ISSN:2348-2079

Volume-6 Issue-2

International Journal of Intellectual Advancements and Research in Engineering Computations

Performance and Cost Evaluation of an Distribution Adaptive Multilevel Encryption Architecture for Cloud

[1]Mr. C. Mani M.C.A., M.Phil., M.E., Associate Proffessor,
[2]Mr. P.Anbalagan Final MCA.,
Department of MCA, Nandha Engineering College (Autonomous), Erode-52.
E-Mail ID: cmanimca@gmail.com, pganbalagan@gmail.com

***Abstract*-Database as a service paradigm (DBaaS) poses several research challenges in terms of security and cost evaluation from a tenant's point of view. 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. This thesis proposes a novel architecture for adaptive encryption of public cloud databases that offers an interesting alternative to the tradeoff between the required data confidentiality level and the flexibility of the cloud database structures at design time.**

This research proposes a novel architecture for adaptive encryption of public cloud databases that offers a proxy-free alternative to the system. The proposed system demonstrates the feasibility and performance of the proposed solution through a software prototype. The proposed architecture manages five types of information: plain data represent the tenant information, encrypted data are the encrypted version of the plain data, and are stored in the cloud database, plain metadata represent the additional information that is necessary to execute SQL operations on encrypted data; encrypted metadata are the encrypted version of the plain metadata, and are stored in the cloud database; master key is the encryption key of the encrypted metadata, and is known by legitimate clients.

The propose a model and a methodology that allow a tenant to estimate the costs of plain and encrypted cloud database services even in the case of workload and cloud price variations in a medium-term horizon. By applying the model to actual cloud provider prices, we can determine the encryption and adaptive encryption costs for data confidentiality. Future research could evaluate the proposed or alternative architectures for multi-user key distribution schemes and under different threat model hypotheses.

***Keywords*:SLA,ICT,SOL,VMs**

I. INTRODUCTION

Improving the confidentiality of information stored in cloud databases represents an important contribution to the adoption of the cloud as the fifth utility because it addresses most user concerns. Our proposal is characterized by two main contributions to the state of the art: architecture and cost model.

Data encryption seems the most intuitive solution for confidentiality, its application to cloud database services is not trivial, because the cloud database must be able to execute SQL operations directly over encrypted data without accessing any decryption key. the tenant has two alternatives: download the entire database, decrypt it, 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 latter 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. An initial solution presented is based on data aggregation techniques that associate plaintext metadata to sets of encrypted data. However, plaintext metadata may leak sensitive information and data aggregation introduces unnecessary network overheads.

The use of fully hemimorphic encryption would guarantee the execution of any operation over encrypted data, but existing implementations are affected by huge computational costs to the extent that the execution of SQL operations over a cloud database would become impractical. Other encryption algorithms characterized by acceptable computational complexity support a subset of SQL operators.

LITERATURE SURVEY

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 related review part of this section describes the related adaptive encryption schemes, the related works on encrypted metadata stored in the cloud database, and the main related operations for the management of the encrypted cloud database

Rajkumar Buyya, et al stated the significant advances in Information and Communications Technology (ICT) over the last half century, there is an increasingly perceived vision that computing will one day be the 5th utility (after water, electricity, gas, and telephony). This computing utility, like all other four existing utilities, will provide the basic level of computing service that is considered essential to meet the everyday needs of the general community. To deliver this vision, a number of computing paradigms have been proposed, of which the latest one is known as Cloud computing.

Hence, in this paper, to define Cloud computing and provide the architecture for creating Clouds with market-oriented resource allocation by leveraging technologies such as Virtual Machines (VMs). They also provide insights on market-based resource management strategies that encompass both customer-driven service management and computational risk management to sustain Service Level Agreement (SLA)-oriented resource allocation. In addition, they reveal their early thoughts on interconnecting Clouds for dynamically creating global Cloud exchanges and markets.

Chee Shin Yeo et al were stated that the grid computing enable the virtualization and dynamic delivery of computing services on demand to realize utility

With this new outsourcing service model, users are able to define their service needs through Service Level Agreements (SLAs) and only have to pay when they use the services. They do not have to invest on or maintain computing infrastructures themselves and are not constrained to specific computing service providers. Thus, a commercial computing service will face two new challenges: (i) what are the objectives or goals it needs to achieve in order to support the utility computing model, and (ii) how to evaluate whether these objectives are achieved or not. To address these two new challenges, this paper first identifies four essential objectives that are required to support the utility computing model: (i) manage wait time for SLA acceptance, (ii) meet SLA requests, (iii) ensure reliability of accepted SLA, and (iv) attain profitability.

