Performance and cost evaluation of public cloud databases through adaptive encryption techniques

MadiReddy Vijay Reddy [1], Dr. Aasim Zafar [2]
[1]Assistant professor, Dept. of CSE, RamanandaTirtha Engineering College, Nalgonda,Andhra Pradesh, India
[2] Professor, Dept. of CSE , Aligarh Muslim University, Aligarh, India

Abstract: 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. We propose a novel architecture for adaptive encryption of public cloud databases that offers an interesting alternative to the trade-off between the required data confidentiality level and the flexibility of the cloud database structures at design time. We propose a novel architecture for adaptive encryption of public cloud database. Adaptive encryption allow any sql operation over encrypted data.The novel cloud database architecture that uses adaptive encryption technique with no intermediate servers.This scheme provides cloud provider with the best level of confidentiality for any database workload.we can determine the encryption and adaptive encryption cost of data confidentiality from the research point of view.

Keywords- SecureDBaaS, cloud, security, DBaaS, Cloud database, confidentiality, encryption, adaptivity, cost model.

I. INTRODUCTION

Cloud computing has been increased for providing services over the internet. These services must be potential to fulfill all the requirements of the customers. An organization acquires these cloud services for the model of computing, storage and model for communication over the internet. These services must provide scalability, availability and elasticity properties of databases. Cloud computing growing rapidly due to interest in recent years for handling large amount of data with its elasticity, flexibility properties. Whereas privacy and security policies monitors the organizations information system and standards, procedures and controls guidelines for preserving confidentiality ,integrity and availability of cloud database system. Organizations must need a particular securitymanagement services and control management over the cloud computing. For ensuring privacy and security requirements for cloud computing data should be in encrypted format. There are too many security issues are available to protect clouds from outside threats are similar to those who already facing big data centers. In the cloud, however, this responsibility is divided among potentially many parties, including the cloud user, the cloud vendor, and any third-party vendors that users rely on for security-sensitive software or configurations.

A cloud database is a one that typically executes on a cloud computing platform. For the storage and management of structured data database as a service of cloud based approach is used. On the other hand in cloud based approach DBaaS is an oriented toward self-service and easy management provides a flexible, scalable and on demand platform. Users may purchase access to a database service maintained by a cloud database provider. In a DBaaS application owners do not need maintain and install the database instead the service provider takes care of the responsibility for maintaining and installing database and they pay according to their use All the data needs to be providing data confidentiality because database containing highly valuable information. Users wishing to protect their sensitive information among untrusted proxies or third parties; so plaintext data should be visible only to trusted parties excluding cloud provider and internet. Whereas in untrusted context data must be in encrypted format for preventing unauthorized access to sensitive information. Here proposed an architecture named as a SecureDBaaS for allowing multiple, independent and distributed users to perform simultaneous operations on encrypted data. Reason behind of proposing SecureDBaaS is to allow concurrent and

independent execution of operation on encrypted data through SQL statements with database qualities such as elasticity, scalability and ease of availability. The main focus is using SQL statements to modify the database structure. In earlier because of intermediate proxies, sometimes failure occurs that results in bottleneck that limits the qualities of database service. So intermediate proxies are eliminated to increase the data confidentiality level of database system. This architecture is solution for geographically distributed users who want to access its database concurrently and independently. SecureDBaaS uses a various isolation techniques, cryptographic techniques for managing encrypted metadata on the untrusted cloud databases. To use SecureDBaaS architecture here it should achieve also the reliability and availability and elasticity properties of cloud DBaaS. The best quality of the SecureDBaaS is that it is immediately applicable to any DBMS because it does not require changes to the cloud database services. Here SecureDBaaS is a new

architecture is designed where concurrent and independent n distributed clients can access the information through cloud database by sql statements. In a new approach in this work two types of encryptions are defined, one is DES and another is AES. by comparison determined result is AES is better than DES in all way like security, availability and scalability. The overall conclusion of paper is very crucial because first time it demonstrates the availability and applicability of encryption cloud DBaaS with respect to performance and overhead authorities.

II. RELATED WORKS

Unlimited numbers of computing resources are available on demand, quickly for the cloud computing users. One main aspect in the cloud service is the able to pay for use of computing resources on a short-time basis as needed and release them as it is no longer needed. An internet service provider provides the cloud services where quality of service is dependent on the cost, so a customer doesn't jump on lowest cost service. Security is the most important objections to cloud computing. Many of the security issues are evolved during protecting clouds from cloud users, vendors and thirdparty vendors. most of the security concerns is to protect cloud from cloud provider. Various techniques like the standard defense and user level encryption are effective in the cloud [1].

