Volume 6, Issue 1, January 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Secure Architecture Using Multiuser Key Distribution for Cloud Database

Anjali Nayak, Dr. Sadhna K Mishra
Dept. of CSE, LNCTS, Bhopal,
M.P. India

Abstract- Cloud computing is a computing in which large groups of remote servers are networked to allow centralized data storage and online access to computer services or resources. The cloud aims to cut costs, and help the users focus on their core business instead of being impeded by IT obstacles. Cloud storage, Data as a service (DaaS) and Database as a service (DBaaS) are the different terms used for data management in the cloud. They differ on the basis of how data is stored and managed. Cloud storage is a virtual storage that enables users to store documents and objects. Cloud database should support features of cloud computing as well as of traditional databases for wider acceptability. The potential challenges associated with cloud database are scalability, high availability and fault tolerance, data consistency and integrity, confidentiality and many more. We propose a secure architecture using encryption and multiuser key distribution for cloud database.

Keywords-Cloud database, adaptive encryption, multiuser key.

I. INTRODUCTION

1.1 Overview

Cloud computing[1] can be defined as new computing that has focus on both industry and academia. Cloud computing is the result of evolution and adoption of existing technologies and paradigms. The goal of cloud computing is to allow users to take benefit from all of these technologies, without the need for deep knowledge about or expertise with each one of them. Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources(e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. A style of computing where massively scalable IT-related[2] capabilities are provided 'as a service' using Internet technologies to multiple external customers. At its simplest, cloud computing is the dynamic delivery of information technology resources and capabilities as a service over the Internet. This cloud model is composed of five essential characteristics, three service models, and four deployment models[1].Massive growth in digital data, changing data storage requirements, better broadband facilities and Cloud computing led to the emergence of cloud databases .Cloud Storage, Data as a service (DaaS) and Database as a service (DBaaS) are the different terms used for data management in the Cloud. They differ on the basis of how data is stored and managed. Cloud storage is virtual storage that enables users to store documents and objects. Dropbox, iCloud etc. are popularcloud storage services. DaaS allows user to store data at a remote disk available through Internet.Cloud database is designed for virtualized computer environment. It is not as simple as taking relational database and deploying it over a cloud server.[3] Cloud database as a service has to fulfill all the characteristics of relational database as well as cloud database. Cloud storage cannot work without basic data management services. So,these two terms are used interchangeably.DBaaS[4] isonestep ahead. It offers complete database functionality and allows users to access and store their database at remote disks anytime from any place through Internet. Amazon's SimpleDB, Amazon RDS, Google's BigTable, Yahoo's Sherpa and Microsoft's SQL Azure Database are the commonly used databases in the Cloud [2]. The cloud will offer all of us amazing flexibility as we can specify the exact amount of computing power, data, or applications we need for each task we are working on.

The data should be kept secured and should not be exposed to anyone at any cost. Confidentiality of data is another security issue associated with cloud computing. The different security issues in cloud are scalability, heterogeneity, Data Intrusion, Data Integrity, Non- Repudiation, Confidentiality, access control, authentication and authorization.

The major issue with the cloud database is that it requires a very high level security. For this previously adaptive encryption scheme is given using a single master key. Now in the new proposed work a multi-user key distribution is supported that will ensure more confidentiality and security in the cloud database. The architecture will support two keys a master key and a multi-user key. The metadata structure is also supported in the proposed work. Cloud architecture will be developed for this proposed work using java as a platform. The main concept of this proposed work is cloud database as a service (DBaaS). There are two main tenant concerns that may prevent the adoption of the cloud: data confidentiality and cost. The proposed work will address both these issues with multi-user key distribution. We have performed experiments to validate the privacy, accuracy, and efficiency of our solutions.

1.2 Objective

The objective of work is as follows
To improve the security of cloud database
To improve the performance of database as a service in cloud computing
To decrease the cost of the cloud database.

