QINCLOUD: AN AGENT-BASED SYSTEM FOR QUERING ENCRYPTED DATA IN CLOUD DATABASES

MASHAEL M. ALSULAMI and AMIN Y. NOAMAN

Department of computer science,
King AbdulAziz University, Saudi Arabia

ABSTRACT

With the rapid growth of technology, cloud computing become more and more popular. Many organizations have been attracted by the variety of services that have been offered by the clouds in form of resources and applications. Database system is one of the most widely used systems in industry. Cloud providers offer database-as-a-service to attract more clients to use their services. However, executing queries against a cloud database is a challenging process since data are stored in encrypted form for security purposes. Cloud database server is responsible of performing the user's query on the encrypted database without any knowledge about the meaning of the requested data to ensure the data confidentiality. Most of the existing methods focus on data confidentially and do not guarantee the performance of their techniques. In this paper, an agent-based system called QinCloud is introduced to execute query efficiently over encrypted data in cloud databases. The proposed system allows users to execute queries on a cloud database server without having any intermediate proxy as a trusted server. Nevertheless, the proposed system is designed to support wide range of queries that are performed completely on the cloud database server side. The proposed system overcomes many issues in existing systems.

KEYWORDS

Cloud Databases, Database as a Service, Security, Query Execution and Data Confidentiality

1. INTRODUCTION

Cloud computing is a promising technology for hosting other important technologies in industry. Many companies and organizations have moved towards cloud computing to grant better performance or to pay less for the infrastructure. Furthermore, users use cloud databases to store and access their data and benefit from other services that provided in the cloud [9].

A cloud database can be defined as a database that can be accessed remotely via the internet from cloud database service provider and application owners pay based on their usage [4]. Customers can plug-in with their own applications to cloud databases. Amazon Aurora, Google Cloud SQL and Microsoft Azure are some examples of cloud database services that are available in the market and offer SQL and NoSql database services. [10]

Natarajan Meghanathan et al. (Eds) : NeCoM, SEAS, CMCA, CSITEC - 2017
pp. 85– 96, 2017. © CS & IT-CSCP 2017 DOI : 10.5121/csit.2017.71207

As a result of the evolution on the area of cloud database services, many issues and challenges have been addressed and been under researches. Cloud database services promises include scalability, availability and elasticity. These promises or some of them are keys for many companies and customers to move toward cloud and reduce the overhead of providing these characteristics by themselves. However, several issues have been raised in order to satisfy these promises. Security, privacy, integrity and data confidentiality are the fundamental concerns in using cloud databases services.

When it comes to store data over cloud databases, clients need to perform certain queries for certain purposes on that cloud database. For security and data confidentiality sake, data are usually stored in encrypted form in cloud databases. The encryption process can be end-to-end encryption or client-side encryption. Many well-defined cloud database vendors provide security guarantee; however, the leak of data could come from the provider itself. Thus, executing queries over encrypted data in a cloud database server is an expensive process in term of performance and security. Some companies focus on the performance of executing their queries with other factors, which can be measured by computing the response time of query execution. On the hands, other companies focus on securing their sensitive data. Many researches have been focused on preserving the data confidentiality when executing encrypted queries over encrypted data in cloud databases. Furthermore, many encryption techniques have been proposed and discussed in the literature to ensure secure and efficient query execution. The ultimate goals for customers to use cloud databases are:

1. Preserving the privacy of their data from any outside (hackers/attacker) leak or inside leak (service provider).

2. Executing queries efficiently in term of computational cost and communication cost.

3. Supporting wide range of queries [5].

The term database-as-a-service refers to a database that is ran on the cloud environment and maintained by a service provider. The structure of cloud databases is complex since data that held in them located in different locations and stored in different data centers. Many companies are moving towards cloud databases [13].

Adopting cloud database has several advantages. First, the main advantage of cloud database is the cost reduction. Users do not have to install specific software to use certain technology. Instead, they can benefit from the capabilities of DBMS without installing one in their local machines. Second, pay-as-you-use feature drives many individuals and companies to use cloud database since they save a lot of money by doing so. Third, companies that are running their databases on the cloud do not have to pay attention to configuration details or maintaining the infrastructure because these consider as provider's responsibilities. Forth, the distribution feature of the cloud allows databases running on it to access variety of information that could be located in different locations. Finally, the most important advantage is the ability of cloud databases to handle big data and perform queries over them [11].

On the other hands, using cloud databases has some disadvantages, which could drive companies to rethink of using cloud environment instead of their local ones. Low security, insured privacy and data confidentiality are the most fundamentals concerns when using services in the cloud [2].

The most challenging issues that face service providers and vendors are related to security. Since database owner does not always control cloud databases but instead a service provider control the cloud DBMS. The focus on recent researches is on preserving the security properties such as confidentiality, integrity and privacy of data in the cloud. Many methods have been proposed to solve issues related to security in the cloud [14].

Data in databases in general should satisfy two aspects: data privacy (confidentiality) and data integrity. Privacy or data confidentiality is the concept of keeping information safe from any disclosure or any unauthorized access. Integrity means that data need to be always valid and correct. More precisely, query integrity means that results returned by database server must be always valid, consistent and correct to ensure that nobody has modified the database except those who are authorized. Furthermore, query results should include all possible records that satisfy the query condition and be the most updated ones [1].

2. RELATED WORK

Many studies have been conducted to discuss the problem of executing queries over encrypted data in cloud databases. Some companies work with sensitive data that are related to their customers like credit card number, annual income and other. Usually, companies try to encrypt these sensitive information if they want to use cloud technology for their business. Literature has discussed this problem from two perspectives: architectural designs and encryption techniques [6].

