The Implementation of "Evaluation of an Adaptive Encryption Architecture for Cloud Databases: Performance and Cost Perspective

MS Sarika Malani[1], Prof Ganesh Regulwar[2]

[1] Ms. Sarika Malani, ME second Year CSE, BNCOE Pusad, MH
[2] Prof. Ganesh B.Regulwar ,Professor Dept. of CSE, BNCOE Pusad, MH

---***---

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 tradeoff between the required data confidentiality level and the flexibility of the cloud database structures at design time. We demonstrate the feasibility and performance of the proposed solution through a software prototype. Moreover, we propose an original cost model that is oriented to the evaluation of cloud database services in plain and encrypted instances and that takes into account the variability of cloud prices and tenant workloads during a medium-term period.*

Key Words: adaptively, cloud database, cost model, confidentiality, encryption

1.INTRODUCTION:

Although this adaptive encryption architecture is attractive because it does not require defining at design time which database operations are allowed on each column, it poses novel issues in terms of feasibility in a cloud context, and storage and network costs estimation. In this system, we investigate each of these issues and we reach original conclusions in terms of prototype implementation, performance evaluation, and cost evaluation. We implement the first proxy-free architecture for adaptive encryption of cloud databases. It does not limit the availability, elasticity and scalability of a plain cloud database, because concurrent clients can issue parallel operations without passing through some centralized component as in alternative architectures. We evaluate the performance through this prototype implementation by considering the clever cloud platform for the workload and different network latencies. we show that most performance overheads of adaptively encrypted cloud databases are masked by network latency values that are quite typical of a cloud scenario. Other performance evaluations carried out in assumed a LAN scenario and no network latency. Moreover, we propose the first analytical cost estimation model for evaluating cloud database costs in plain and encrypted instances from a tenant`s point of view in a medium-term period. It takes also into account the variability of cloud prices and the possibility that the database workload may change during the evaluation period. This model is instanced with respect to several cloud provider offers and related real prices. As expected, adaptive encryption influences the costs related to storage size and network usage of a database service. However, it is important that a tenant can anticipate the final costs in its period of interest, and can choose the best compromise between data confidentiality and expenses

2. REQUIREMENTS ELICITATION:

2.1 Existing System :

Most results concerning encryption for cloud-based 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 latter 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.

2.2 Proposed System:

The 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 the 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. Although this adaptive encryption architecture is attractive because it does not require to define at design time which database operations are allowed on each column, it poses novel issues in terms of applicability to a cloud context, and doubts about storage and network costs. We investigate each of these issues and we reach three original conclusions in terms of prototype implementation, performance evaluation, and cost evaluation. We 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. we show that most performance overheads of adaptively encrypted cloud databases are masked by network latencies that are typical of a geographically distributed cloud scenario. Finally, we propose the first analytical cost estimation model for evaluating cloud database costs in plaintext and encrypted configurations from a tenant's point of view over a medium-term period. This model also considers the variability of cloud prices and of the database workload during the evaluation period, and allows a tenant to observe how adaptive encryption influences the costs related to storage and network usage of a database service. By applying the model to several cloud provider offers and related prices, the tenant can choose the best compromise between the data confidentiality level and consequent costs in his period of interest.

3. Design Modules:

3.1 Architecture design:

The proposed system supports adaptive encryption methods for public cloud database service, where distributed and concurrent clients can issue direct SQL operations. By avoiding an architecture based on one or multiple intermediate servers between the clients and the cloud database, the proposed solution guarantees the same level of scalability and availability of the cloud service. Figure 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 is the tenant information;
- encrypted data is stored in the cloud database;
- plain metadata represent the additional information that is necessary to execute SQL operations on encrypted data;
- encrypted metadata is the encrypted version of the metadata that are stored in the cloud database;
- master key is the encryption key of the encrypted metadata that is distributed to legitimate clients.