Continuous checking of secured information requires maintaining awareness of threats ,security controls and vulnerabilities to support risk management. For fulfilling the operation of monitoring the organization is dependent on cloud provider. An analysis of system security controls and security features are used for vulnerabilities identification to protect cloud environment. Cloud computing is emerging technology as technology advances there need to improve performance and other quality services from public cloud and including privacy and security [2].DBaaS provides various features related to security and database services. Working in this field varies with time as well as new techniques to improve the performance of remote database services. Data confidentiality is the important aspect of cloud database services and concurrent/simultaneous execution of operations with distributed manner. Also new techniques arrived which allows distributed access to database with platform independent properly. SecureDBaaS provides data confidentiality by executing sql operations on encrypted data which allows concurrent read, write and modification to the database structure. It maintains databases properly elasticity, scalability and availability of cloud database because it does not need any intermediate server. It always removed a trusted proxy because tenant and metadata are always in encrypted format.

Its use for relational database which are very applicable to different database management system implementation.[3].A number of group of trusted clients outsources an arbitrary computational service to a remote provider, which they do not fully trust and that may be cause to attacks. Here presented a novel protocol that guarantees atomic operations to all clients when the provider is correct and fork-linearizable semantics when it is faulty; this means that all clients which observe each other's operations are consistent, in the sense that their own operations, plus those operations whose effects they see, have occurred atomically in same sequence.

This protocol generalized previous approaches that provided guarantees for outsourced storage services [4].This article defines the design, implementation, and evaluation of Depot, a cloud storage system that minimizes assumptions of trust. Depot tolerates malicious behavior by number of clients or servers, yet it provides safety of guarantees to correct clients. It provides the guarantees by using two-layer architecture. Second, Depot implements some protocols that uses this consistent of updates to provide consistency, staleness, durability, and recovery properties. Here our evaluation suggests that the costs of these guarantees are modest and that Depot may tolerate faults and maintain good availability, latency, overhead, and staleness even when significant faults occur [5].

In this cloud environment an inserting secure important information in untrusted third parties, which causes risks of the confidentiality of data. Where it takes care of guarantees confidentiality of the Database as a service (DBaaS) which remains a problem. Therefore to resolve the Confidential and Concurrent to SecureDBaaS is determined because there is initial resolution to produce availability, accessibility, reliability and security which not exposing unencrypted information to the cloud provider. It additionally permits multiple, freelance and regionally distributed clients to execute synchronic operations on encrypted and preserve information confidentiality at the consumer and cloud level. It removes intermediate server between the cloud consumer and the cloud provider. To realize that Confidential concurrent to Secure DBaaS integrates existing cryptographic schemes, isolation mechanisms and management of encrypted information on the untrusted cloud information.[6].

Here proposed a fully homomorphism encryption scheme - i.e., a scheme that allows one to evaluate circuits over encrypted data without being able to decrypt. So solution comes in three steps. First, construct an encryption scheme which permits evaluation of arbitrary circuits , that suffices to construct an encryption scheme that may evaluate its own decryption circuit ;in this work called a scheme that can evaluate its (augmented) decryption circuit boots trappable. Next, here given description of a public key encryption scheme using ideal lattices that is almost boots trappable. Lattice-based cryptosystems specifically have decryption algorithms with low complexity, often dominated by an inner product computation. Also, ideal lattices provide both additive and multiplicative homeomorphisms, as needed to evaluate general circuits. There by obtain a boots trappable encryption scheme, without reducing the depth that the scheme can evaluate. Here accomplished this by enabling the encrypted to start the decryption process, leaving less work for the decrypted, much like the server leaves less work for the decrypted in a cryptosystem [7]. The technological aspects of developing database as a service lead to new research challenges. The service provider always would need to provide sufficient security measures to protect the privacy of data. Here proposed data encryption as the solution to this problem. Second key challenge is that of performance. Since the interaction between the database service provider and users takes place in a different medium, the network, than it does in traditional databases, there are potential overheads introduced by this architecture. Data privacy can be achieved by using a suitable encryption algorithm. here proposed, implemented, and evaluated different encryption schemes [6].

III. PROPOSED SYSTEM

We describe the architecture we propose to guarantee data confidentiality through adaptive encryption methods in cloud database environments.