Y. Fu, J. Chase et al were stated that the presents Sharp, a framework for secure distributed resource management in an Internet-scale computing infrastructure. The cornerstone of Sharp is a construct to represent cryptographically protected resource claims-promises or rights to control resources for designated time intervals-together with secure mechanisms to subdivide and delegate claims across a network of resource managers.

The present experimental results from a Sharp prototype for PlanetLab, and illustrate its use with a decentralized barter economy for global PlanetLab resources. The results demonstrate the power and practicality of the architecture, and the effectiveness of oversubscription for protecting resource availability in the presence of failures.

These systems need effective resource management for fair sharing of community resources, performance isolation and predictability, and adaptivity to changing conditions. Come to one motivating example, shows a classic "tragedy of the commons" for PlanetLab during a period of high demand. Here, a growing number of PlanetLab users simultaneously request "slices" of resources from arbitrarily selected nodes to host distributed systems experiments. In this system, the PlanetLab nodes schedule their requests locally, with no mechanism to discover or reserve resources, coordinate resource usage across the system, or control resource usage by users or groups. Users have little basis to predict the resources available to them at each site, creating an incentive to request more resources than needed.

Users who obtain poor results due to overloading at one or more sites either retry their experiments-consuming even more resources-or give up. The scenario is similar to congestion collapse in the Internet.

K. Lai, L. Rasmusson, E. Adar, L. Zhang and B. A. Huberman. stated that the Distributed clusters like the Grid and PlanetLab enable the same statistical multiplexing efficiency gains for computing as the Internet provides for networking. One major challenge is allocating resources in an economically efficient and low-latency way. A common solution is proportional share, where users each get resources in proportion to their pre-defined weight.

Due to the way resources very simple distributed in PlanetLab, the rise in contention decreases the portion of resources received by any individual user, thereby reducing the amount of useful work that can be completed in the system. They argue that given the appropriate mechanisms, end-users can cooperate to arrive at an optimal resource allocation in spite of excess demand.

David Irwin, Jeffrey Chase et al proved that the This paper presents the design and implementation of Shirako, a system for on-demand leasing of shared networked resources. Shirako is a prototype of a serviceoriented architecture for resource providers and consumers to negotiate access to resources over time, arbitrated by brokers. It is based on a general lease abstraction: a lease represents a contract for some quantity of a typed resource over an interval of time. Resource types have attributes that define their performance behavior and degree of isolation.

III. PROBLEM FORMULATION

Initially design 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. Then, they evaluate the performance of encrypted database services by assuming the standard TPC-C benchmark as the workload and by considering different network latencies. The testbed show that most performance overheads of adaptively encrypted cloud databases are masked by net-work latencies that are typical of a geographically distributed cloud scenario.

This thesis has been focus on database services and takes an opposite direction by evaluating the cloud service costs from a tenant's point of view. This approach is quite original because related thesis evaluate the pros and cons of porting scientific applications to a cloud platform, as focusing on specific astronomy software and a specific cloud provider, and presenting a compos able cost estimation model for some classes of scientific applications. Besides the focus on a different context (scientific versus database applications), the proposed model can be applied to any cloud database service provider, and it takes into account that over a medium-term period the database workload and the cloud prices may vary.

IV. EXISTING METHODOLOGY

In the existing system, all data and metadata stored in the cloud database are encrypted and application running on a legitimate client can transparently issue SQL operations SELECT, INSERT, UPDATE and DELETE command to the encrypted cloud database through the encrypted database interface. Data transferred between the user application and the encryption engine is 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 that retrieves the encrypted metadata and decrypts them with the master key.

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.

Like existing system, proposed system also manages the data using both cloud server side and client side. In addition, user group is maintained so that a single key is distributed to multiple users in the same group to reduce the key preparation overhead for each user. This makes less computation overhead in both client and server side. Also, based on the security level, different data is encrypted with different encryption mechanism and allowed to secure the data in inexpensive manner.

Due to this redundancy the data can be easily modified by unauthorized users which can be stored in the database. This leads to loss of data privacy and security to database. The proposed scheme ensures that cyclic redundancy check and time-tested practices and technologies for managing trust relationships in traditional enterprise environments can be extended to work effectively in both private and public clouds. Those practices

include data encryption, strong authentication and fraud detection, etc.

The organization has to buy a public cloud with a resources, this is a direct cost. There are many direct benefits of covering the manual system to computerized system. The user can be given responses on asking questions, justification of any capital outlay is that it will reduce expenditure or improve the quality of service or goods, which in turn may be expected to provide the increased profits.