1.3 Motivation

The cloud database as a service is a novel paradigm that can support several Internet-based applications, but its adoption requires the solution of information confidentiality problems. The different security issues in cloud are scalability, heterogeneity, Data Intrusion, Data Integrity, Non- Repudiation, Confidentiality, access control, authentication and authorization. Other encryption schemes that allow the execution of SQL operations over encrypted data either have performance limits or require the choice of which encryption scheme must be adopted for each database column and SQL operation. Most of the technique concerning encryption for cloud-based services are inapplicable to the database paradigm. The number of leaked sensitive data records has increased dramatically during the last few years. Detecting and preventing data privacy[5] requires a set of complementary solutions, which may include data-privacy detection, data detention, stealthy malware detection, and policy enforcement. Deliberately planned attacks, inadvertent leaks such as forwarding confidential emails to unclassified email account, and human mistakes such as assigning the wrong privilege) lead to most of the data-privacy incidents. Database as a service (DBaaS) that poses several research challenges in terms of security and cost evaluation from a tenant's point of view.

1.4 Problem Statement

The paper focuses on the following problem:
How to design and develop secure architecture for tenant concerns that may prevent the adoption of the cloud as data confidentiality and costs for multi-user key distribution schemes and under different threat model premises. How to improve the performance of database as a service in cloud computing.
In the context of the problem, the two key participants are (1) DaaS providers and (2) DaaS customers (3) Daas Cost. We are interested in the database as a service paradigm (DBaaS) [5] that poses several research challenges in terms of security and cost evaluation from a tenant's point of view. Improving the confidentiality of information stored in cloud databases represents an important contribution to the adoption of the cloud. Most of the technique concerning encryption for cloud-based services are inapplicable to the database paradigm. By having multi-user key cloud database can ensure more security and confidentiality and an improved performance can be achieved in terms of encryption.
The rest of the thesis is organized as follows.
Section 2 provides the background, related workand literature review relevant for the context. Section 3 provides the proposed methodology, proposed algorithm and description of proposed methodology. Section 4 represents the implementation of proposed methodology, discussion on simulation Results and performance analysis of simulation results. Section 5 concludes the thesis with a summary of the main findings concluding remarks, limitation discussion and an outlook on future research directions.

II. RELATED WORK

[1] are interested in the database as a service paradigm that poses several research challenges in terms of security and cost evaluation from tenant's point of view. Most results concerning encryption for cloud database services are inapplicable to the database paradigm. Other encryption schemes that allow the execution of SQL operations over encrypted data either have performance limits or require the choice of which encryption scheme must be adopted for each database column and SQL operation. These proposals are fine when the set of queries can be statically determined at design time, while we are interested in other common scenarios where the workload may change after the database design. [1] a novel architecture for adaptive encryption of public cloud databases is proposed that offers a proxy-free alternative to the system described. This proposed architecture guarantees in an adaptive way the best level of data confidentiality for any database workload, even when the set of SQL queries dynamically changes. The adaptive encryption scheme, which was initially proposed for applications not referring to the cloud, encrypts each plain column to multiple encrypted columns, and each value is encapsulated in different layers of encryption, so that outer layers guarantee higher confidentiality but support fewer computation capabilities with respect to the inner layers. The outer layers are dynamically adapted at runtime when new SQL operations are added to the workload. In previous work [6], the first proxy-free architecture for adaptive encryption of cloud databases that does not limit the availability, elasticity and scalability of a plain cloud database because multiple clients can issue concurrent operations without passing through some centralized component as in alternative architectures. Although data encryption seems the most intuitive solution for confidentiality, its application to cloud databases services is not trivial, because the cloud database must be able to execute SQL operations directly over encrypted data without accessing any decryption key. Naive solutions encrypt the whole database with some standard encryption algorithm that does not allow executing any SQL operation directly on the cloud. As the consequence, the tenant has two alternatives: download the entire database, decrypt temporarily the cloud database, execute the query and, if the operation modifies the database, encrypt and upload the new data, decrypt temporarily the cloud database, execute the query, and re-encrypt it. The former solution is affected by huge communication and computation overheads, and consequent costs that would make cloud database services quite inconvenient, the later solution does not guarantee data confidentiality because the cloud provider obtains decryption

keys. The right alternative is to execute SQL operations directly on the cloud database, without giving decryption keys to the provider. Many algorithms were proposed for this work like aggregation technique, fully homomorphism encryption etc. The drawback related to these feasible encryption algorithms is that in a medium-long term horizon, the database operations will be required over each database column. A solution to these problems were then given , the proposed architecture allows multiple clients to issue concurrent SQL operations to an encrypted database without any intermediate trusted server, but it assumes that the set of SQL operations do not change after the database design. The [5] develops the initial design through a prototype implementation, novel experimental results and an original cost model.