A. Architecture model

We refer to the distributed architecture represented in Fig.1, where we assume that independent and distributed clients (Client 1 to N) access a public cloud database service [7]. All information (i.e., data and metadata) is stored encrypted in the cloud database. The proposed architecture manages five types of information.

• **Plain data:** the informative content provided by the client users.
• **Encrypted data**: the encrypted data those are stored in the cloud database.
• **Plain metadata:** all the information required by the clients to manage encrypted data on the cloud database.

• **Encrypted metadata:** the encrypted metadata that is stored in the cloud database.
• **Master key:** the encryption key of the encrypted metadata. We assume that it is distributed to all legitimate clients.

A legitimate client can issue SQL operations (SELECT, INSERT, UPDATE, DELETE) to the encrypted cloud database by executing the following steps. It retrieves encrypted metadata, and obtains plain metadata by decrypting them through the master key. The metadata are cached locally in a volatile representation that is used for improving performance. Then, the client can issue SQL operations over the encrypted data (i.e., the real informative content), because it is able to encrypt the queries, their parameters, and decrypt their results by using the local plain metadata.

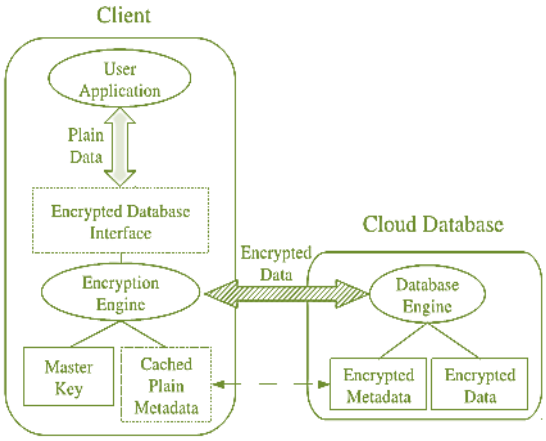

Fig. 1: Encrypted cloud database architecture

This architecture guarantees confidentiality of data in a security model in which the WAN network is untrusted (malicious), while client users are trusted, that is, they do not reveal any information about plain data, plain metadata, and the master key. The cloud provider administrator is semihonest [8] (also called honest-but-curious), because he could try accessing information stored in the database, but he does not modify internal data and SQL operations results.

B. Adaptive encryption techniques

We consider SQL-aware encryption algorithms that guarantee data confidentiality and allow the cloud database server to carry out a large set of SQL operations over encrypted data. Each algorithm supports a specific subset of SQL operators. This paper refers to the following encryption schemes.

Deterministic (Det): it deterministically encrypts data, so that the encryption of an input value always guarantees the same output value. It supports the equality operator.
Order Preserving Encryption (OPE) [4]: this encryption scheme preserves in the encrypted values the numerical order of the original unencrypted data. It supports the following SQL
operators: equal, unequal, less, less or equal, greater, greater or equal.

Sum: this encryption algorithm is homomorphic with respect to the sum operation: summing unencrypted data is equivalent to multiplying the correspondent encrypted values.
It supports the sum operator between integer values.

Search: it supports equality check on full strings (i.e., the LIKE operator) that do not include fragments of words.

Random (Rand): It is a semantic secure encryption (INDCPA) that does not reveal any information of the original plain value. It does not support any SQL operator.
Plain: a special kind of "encryption" that leaves values unencrypted. It supports all SQL operators, and it is included to store publicly available data, or some anonymous values that do not require any data confidentiality.
In order to have architecture to support at runtime the SQL operations issued by the clients, while preserving a high level of confidentiality on the columns that are not involved in any operation. Therefore we organize the encryption schemes into structures called Onions. Each Onion is composed by different encryption algorithms, called (Encryption) Layers, one above the other. Outer Layers guarantee higher data confidentiality and lower number of allowed operations, and each Onion supports a specific set of operators.

In this paper, we consider and design the following Onions, which are also represented in Fig. 2.

Onion-Eq: it manages the equality operator.
Onion-Ord: it manages the following operators: less, less or equal, greater, greater or equal, equal, unequal.
Onion-Sum: it manages the sum operator.
Onion-Search: it manages the string equality operator.
Onion-Single-Layer: a special type of Onion that supports only a single Encryption Layer. It is recommended for columns in which operations to be supported are known at design time.