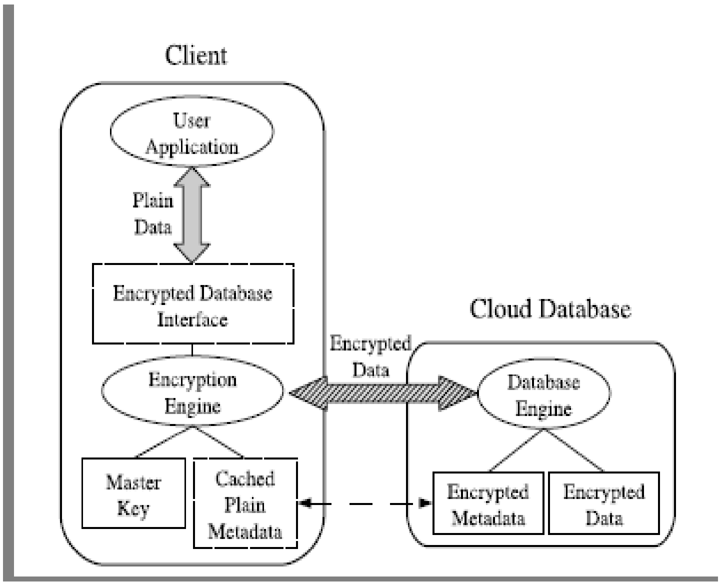

Fig. 1: Encrypted cloud database architecture

All data and metadata stored in the cloud database are encrypted. Any application running on a legitimate client can transparently issue SQL operations (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 in plain format, 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 it through the master key. In order to improve performance, the plain metadata are cached locally by the client as volatile information. After obtaining the metadata, the encryption engine is able to execute the SQL operation on encrypted data, and then to decrypt the results. The results are returned to the user application through the encrypted database interface.

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 could be interested in accessing 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.

3.2 System Design:

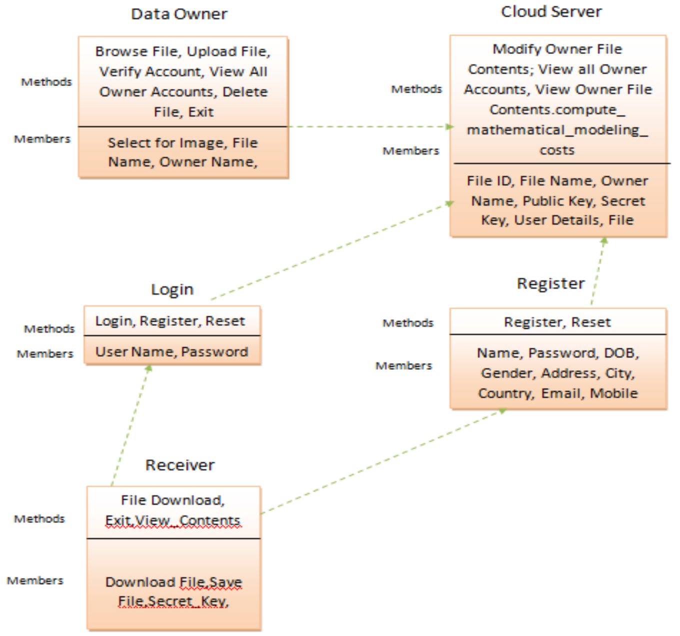
Data Owner
Methods
Browse File, Upload File, Verify Account, View All Owner Accounts, Delete File, Exit
Members
Select for Image, File Name, Owner Name,
Cloud Server
Methods
Modify Owner File Contents; View all Owner Accounts, View Owner File Contents.compute_ mathematical_modeling_ costs
Members
File ID, File Name, Owner Name, Public Key, Secret Key, User Details, File
Login
Methods
Login, Register, Reset
Members
User Name, Password
Register
Methods
Register, Reset
Members
Name, Password, DOB, Gender, Address, City, Country, Email, Mobile
Receiver
Methods
File Download, Exit,View, Contents
Members
Download File,Save File,Secret_Key,

Fig. 2: Class Diagram

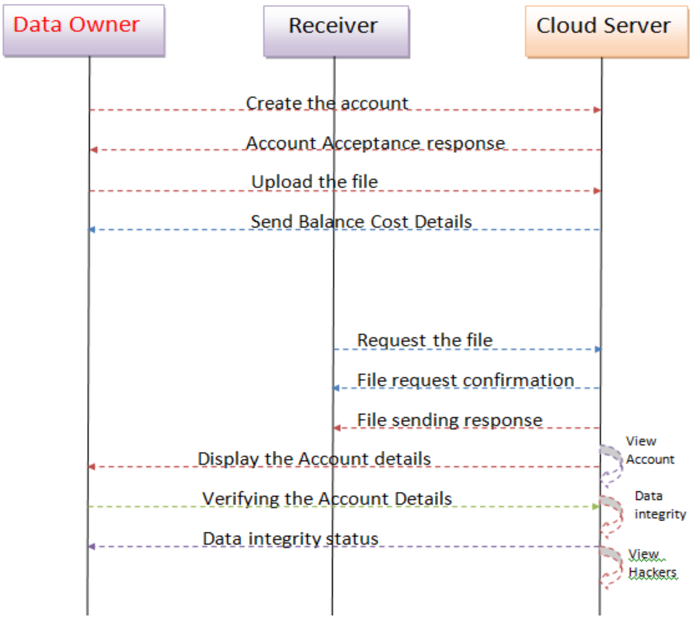

Fig. 3: Sequence Diagram

4. IMPLEMENTATION MODULES:

1. Adaptive encryption
2. Metadata structure
3. Encrypted database management
4. Cost Estimation of cloud database services
5. Cost model
6. Cloud pricing models
7. Usage Estimation

4.1 Adaptive encryption:

We 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.

