ISSN 2348 – 9928
IJAICT Volume 5, Issue 4, April 2018 Doi:01.0401/ijaict.2015.11.07 Published on 05 (04) 2018

DISTRIBUTED AND CONCURRENT OPERATIONS ON ENCRYPTED CLOUD

A.Nelson ***PG Scholar***
S.Ayyasamy ***PG Scholar***
P.Sujatha ***PG Scholar***
V.Yuvaraj ***Teaching Assistant***

Computer Science and Engineering,
Anna University Regional Centre,
Coimbatore, Tamilnadu, India.

Abstract—Cloud computing is computing in which large groups of remote servers are networked to allow centralized data storage and online access to computer services or resources. Data sharing is an important functionality in cloud storage. Cloud storage is a model of data storage where the digital data is stored in logical pools, the physical storage spans across multiple servers (and often locations), and the physical environment is typically owned and managed by a hosting company. 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. A novel architecture is proposed that integrates cloud database services with data confidentiality and the possibility of executing concurrent operations on encrypted data. This supports geographically distributed clients to connect directly to an encrypted cloud database, and to execute concurrent and independent operations including those modifying the database structure. This eliminates intermediate proxies that limit the elasticity, availability, and scalability properties that are intrinsic in cloud-based solutions

Keywords— Cloud, Security, Confidentiality, Securedbaas, Database.

I. INTRODUCTION

Cloud computing has been envisioned as the next generation information technology (IT) architecture for enterprises, due to its long list of unprecedented advantages in the IT history: on-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing and transference of risk. As a disruptive technology with profound implications, cloud computing is transforming the very nature of how businesses use information technology. One fundamental aspect of this paradigm shifting is that data are being centralized or outsourced to the cloud. From users' perspective, including both individuals and IT enterprises, storing data remotely to the cloud in a flexible on-demand manner brings appealing benefits relief of the burden for storage management, universal data access with location independence, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc.

Cloud computing is mostly used to sell hosted services in the sense of application service provisioning that run client server software at a remote location. Such services are given popular acronyms like 'SaaS' (Software as a Service), 'PaaS' (Platform as a Service), 'IaaS' (Infrastructure as a Service), 'HaaS' (Hardware as a Service) and finally 'EaaS' (Everything as a Service). End users access cloud-based applications through a web browser, thin client or mobile app while the business software and user's data are stored on servers at a remote location.

Cloud services means services made available to users on demand via the Internet from a cloud computing provider's servers as opposed to being provided from a company's own on-premises servers. SaaS is a software distribution model in which applications are hosted by a vendor or service provider and made available to customers over a network, typically the Internet. PaaS refers to the delivery of operating systems and associated services over the Internet without downloads or installation. IaaS involves outsourcing the equipment used to support operations, including storage, hardware, servers and networking components, all of which are made accessible over a network.

II. RELATED WORK

SecureDBaaS provides several original features that differentiate it from previous work in field of security for remote database services.

- It guarantees data confidentiality by allowing a cloud database server to execute concurrent SQL over encrypted data.
- It provides the same availability, elasticity, and scalability of the original cloud DBaaS because it does

not require any intermediate server. Response times are affected by cryptographic overheads that for most SQL operations are masked by network latencies.

- Multiple clients, possibly geographically distributed, can access concurrently and independently a cloud database service.

- It does not require a trusted broker or a trusted proxy because tenant data and metadata stored by the cloud database are always encrypted.

- It is compatible with the most popular relational database servers, and it is applicable to different DBMS implementations because all adopted solutions are database agnostic.

Different approaches guarantee some confidentiality (e.g., [1], [2]) by distributing data among different providers and by taking advantage of secret sharing [3]. one cloud provider to read its portion of data, but information can be reconstructed by colluding cloud providers. A step forward is proposed in [4], that makes it possible to execute range queries on data and to be robust against collusive providers. SecureDBaaS differs from these solutions as it does not require the use of multiple cloud providers, and makes use of SQL-aware encryption algorithms to support the execution of most common SQL operations on encrypted data.

SecureDBaaS relates more closely to works using encryption to protect data managed by untrusted databases. In such a case, a main issue to address is that cryptographic techniques cannot be naı¨vely applied to standard DBaaS because DBMS can only execute SQL operations over plaintext data.

Some DBMS engines offer the possibility of encrypting data at the file system level through the so-called Transparent Data Encryption feature [5], [6]. This feature makes it possible to build a trusted DBMS over untrusted storage. However, the DBMS is trusted and decrypts data before their use. Hence, this approach is not applicable to the DBaaS context considered by SecureDBaas, because we assume that the cloud provider is untrusted such as [7], allow the execution of operations over encrypted data.