The cost and benefit analysis may be concluded that computerized system is favorable in today's fast moving world. The assessment of technical feasibility must be based on an outline design of the system requirements in terms of input, output, files, programs and procedure. The project aims to presents an innovative and effective pattern discovery technique which includes the processes of pattern deploying and pattern evolving, to improve the effectiveness of using and updating discovered patterns for finding relevant and interesting information. The current system aims to overcome the problems of the existing system. The current system is to reduce the technical skill requirements so that more number of users can access the application

The Proposed system accessing process to solves problems what occurred in existing system. The current day-to-day operations of the organization can be fit into this system. Mainly operational feasibility should include on analysis of how the proposed system will affects the organizational structures and procedures.

V. RESEARCH METHODOLOGY ARCHITECTURE DESIGN

The proposed system 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 server seen the clients and the cloud database, the proposed solution guarantees the same level of scalability and availability of the cloud service. Fig. 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:

- Plain data represent the tenant information
- Encrypted data are the encrypted version of the plain data, and are stored in the cloud database
- Plain metadata represent the additional information that is necessary to execute SQL operations on encrypted data;
- Encrypted metadata are the encrypted version of the plain metadata, and are stored in the cloud database
- Master key is the encryption key of the encrypted metadata, and is known by legitimate clients.

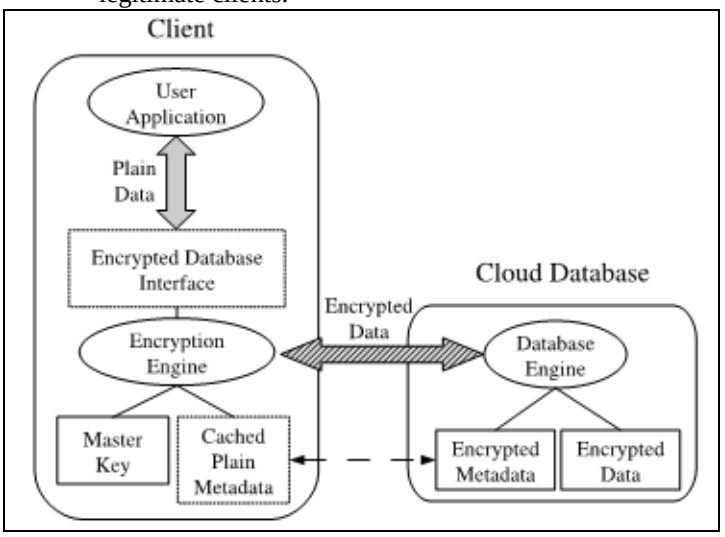

Fig. 4.1Encrypted Cloud Database Architecture

Data transferred between the user application and the encryption engine is 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 that retrieves the encrypted metadata and decrypts them with the master key. 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 state- ments to the cloud database, and then to decrypt the results. The results are returned to the user application through the encrypted database interface.

B. ADAPTIVE ENCRYPTION SCHEMES

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, 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 sup- ports 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 (i.e., ;<; ;>;).
- 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 sup- port 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. As an example, if we consider a database

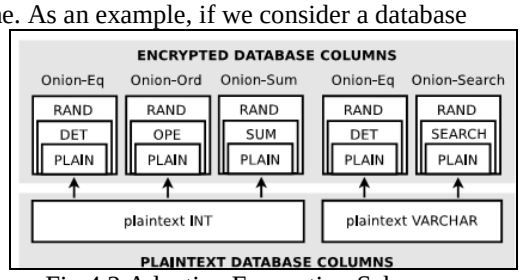

Fig 4.2 Adaptive Encryption Schemes

C. METADATA STRUCTURE

Metadata include all information that allows a legitimate client knowing the master key to execute SQL operations over an encrypted database. They are organized and stored at table-level granularity to reduce communication overhead for retrieval, and to improve management of concurrent SQL operations. Define all metadata information associated with a table as table metadata. Let us describe the structure of a table metadata by referring

Table metadata include the correspondence between the plain table name and the encrypted table name because each encrypted table name is randomly generated. Moreover, for each column of the original plain table they also include a set of column metadata containing the name and the data type of the corresponding plain column (e.g., integer, varchar, datetime). Each set of column metadata is associated with as many sets of onion metadata as the number of onions associated with the column. Onion metadata describe all the encryption information about an onion and its layers, hence they are organized in a data structure that contains the following attributes:

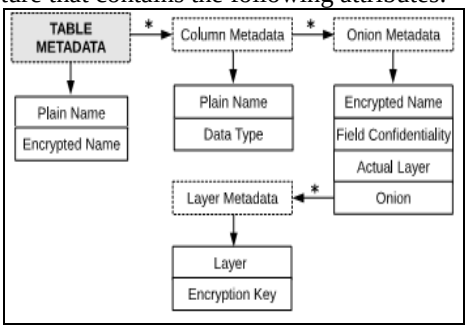

Fig 4.3 Metadata Structure

- Encrypted name is the name of the encrypted column (i.e., the onion) in the encrypted cloud database
- Actual encryption layer is the name of the most external layer of the encrypted data (e.g., Rand) stored in the column;
- Field confidentiality denotes which set of keys must be used to encrypt a column data, because only columns that share the same encryption keys can be joined and identify three types of field confidentiality parameters: self denotes a private set of keys for the column, multi column identifies the sharing of the same set of keys among two columns, database imposes the same set of keys on all columns of the same data type.
- Onion parameter identifies the type of onion that is used to encrypt data.

Each set of onion metadata is associated with as many sets of layer metadata as the number of layers required by the onion type. Each set of layer metadata includes an encryption key and a label identifying the corresponding encryption algorithm. The set of encryption keys for each onion is generated according to the field confidentiality parameter imposed on each encrypted column.

D. EXECUTION OF SQL OPERATIONS

When a user/application wants to execute an operation on the cloud database, the client encryption engine analyzes the SQL command structure and identifies which tables, columns and SQL operators are involved. The client issues a request for the table metadata for each involved table, and decrypts the metadata with the master key. Then, the client determines whether the SQL operators are supported by the actual layers of the onions associated with the involved columns.

If required, the client issues a request for layer removal in order to support the SQL operators at runtime. By using the information stored in the table metadata, the client is able to encrypt the parameters of the SQL operations: tables and columns names, and constant values. The client issues this new statement called encrypted SQL operation to the cloud database which transparently executes it over encrypted data. The encrypted results are decrypted using information contained in the metadata.

The adaptive layer removal is the process that dynamically removes the external layer of an onion in order to adaptively support SQL operations issued by legitimate clients. The cloud database can execute the adaptive layer removal if and only if a legitimate client invokes the stored procedure and gives to it the decryption key of the most external encryption layer. As each layer has a different encryption key, the data remain encrypted and the cloud provider cannot access plaintext data. For security reasons they also assume that the adaptive layer removal mechanism does never expose the Plain layer of an onion.

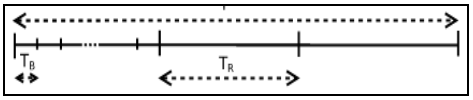

Fig 4.4 Execution of SQL Operations

CONCLUSION

There are two main tenant concerns that may prevent the adoption of the cloud as the fifth utility: data confidentiality and costs. This project addresses both issues in the case of cloud database services. These applications have not yet received adequate attention by the academic literature, but they are of utmost importance if it is considered that almost all important services are based on one or multiple databases.

It addresses the data confidentiality concerns by proposing a novel cloud database architecture that uses adaptive encryption techniques with no intermediate servers. This scheme provides tenants with the best level of confidentiality for any database workload that is likely to change in a medium-term period. The results demonstrate that the network latencies that are typical of cloud database environments hide most overheads related to static and adaptive encryption.

At present, the project demonstrates the feasibility and performance of the proposed solution through a software prototype. In future research, the project could evaluate the proposed or alternative architectures for multi-user key distribution schemes and under different threat model hypotheses.

REFERENCES

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

[2] C. S. Yeo and R. Buyya. Integrated Risk Analysis for a Commercial Computing Service. In Proceedings of the 21st IEEE International Parallel and Distributed Processing Symposium (IPDPS 2007), Long Beach, USA, Mar. 2007.

[3] Baker, M. and R. Buy ya: 1999, 'Cl uster Computi ng: The Comm o di ty Sup ercomputer'. Software: Practice and Experience 29(6), 551–576.

[4] Foster, I. and C. Kesselman (eds.): 2003, The Grid 2: Blueprint for a New Computing Infrastructure. San Francisco, CA: Morgan Kaufmann.

[5] Yeo, C. S., R. Buyya, M. D. de Assuncao, J. Yu, A. Sulistio, S. Venugopal, and M. Placek: 2007, 'Utility Computing on Global Grids'. In: H. Bidgoli (ed.): The Handbook of Computer Networks. Wiley, Chapt. 143.

[6] Yeo, C. S. and R. Buyya: 2006a, 'A Taxonomy of Market-based Resource Management Systems for Utility-driven Cluster Computing'. Software: Practice and Experience 36(13), 1381–1419.