1. Random (Rand): it is the most secure encryption because it does not reveal any information about the original plain value (IND-CPA) . It does not support any SQL operator, and it is used only for data retrieval.
2. Deterministic (Det): it deterministically encrypts data, so that equality of plaintext
data is preserved. It supports the equality operator.
3. 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., =; <;<_; >;>_).
4. 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.
5. Search: it supports equality check on full strings (i.e., the LIKE operator).
6. Plain: it does not encrypt data, but it is useful to support all SQL operators on non confidential data.

4.2 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 a table-level granularity to reduce communication overhead for retrieval, and to improve management of concurrent SQL operations. We define all metadata information associated to a table as table metadata. Let us describe the structure of a table metadata .Table metadata includes 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 it also includes a column metadata parameter containing the name and the data type of the corresponding plain column (e.g., integer, string, timestamp). Each column metadata is associated to one or more onion metadata, as many as the number of onions related to the column.

4.3 Encrypted database management:

The database administrator generates a master key, and uses it to initialize the architecture metadata. The master key is then distributed to legitimate clients. Each table creation requires the insertion of a new row in the metadata table. For each table creation, the administrator adds a column by specifying the column name, data type and confidentiality parameters. These last are the most important for this paper because they include the set of onions to be associated with the column, the starting layer (denoting the actual layer at creation time) and the field confidentiality of each onion. If the administrator does not specify the confidentiality parameters of a column, then they are automatically chosen by the client with respect to a tenant`s policy. Typically, the default policy assumes that the starting layer of each onion is set to its strongest encryption algorithm.

4.4 Cost Estimation of cloud database services:

A tenant that is interested in estimating the cost of porting its database to a cloud platform. This porting is a strategic decision that must evaluate confidentiality issues and the related costs over a medium-long term. For these reasons, we propose a model that includes the overhead of encryption schemes and variability of database workload and cloud prices.

4.5 Cost model:

The cost of a cloud database service can be estimated as a function of three main parameters:

Cost = f(Time, Pricing,Usage)
where:

- Time: identifies the time interval T for which the tenant requires the service.
- Pricing: refers to the prices of the cloud provider for subscription and resource usage; they typically tend to diminish during T .
- Usage: denotes the total amount of resources used by the tenant; it typically increases during T .In order to detail the pricing attribute, it is important to specify that cloud providers adopt two subscription policies: the on-demand policy allows a tenant to pay per-use and to withdraw its subscription anytime; the reservation policy requires the tenant to commit in advance for a reservation period. Hence, we distinguish between billing costs depending on resource usage and reservation costs denoting additional fees for commitment in exchange for lower pay-per-use prices. Billing costs are billed periodically to the tenant every billing period.

4.6 Cloud pricing models:

Popular cloud database providers adopt two different billing functions, that we call linear L and tiered T . Let us consider a generic resource x, we define as xb its usage at the b-th billing period and px b its price. If the billing function is tiered, the cloud provider uses different prices for different ranges of resource usage. Let us define Z as the number of tiers, and [ˆx1, . . . , ˆxZ−1] as the set of thresholds that define all the tiers. The uptime and the storage billing functions of Amazon RDS are linear, while the network usage is a tiered billing function. On the other hand, the uptime billing functions of Azure SQL is linear, while the storage and network billing functions are tiered.

4.7 Usage Estimation:

The uptime is easily measurable, it is more difficult to estimate accurately the usage of storage and network , since they depend on the database structure, the workload and the use of encryption. We now propose a methodology for the estimation of storage and network usage due to encryption. For clarity, we define sp, se, sa as the storage usage in the plaintext, encrypted, and adaptively encrypted databases for one billing period. Similarly, np, ne, na represent network usage of the three configurations. We assume that the tenant knows the database structure and the query workload and we assume that each column a A stores ra values. By denoting as vp a the average storage size of each plaintext value stored in column a, we estimate the storage of the plaintext database.

5. PERFORMANCE EVALUATION:

This section aims to verify whether the overheads of adaptive encryption represent an acceptable compromise from the performance point of view for guaranteeing data confidentiality in cloud database services. To this purpose, we design a suite of performance tests that allow us to evaluate the impact of encryption and adaptive encryption on response time and throughput for different network latencies and for increasing numbers of concurrent clients. The clever cloud platform we used to calculate the workload and different network latencies.

We consider three databases on the clever cloud platform

- **Plaintext (PLAIN)** is based on plaintext data.
- **Encrypted (ENC)** refers to a statically encrypted database where each column is encrypted at design time through only one encryption algorithm.
- **Adaptively encrypted (ADAPT)** refers to an encrypted database in which each column is encrypted with all the onions supported by its data type.