These approaches preserve data confidentiality in scenarios where the DBMS is not trusted; however, they require a modified DBMS engine and are not compatible with DBMS software (both commercial and open source) used by cloud providers.

On the other hand, SecureDBaaS is compatible with standard DBMS engines, and allows tenants to build secure cloud databases by leveraging cloud DBaaS services already available. For this reason, SecureDBaaS is more related to [8] and [9] that preserve data confidentiality in untrusted DBMSs through encryption techniques, allow the execution of SQL operations over encrypted data, and are compatible with common DBMS engines. However, the architecture of these solutions is based on an intermediate and trusted proxy that mediates any interaction between each client and the untrusted DBMS server. The approach proposed in [8] by the authors of the DBaaS model [10] works by encrypting blocks of data instead of each data item. Whenever a data item that belongs to a block is required, the trusted proxy needs to retrieve the whole block, to decrypt it, and to filter out unnecessary data that belong to the same block.

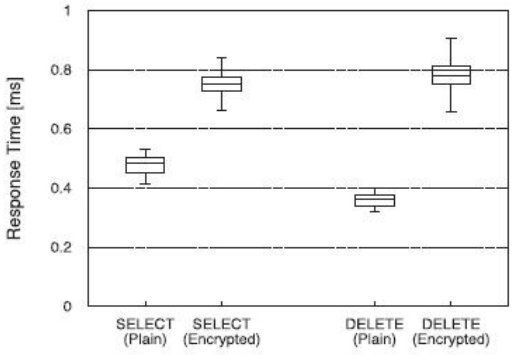
As a consequence, this design choice requires heavy modifications of the original SQL operations produced by each client, thus causing significant overheads on both the DBMS server and the trusted proxy. Other works [11], [12] introduce optimization and generalization that extend the subset of SQL operators supported by [8], but they share the same proxy-based architecture and its intrinsic issues. On the other hand, SecureDBaaS allows the execution of operations over encrypted data through SQL-aware encryption algorithms.

This technique, initially proposed in CryptDB [9], makes it possible to execute operations over encrypted data that are similar to operations over plaintext data. In many cases, the query plan executed by the DBMS for encrypted and plaintext data is the same. A proxy-based architecture requiring that any client operation should pass through one intermediate server is not suitable to cloud-based scenarios, in which multiple clients, typically distributed among different locations, need concurrent access to data stored in the same DBMS. On the other hand, SecureDBaaS supports distributed clients issuing independent and concurrent SQL operations to the same database and possibly to the same data.

III. OPERATIONS

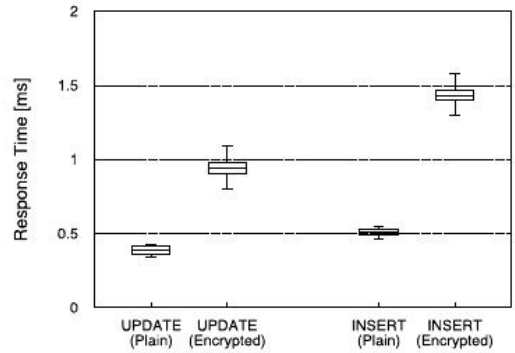
In this phase, Plain name: the name of the corresponding column of the plaintext table. Coded name: the name of the column of the secure table. This is the only information that links a column to the corresponding plaintext column because column names of secure tables are randomly generated.

Secure type: the secure type of the column, as defined in Section This allows a SecureDBaaS client to be informed about the data type and the encryption policies associated with a column.



Plain versus encrypted SELECT and DELETE operations.

Encryption key: the key used to encrypt and decrypt all the data stored in the column. SecureDBaaS stores metadata in the metadata storage table that is located in the untrusted cloud as the database. This is an original choice that augments flexibility, but opens two novel issues in terms of efficient data retrieval and data confidentiality.



Plain versus encrypted UPDATE and INSERT operations.

IV. SYSTEM MODULES

4.1 User Registration

This module is for registration of users, if any new member wants to enter, he want to register into the system. Users can use the files in the cloud storage space. Users can download the files which are stored by the data owner. Before downloading the file, the user must be authenticated. User must get the secret key from the admin for file decryption.

4.2 Admin Module

Data owner has a large amount of data to be stored in the cloud. Cloud service provider provides data storage service and has enough storage spaces and computation resources.

Data owner can upload the files after encrypting the files. These files are stored in the cloud storage space.

4.3 Data Encryption

Resource scheduling process is initiated in the cloud server for the training process. Back Propagation Neural network (BPN) algorithm is used for the training process. Random sharing algorithm is used in the data splitting process to secure the intermediate data values. Training process results are redirected to the data provider.

4.4 Accessing Controlling

Trained data values are collected from the cloud server. Data provider decrypts the trained data values. Data encryption/decryption tasks are carried out using secure scalar product and addition mechanism. Test data values are compared with the trained data values for the class assignment process.