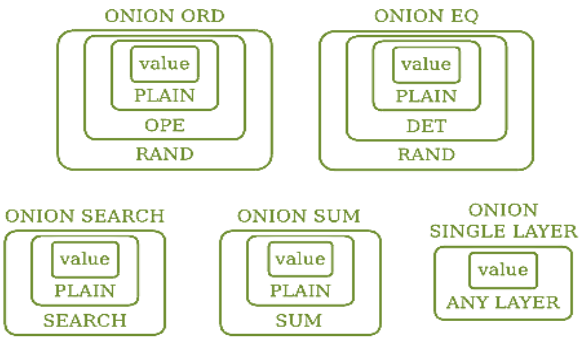

Figure 2. Onions and layers structure

Fig. 3 represents an example of the Onions in the structure of the encrypted database table. Each column's Onion corresponds to a different encrypted column. We have two columns: an integer column id, with Onion-Eq and Onion-Ord, and a string column name, with Onion-Search. We observe that the representation of the encrypted table in this figure is just for clarity, because in the real implementation the table and the columns names should be encrypted too.

Table 1 (Plain)

id	name
1	Mark
3	Luke
4	John

Table 1 (Encrypted)

id-OnionEq	id-OnionOrd	name-OnionSearch
x4cx52	x2bx3xc2	xz53j2hzfap3hx
xd2vsd	xbcv3b3f	x34k2x3243mgj3
x34ds2	x4nj3h3x	x45h23x3cxfhx2

Figure 3. Onions in the encrypted database

The main benefit of the Onions is to allow our architecture to adapt the level of data confidentiality to the current SQL workload by decrypting an encrypted column's outer Layer(s). In such a way, it supports at runtime any SQL operation issued by a user.

The two main phases involved in the column re-encryption operation are re-encryption invocation on the client side and the re-encryption execution on the cloud database side. We describe these two phases with reference to Fig. 4.

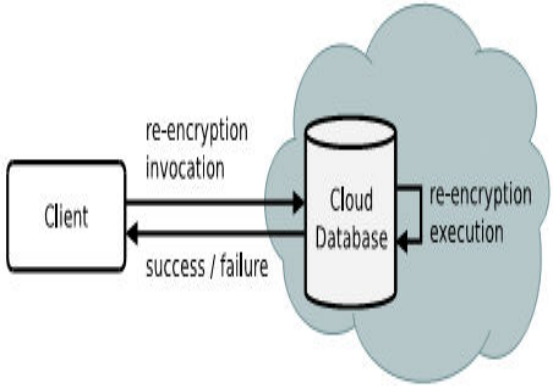

Figure 4. Automatic column re-encryption

In the re-encryption invocation phase, the client examines the plaintext query issued by the user (which can also be an external application) and evaluates whether the involved SQL operators (e.g., equality checks and order comparisons) are supported with respect to the Actual Layers of the Onions available on the involved columns. In the re-encryption execution phase, the cloud database engine executes a properly defined stored procedure that diminishes the Actual Layer of an Onion by decrypting its row values one by one. After the stored procedure execution, the cloud database sends the information about its outcome (success or failure) to the client that issued the request for reencryption.

IV. CONCLUSION

We proposed an architecture that supports adaptive data confidentiality in cloud database environments without requiring any intermediate trusted proxy. Adaptive encryption mechanisms have two main benefits: they guarantee at runtime the maximum level of data confidentiality for any SQL workload, and they simplify database configuration at design

time. Though, they are affected by high computational costs with respect to non adaptive encryption schemes. This paper demonstrated that applying adaptive encryption methods to cloud

database services is a suitable solution, because network latency masks the overhead caused by adaptive encryption for most SQL operations.

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. Mazie`res, and D. Shasha, "Secure Untrusted Data Repository (SUNDR)," Proc. Sixth USENIX Conf. Opearting 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] H. Hacig ¨um ¨us¸, B. Iyer, C. Li, and S. Mehrotra, "Executing sql over encrypted data in the database-service-provider model," in Proc. ACM SIGMOD Int'l Conf. Management of data, June 2002.

[9] L. Ferretti, M. Colajanni, and M. Marchetti, "Distributed, concurrent, and independent access to encrypted cloud databases," IEEE Trans. Parallel and Distributed Systems, vol. 25, no. 2, Feb. 2014.

[10] 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.

[11] C. Gentry, "Fully homomorphic encryption using ideal lattices," in Proc. 41st ACM Symp. Theory of computing, May 2009.

[12] A. Boldyreva, N. Chenette, and A. O'Neill, "Order-preserving encryption revisited: Improved security analysis and alternative solutions," in Proc. Advances in Cryptology – CRYPTO 2011.