A SERVICE ARCHITECTURE TO ACCESS ENCRYPTED CLOUD DATABASES

J. Sasi Kiran¹, L. Sunitha², A. Venkata Mani Kumar³, M. Naveen Kumar⁴
¹Professor, CSE Department, Vidya Vikas Institute of Technology, Chevella, Telangana
²Associate Professor, CSE Department, Vidya Vikas Institute of Technology, Chevella, Telangana
³M.Tech Scholar, Vidya Vikas Institute of Technology, Chevella, Telangana
⁴M.Tech Scholar, Vidya Vikas Engineering College, Chevella, Telangana

ABSTRACT:
This is a review report on architecture that integrates cloud database services with data confidentiality and the possibility of executing concurrent operations on encrypted data. This 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 inherent properties that are elasticity, availability, and scalability in cloud-based solutions.

Keywords: Database, Cloud, DBaaS, Secure DBaaS, SQL Operations

I. INTRODUCTION

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

II. DATABASE CLOUD SERVICE (DATABASE AS A SERVICE)

Placing the critical data over cloud infrastructures of un-trusted third parties, ensuring data confidentiality is major important. This requirement imposes clear data management choices: 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. To ensure 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) we propose Secure DBaaS as the first solution that allows cloud tenants to take full advantage of DBaaS qualities exposing unencrypted data to the cloud provider. The architecture design was motivated by a threefold goal: to allow multiple, independent, and geographically distributed clients to execute concurrent operations on encrypted data, including SQL statements that modify the database structure; to preserve data confidentiality and consistency at the client and cloud level; to eliminate any intermediate server between the cloud client and the cloud provider. The possibility of combining availability, elasticity, and scalability of a typical cloud DBaaS with data confidentiality is demonstrated through a prototype of Secure DBaaS that supports the execution of concurrent and independent operations to the remote encrypted database from many geographically distributed clients as in any unencrypted DBaaS setup. To achieve these goals, Secure DBaaS integrates existing cryptographic schemes, isolation mechanisms, and novel strategies for management of encrypted metadata on the untrusted cloud database. This paper is an overview on a theoretical discussion about solutions for data consistency issues due to concurrent and independent client accesses to encrypted data. In this context, we cannot apply fully homomorphic encryption schemes because of their complex computations. The Secure DBaaS architecture is designed to cloud platforms and does not introduce any intermediary proxy or broker server between the client and the cloud provider. Eliminating any trusted intermediate server allows Secure DBaaS to achieve the same availability, reliability, and elasticity levels of a cloud DBaaS. Other proposal based on intermediate server(s) were considered impracticable for a cloud-based solution because any proxy represents a single point of failure and a system bottleneck that limits the main benefits of a database service deployed on a cloud platform. Unlike Secure DBaaS, architectures relying on a trusted intermediate proxy do not support the most typical cloud scenario where geographically dispersed clients can concurrently issue read/write operations and data structure modifications to a cloud database.

A large set of experiments based on real cloud platforms demonstrate that SecureDBaaS is immediately applicable to any DBMS because it requires no modification to the cloud database services. Other studies where the proposed architecture is subject to the TPC-C standard benchmark for different numbers of clients and network latencies show that the performance of concurrent read and write operations not modifying the SecureDBaaS database structure is comparable to that of unencrypted cloud database. Workloads including modifications to the database structure are also supported by SecureDBaaS, but at the price of overheads that seem acceptable to achieve the desired level of data confidentiality. The motivation of these results is that network latencies, which are typical of cloud scenarios, tend to mask the performance costs of data encryption on response time. The overall conclusions of this paper are important because for the first time they demonstrate the applicability of encryption to cloud database services in terms of feasibility and performance.