ADAPTIVE ENCRYPTION SCHEME

The [1] consider SQL-aware encryption algorithms that guarantee data confidentiality and allow the cloud database engine to execute SQL operations over encrypted data. As each algorithm supports a specific subset of SQL operators, we refer to the following encryption schemes.

*Random (Rand):*It is the most secure encryption because it does not reveal any information about the original plain value. It does not support any SQL operator, and it is used only for data retrieval.

Deterministic (Det): It deterministically encrypts data, so that equality of plaintext data is preserved. It supports the equality operator.

Order Preserving Encryption (Ope) : It preserves in the encrypted values the numerical order of the original unencrypted data. It supports the comparison SQL operators (=,<,<=,>,>=).

Homomorphic Sum (Sum): It is homomorphic with respect to the sum operation, so that the multiplication of encrypted integers is equal to the sum of plaintext integers. It supports the sum operator between integer values.

Search: It supports equality check on full strings (i.e., the LIKE operator).

Plain: It does not encrypt data, but it is useful to support all SQL operators on non confidential data.

If each column of the database was encrypted with only one algorithm, then the database administrator would have to decide at design time which operations must be supported on each database column. However, this solution is impractical for scenarios in which the database workload changes over time.The proposed system in [7] supports adaptive encryption for public cloud database services, where distributed and concurrent clients can issue direct SQL operations. By avoiding an architecture based on intermediate servers between the clients and the cloud database, the proposed solution guarantees the same level of scalability[8] and availability of the cloud service. Fig.1 [1] shows a scheme of the proposed architecture where each client executes an encryption engine that manages encryption operations. This software module is accessed by external user applications through the encrypted database interface. The proposed architecture manages five

Types of information:

1. Plain data represent the tenant information;
2. Encrypted data are the encrypted version of the plain data, and are stored in the cloud database;
3. Plain metadata represent the additional information that is necessary to execute SQL operations on encrypted data;
4. Encrypted metadata are the encrypted version of the plain metadata, and are stored in the cloud database.
5. Master key is the encryption key of the encrypted metadata, and is known by legitimate clients. All data and metadata stored in the cloud database are encrypted. Any application running on a legitimate client can transparently issue SQL operations[9] (e.g., SELECT, INSERT, UPDATE and DELETE) to the encrypted cloud database through the encrypted database interface. Data transferred between the user application and the encryption engines are not encrypted, whereas information is always encrypted before sending it to the cloud database. When an

Application issues a new SQL operation, the encrypted database interface contacts the encryption engine[10]that retrieves the encrypted metadata and decrypts them with the masterkey.

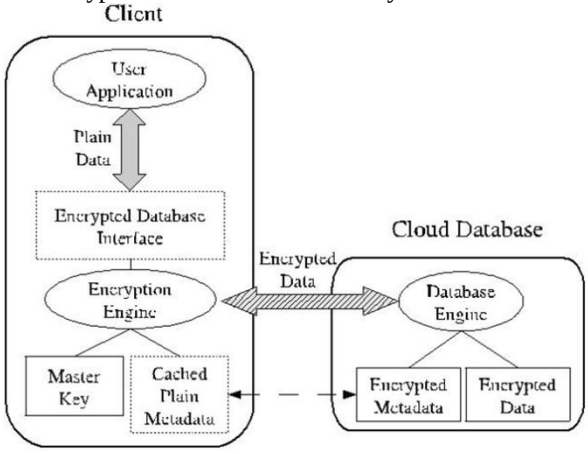

Figure 1 Encrypted cloud database

To improve performance, the plain metadata are cached locally by the client. After obtaining the metadata, the encryption engine is able to issue encrypted SQL statements to the cloud database, and then to decrypt the results. The results are returned to the user application through the encrypted database interface. As in related literature, the proposed