4.5 User Revocation

This module is for revocation of users, it removes the user from the particular user group. The revocation of user takes place, remove access of files in cloud for the particular user without disturbing the other users in that group. User revocation is performed by the group manager through a public available revocation list (RL), based on which group members can encrypt their data files and ensure the confidentiality against the revoked users.

4.6 Keyword Search

The data user is authorized to process multi-keyword retrieval over the outsourced data. The computing power on the user side is limited, which means that operations on the user side should be simplified. The authorized data user at first generates a query. For privacy consideration, which keywords the data user has searched must be concealed. Thus, the data user encrypts the query and sends it to the cloud server that returns the relevant files to the data user. Afterward, the data user can decrypt and make use of the files.

V. ARCHITECTURE DESIGN

In this project is more then used for cloud computing system. I have used in five system modules. In this first section User can send to request for accessing cloud platform send to cloud provider. In cloud provider analysis and verify the user request form. Once this request form is correct then provider responds to user. At the same time provider providing the security key for accessing cloud. User accessing the file system.

User Each an every time files are update and delete files user must be enter in security key after the files are Encryption/decryption store in cloud Database. The Files are encryption/decryption using MD5 algorithms. Finally the Files are stored in Cloud storage. In this system are Multiple,

independent, and geographically distributed clients to execute concurrent operations on encrypted data. This preserves data confidentiality and consistency at the client and cloud level. It eliminates any intermediate server between the cloud client and the cloud provider. This supports the execution of concurrent and independent operations to the remote encrypted database from many geographically distributed.

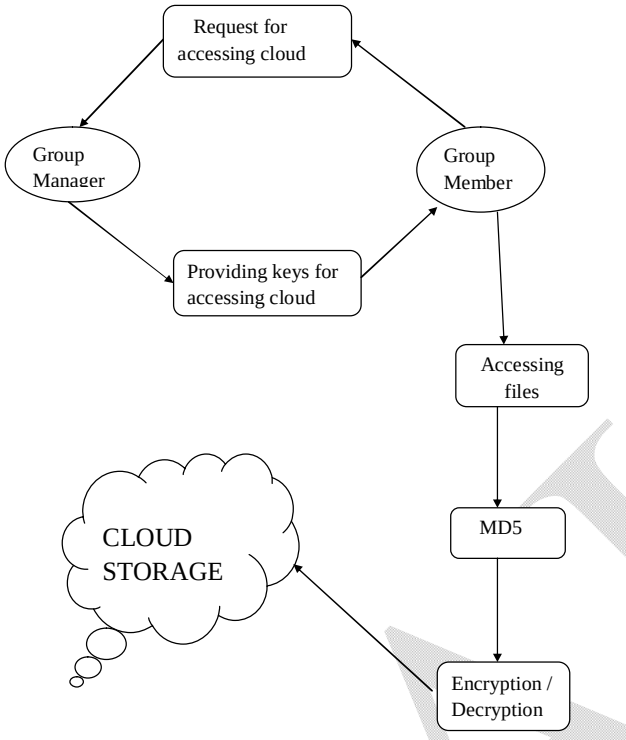


Fig 3: System flow of Proposed System

4.7 Advantages

- It guarantees data confidentiality by allowing a cloud database server to execute concurrent SQL operations over encrypted data.
- It provides the same availability, elasticity, and scalability of the original cloud DBaaS because it does not require any intermediate server.
- It does not require a trusted broker.
- It is compatible with the most popular relational database servers, and it is applicable to different DBMS implementations.

VI. CONCLUSION

Proposed system an innovative architecture is proposed that guarantees confidentiality of data stored in public cloud databases. Unlike state-of-the-art approaches, the proposed 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 operations. This scheme is that MD5 algorithm can be used for large part of the research includes solutions to support concurrent operations. so it will be helpful for user revocation. The authorized data user at first generates a query. For privacy consideration, which keywords the data user has searched must be concealed.

References

[1] D. Agrawal, A.E. Abbadi, F. Emekci, and A. Metwally, "Database Management as a Service: Challenges and Opportunities," Proc. 25th IEEE Int'l Conf. Data Eng., Mar.-Apr. 2009.

[2] 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.

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

[4] 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.

[5] "Oracle Advanced Security," Oracle Corporation,http://www.oracle.com/technetwork/database/options/advanced-security,Apr.13.

[6] 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.

[7] 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.

[8] 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.

[9] 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

[10] H. Hacigu¨mu¨ s¸, B. Iyer, and S. Mehrotra, "Providing Database as a Service," Proc. 18th IEEE Int'l Conf. Data Eng., Feb. 2002.

[11] 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.

[12] 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.