In the two versions of encrypted databases, each column is set to the highest encryption layer required to support the respective SQL operations on the data stored on the clever cloud platform. During each test ,we monitor the number of executed transactions, and the response times of all the SQL operations . We repeat the test for each database configuration (PLAIN, ENC and ADAPT) for increasing number of student data entry thesis(from 1 to many), and for increasing network latencies. In order to guarantee data consistency, the three databases use repeatable read (snapshot) isolation level.

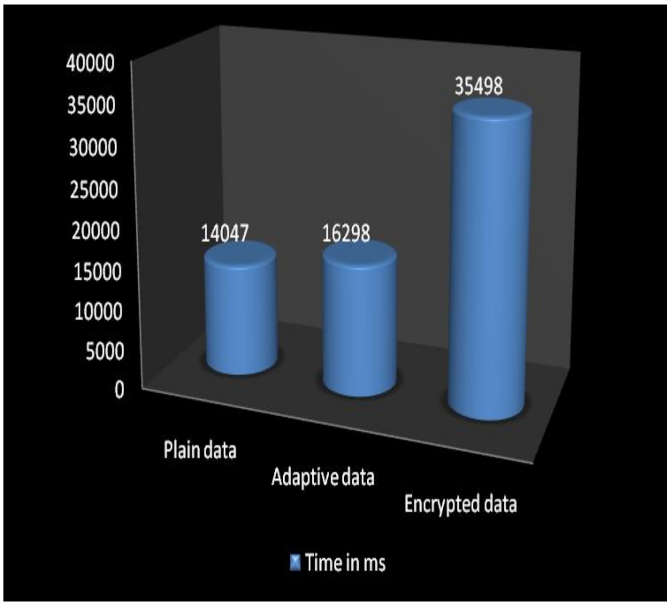

Fig. 4: Encryption times of the SQL operations

We observe that the presented ADAPT configuration represents a worst case scenario that is fully adaptive, because all database columns are encrypted with all the onions supported by its data type.On the other hand, the ENC configuration represents a best case scenario that is completely static, because the user manually defines the single encryption scheme to use on each database column. We observe that a tenant may choose a partially adaptive configuration in which a subset of columns are encrypted through adaptive strategies and others are statically encrypted. As a consequence, performance of adaptive encryption for many realistic workloads fall between the ENC and ADAPT scenarios.

6. RESULTS & CONCLUSION :

There are two main tenant concerns that may prevent the adoption of the cloud as the fifth utility: data confidentiality and costs. This system 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 we consider that almost all important services are based on one or multiple databases. We address 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. Our results demonstrate that the network latencies that are typical of cloud database environments hide most overheads related to static and adaptive encryption. Moreover, we 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 mid-term horizon. By instantiating the model with actual cloud provider prices, we can determine the encryption and adaptive encryption cost of data confidentiality. From the research point of view, it would be also interesting to evaluate the proposed or alternative architectures 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. System,volume. 25, no. 6, pp. 599–616, 2009.

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

[3] H.-L. Truong and S. Dustdar, “Composable cost estimation and monitoring for computational applications in cloud computing environments,” Procedia Comput. Science, volume. 1, no. 1, pp. 2175–2184, 2010.

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

[5] H. Hacig€um€u¸s, B. Iyer, and S. Mehrotra, “Providing database as a service,” in Proceeding. 18th IEEE International. Conference. Data Enggnering., Feb. 2002, pp. 29–38.

[6] G. Wang, Q. Liu, and J. Wu, “Hierarchical attribute-based encryption for fine-grained access control in cloud storage services,” in Proceeding. 17th ACM Conference. Computer and Communications Security, 2010, pp.735–737.

[7] Google. (2014, Mar.). Google Cloud Platform Storage with serverside encryption [Online]. Available:blogspot.it/2013/08/google-cloud-storage-now provides.html.

[8] H. Hacig€um€u¸s , B. Iyer, C. Li, and S. Mehrotra, “Executing SQL over encrypted data in the database-serviceprovider model,” in Proceeding. ACMSIGMODInt’l Conference. Manage. Data, Jun. 2002, pp. 216–227.

[9] L. Ferretti,s M. Colajanni, and M. Marchetti, “Distributed, concurrent, and independent access to encrypted cloud databases,” IEEE Trans. Parallel Distribution. System, volume. 25, no. 2, pp. 437–446, Feb. 2014.

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

[11] C.Gentry,"Fully homomorphic encryption using ideal lattices," in Proceeding. 41st ACM. 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 Proceeding. Advances in Cryptology – CRYPTO 2011.Springer, Aug.2011.

[13] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in Proc. Advances in Cryptology – EUROCRYPT99.Springer, May 1999.