architecture guarantees data confidentiality in a security model in which: the network is untrusted, tenant users are trusted, that is, they do not reveal information about plain data, plain metadata, and the master key; the cloud provider administrators are defined semi-honest or honest-but-curious, that is, they do not modify tenant's data and results of SQL operations, but they may access tenant's information stored in the cloud database. The remaining part of this section describes the adaptive encryption schemes, the encrypted metadata stored in the cloud database, and the main operations for the management of the encrypted cloud database.

III. PROPOSED WORK

In the previous works using adaptive encryption scheme there is used master key only. The multi-user key distribution architecture is not supported by the previous work. Here multi-user key distribution is proposed for the cloud database which will use the adaptive encryption algorithm proposed in the [5] with some modifications. Encryption scheme in database has performance limits and different technique for different SQL operation. Adaptive encryption scheme encrypts each plain column to multiple encrypted columns and each value is encapsulated in different layers of encryption so that the outer layers guarantee higher confidentiality but support fewer computations capabilities with respect to inner layers.Figure 3 is the proposed architecture for the multi-user key distribution in cloud database. The algorithm that will be designed for implementation will consist the following steps-.

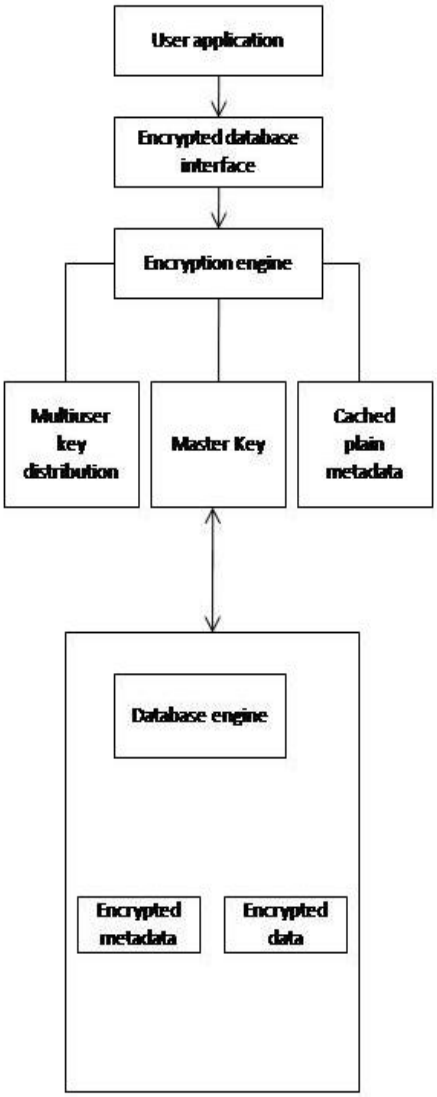

Figure 2 Proposed architecture

1. Master key generation.
2. Multi-user key generation.
3. Multi-user key distribution.
4. Reducing cost using cost pricing model.

By having multi-user key cloud database can ensure more security and confidentiality and an improved performance can be achieved in terms of encryption. Metadata concept used in the [5] is also considered in the proposed work. After generating the multi-user key its distribution is also done in the algorithm. A cost model can also be derived which will tell the mathematical representation of the algorithm performance.

IV. IMPLEMENTATION

Java platform is used for the implementation of the algorithm and Oracle 11g server is used as the back-end. Windows operating system is considered good in the security point of view.The experiments are carried out in lab, which provides us with a set of machines in a controlled environment. Each client machine runs the Java client prototype of our architecture on a Intel PIV machine having a single 3 GHz processor, 2 GB of RAM and two 7200 RPM 500 GB SCSI disks. The database server is Oracle 11g running on Intel machine having a PIV 2.4 GHz processor, 4GB of RAM and a 7,200 RPM 500 GB SATA disk.