III. SECURE DBAAS ARCHITECTURE

Secure DBaaS is designed to allow multiple and independent clients to connect directly to the untrusted cloud DBaaS without any intermediate server. Fig. 1 describes the overall architecture. We assume that a tenant organization acquires a cloud database service from an untrusted DBaaS provider. The tenant then deploys one or more machines (Client 1 through N) and installs a Secure DBaaS client on each of them.
This client allows a user to connect to the cloud DBaaS to administer it, to read and write data, and even to create and modify the database tables after creation. We assume the same security model that is commonly adopted by the literature in this field, where tenant users are trusted, and the network is un-trusted, and the cloud provider is honest-but-curious, that is, cloud service operations are executed correctly, but tenant information confidentiality is at risk. For these reasons, 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. To prevent an untrusted cloud provider from violating confidentiality of tenant data stored in plain form, SecureDBaaS adopts multiple cryptographic techniques to transform plaintext data into encrypted tenant data and encrypted tenant data structures because even the names of the tables and of their columns must be encrypted. 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. Secure DBaaS moves away from existing architectures that store just tenant data in the cloud database, and save metadata in the client machine or split metadata between the cloud database and a trusted proxy. When considering scenarios where multiple clients can access the same database concurrently, these previous solutions are quite inefficient. For example, saving metadata on the clients would require onerous mechanisms for metadata synchronization, and the practical impossibility of allowing multiple clients to access cloud database services independently. Solutions based on a trusted proxy are more feasible, but they introduce a system bottleneck that reduces availability, elasticity, and scalability of cloud database services.

Secure DBaaS proposes a different approach where all data and metadata are stored in the cloud database. Secure DBaaS clients can retrieve the necessary metadata from the untrusted database through SQL statements, so that multiple instances of the Secure DBaaS client can access to the untrusted cloud database independently with the guarantee of the same availability and scalability properties of typical cloud DBaaS. Encryption strategies for tenant data and innovative solutions for metadata management and storage.

IV. ARCHITECTURE DESIGN IMPLEMENTATION

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

4.1 Setup Phase: We describe how to initialize 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.

4.2 Meta Data: In this phase, 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 phase, we design such as Tenant data, data structures, and metadata must be encrypted before exiting from the client.

The information managed by Secure DBaaS 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. Secure DBaaS 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.

4.3 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 SecureDBaaS 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 un-trusted and encrypted cloud database. The results of the translated query that includes encrypted tenant data and metadata are received by the SecureDBaaS client, decrypted, and delivered to the user. The complexity of the translation process depends on the type of SQL statement.

4.4. 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 SecureDBaaS 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)

V. CONCLUSION

This is a review on 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 architecture does not require modifications to the cloud database, and it is immediately applicable to existing cloud DBaaS.

We are explored 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. In particular, concurrent read and writes 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.

VI. ACKNOWLEDGEMENTS

I would like to express my cordial thanks to Sri. CA. BashaMohiuddin, Chairman, Smt. Rizwana Begum-Secretary and Sri. Touseef Ahmed-Vice Chairman, Dr.M.Anwarullah, Principal - Farah Group of Institutions, Hyderabad for providing moral support, encouragement and advanced research facilities. Authors would like to thank the anonymous reviewers for their valuable comments. And they would like to thank Dr.V. Vijaya Kumar, Anurag Group of Institutions for his invaluable suggestions and constant encouragement that led to improvise the presentation quality of this paper.

VII. REFERENCES


AUTHOR PROFILES

DR. J. SASI KIRAN graduated in B.Tech [EIE] from JNTU Hyd. He received Masters Degree in M.Tech [CSE] from JNT University, Hyderabad. He received Ph.D degree in Computer Science from University of Mysore, Mysore. At Present he is working as Professor in CSE and Dean – Academics in Vidya Vikas Institute of Technology, Chevella, R.R. Dist Telangana State, India. His research interests include Image Processing, Data Mining and Network Security. He has published 39 research papers till now in various National, International Conferences, Proceedings and Journals. He has received best Teacher award twice from Vidya Group, Significant Contribution award from Computer Society of India and Passionate Researcher Trophy from Sri. Ramanujan Research Forum, GIET, Rajuhmundry, A.P, India.

L. SUNITHA graduated from kakatiya university Warangal received her M.Tech in Computer Science and Engineering from JNTU, Hyderabad in 2009. She has 15 years of teaching experience, presently she is working as Associate Professor in Vidya Vikas Institute of Technology Chevella, R.R. Dist Telangana State, India and also pursuing PhD in Computer Science and Engineering from JNTU Hyderabad, India. She has received best Teacher award from Vidya Group. Her area of specialization is Data mining. Member in CSI and IAENG, she has published many papers in various National, International Conferences, Proceedings and Journals.


Address for Correspondence

L. SUNITHA
Qr.No:C-6 ,BDL TOWN SHIP
BHANUR,PANTANANCHERU(Mondal)
MEDAK (dist),PIN :502305,TELANGANA
Mobile:9441264836