Our proposed architecture supports insert, update, select, and delete operations on encrypted cloud database.The data is stored in the encrypted form and desired operations also performed on encrypted data to provide higher level of security. The SQL operations are performed on both simple and encrypted database and results and performance are compared. Our system will work on both the situations.

V. CONCLUSION

Cloud computing is a general term for the delivery of hosted services over the internet. Cloud computing enables companies to consume the resources and compute their utility, similar to electricity or a telephone service rather than building and maintaining computing infrastructure. A cloud database is a database that has been optimized or built for a virtualized computing environment. Putting the database in cloud can be an effective way to support and cloud enable business applications as a part of a wider software as a service. The major issue with the cloud database is that it requires a very high level security. For this previously adaptive encryption scheme is given using a single master key. Now in the new proposed work a multi-user key distribution is supported that will ensure more confidentiality and security in the cloud database. The architecture will support two keys a master key and a multi-user key. The metadata structure is also supported in the proposed work. Cloud architecture will be developed for this proposed work using java as a platform. The main concept of this proposed work is cloud database as a service (DBaaS). There are two main tenant concerns that may prevent the adoption of the cloud: data confidentiality and cost. The proposed work will address both these issues with multi-user key distribution. We have performed experiments to validate the privacy, accuracy, and efficiency of our solutions.

A malicious insider or a piece of program may steal sensitive personal or organizational data from a host. Because the malicious adversary can use strong private encryption, steganography or covert channels to disable content-based traffic inspection. Host-based defenses such as detecting the infection onset need to be deployed instead. We plan to focus on designing a host-assisted mechanism for the complete data-leak detection for large-scale organizations.

REFERENCES

[1] Luca Ferretti, Fabio Pierazzi, Michele Colajanni, Micro Marachetti, “Performance and Cost Evaluation of an Adaptive Encryption Architecture for Cloud Database”, IEEE Transactions on Cloud Computing, Vol 2, No 2, 2014, pp. 143-155

[2] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic,“Cloud computing and emerging it platforms: Vision, hype, andreality for delivering computing as the 5th utility,” Future GenerationComput. Syst., vol. 25, no. 6, pp. 599–616, 2009.

[3] T. Mather, S. Kumaraswamy, and S. Latif, Cloud Security and Privacy:An Enterprise Perspective on Risks and Compliance. Sebastopol,CA, USA: O’Reilly Media, Inc., 2009.

[4] H.-L. Truong and S. Dustdar, “Composable cost estimation andmonitoring for computational applications in cloud computingenvironments,” ProcediaComput. Sci., vol. 1, no. 1, pp. 2175–2184, 2010.

[5] E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good, “Thecost of doing science on the cloud: The montage example,” inProc. ACM/IEEE Conf. Supercomputing, 2008, pp. 1–12.

[6] H. Hacig€um€u¸s, B. Iyer, and S. Mehrotra, “Providing databaseas a service,” in Proc. 18th IEEE Int. Conf. Data Eng., Feb. 2002,pp. 29–38.

[7] G. Wang, Q. Liu, and J. Wu, “Hierarchical attribute-based encryptionfor fine-grained access control in cloud storage services,” inProc. 17th ACM Conf. Comput. Commun. Security, 2010, pp. 735–737.

[8] R. A. Popa, C. M. S. Redfield, N. Zeldovich, and H. Balakrishnan,“CryptDB: Protecting confidentiality with encrypted query processing,”in Proc. 23rd ACM Symp. Operating Systems Principles, Oct.2011, pp. 85–100.

[9] C. Gentry, “Fully homomorphic encryption using ideallattices,” in Proc. 41st ACM Symp. Theory Comput., May. 2009,pp. 169–178.

[10] A. Boldyreva, N. Chenette, and A. O’Neill, “Order-preservingencryption revisited: Improved security analysis and alternativesolutions,” in Proc. 31st Annu. Int. Conf. Adv. Cryptology, Aug. 2011, pp. 578–595.