In architectural perspective, several studies suggest some models and architecture designs to ensure data confidentiality while performing queries against encrypted database in the cloud. In encryption techniques perspective, many research discuss various encryption techniques to ensure the security of data while performing queries.

There are two well-defined systems that are used as solutions to this issue with some limitations in both. These two systems are: CryptDB system and MONMI system. These systems are designed to allow queries to be performed over encrypted data in cloud databases.

CryptDB system has been proposed by Raluca Ada Popa et. al [8]. It mainly works on encrypted databases stored in cloud. The idea of this system based on proxy-based architecture. A trusted intermediate proxy is used between clients and cloud DB server. The proxy is responsible for getting data from database owner as a plain text then encrypts these data and stores them in cloud database. In case of a client publishing a query, the proxy is responsible for getting this query from the client as a plain text then encrypts it and sends it to the cloud database to be executed over encrypted data. Cloud DBMS sends the encrypted results to proxy, which in turn decrypts the results, and sends it back to the client. CryptDB system uses column based encryption algorithm to encrypt both data and query. This encryption algorithm is a powerful one in preserving the confidentiality of the data by encrypting each column in the database using a different key. This technique makes inferring any information from client's queries is a hard task [8].

On the other hands, CryptDB system suffers from three limitations or drawbacks. First, the system suffers from a communication overhead since the existence of proxy as intermediate server between client and DB server, which causes an increase in the communication cost. In this situation, client communicates with the proxy when issuing a query, the proxy communicates

with DB server, then the DB server communicates with the proxy, and finally the proxy communicates back with the client. The second drawback of CryptDB system is the single point of failure, which could make the system hard to scale. The last and critical drawback of the system is that CryptDB doesn't support range and computational queries [7].

MONOMI system is a system that is proposed by [12]. It designed to execute queries over analytical encrypted databases. The idea of MONOMI system is to divide the execution of the query between client side and DB server side. On the other words, a query is analyzed and divided to be executed based on the computations that DB server could handle with no need to decrypt data. Furthermore, MONOMI system executes part of a query in client side and the other part in DB server side to allow more kinds of queries to be executed. One of the biggest contributions of this system is that it could handle 19 out of 22 kinds of queries. However, the drawback of this system includes overhead of splitting queries, poorly use of DBMS capabilities, no security constraints are considered, and finally the cloud DBMS needs to be modified to allow this system to work [7].

As stated previously, many studies have been conducted to solve the issue of executing queries over encrypted data in cloud databases. These solutions focused on the problem from two points of views: architectural designs and encryption techniques.

2.1 Architectural Designs

Many architecture designs have been proposed to find secure and efficient way to execute queries over encrypted data. There are three types of architectures that describe the process of query execution in the cloud databases and preserve the data confidentiality.

The first architecture is a proxy server-based architecture. This type of architecture has been proposed by [12]. The idea of this architecture is to have an intermediate proxy (server) between clients and DB server. The role of this proxy is to perform encryption for both data and queries and to decrypt results coming from DB server. The drawbacks of this kind of architecture include: single point of failure, scalability and consistency.

The second architecture is a proxy server-less architecture with distributed metadata. This type of architecture proposed in [3]. The basic idea of this type is to have a proxy within each client instead of having an intermediate proxy between client and database server. The architecture is based on the elimination of the intermediate proxy to solve the problem of single point of failure. Each client in this architecture has a proxy containing an encryption engine and a maintained copy of metadata. Nevertheless, each client is responsible of encrypting queries sent to cloud database server. Also, each client is responsible of maintaining the metadata after any modification to it. This behavior could result in consistency problem among different copies of metadata especially when multiple clients access the cloud database at the same time.

The third architecture is a proxy server-less architecture with metadata in the cloud. This type of architecture has been proposed by [6]. It designed to solve the inconsistency issue that occurs when concurrent clients access the cloud database at the same time. The idea of this architecture is to store the metadata in the cloud database server. In this case, all clients will access the same copy of metadata and allow for more availability, scalability and elasticity. However, one again this type suffers from single point of failure limitation.

2.2 Encryption techniques

Encryption is a technique that required while using cloud databases to preserve the data confidentiality. Encryption could be performed in the cloud environment from two points of views: End-to-End encryption or Client-side encryption. End to end encryption indicates that cloud database providers are trusted and have keys to decrypt data being transferred or stored on their side. This approach focuses on the security of data while transferring via the Internet. Furthermore, many service providers in cloud environment have depended on this approach of encryption to protect data of their customers. However, this kind of encryption seems to not be applicable with users' needs to protect their sensitive data even from potential threats that could come in the form of curious service providers. Thus, Client-side encryption seems to be the suitable solution for privacy issue. In this approach, data are encrypted by the database owner and stored in the cloud in encrypted format. In this case, the database owner is the only one capable of decryption data or distributing keys over trusted clients. Client-side encryption illustrates the 0-knowledge policy that prevents any sort of unauthorized access to information in the cloud [10].

Generally, encryption techniques can be categorized into two categories: symmetric key encryption (secret key encryption) and asymmetric key encryption (public key encryption). Symmetric key encryption techniques use one master key (secret key) between sender and receiver. Data are encrypted using the secret master key and transferred via the Internet then a receiver decrypts the transferred data using the same master key. The main issue of this technique is the exchanging keys between sender and receiver is not always secured [13].

On the other hands, asymmetric key encryption technique is based on having pair of keys: public and private. Public key is available for anyone wants to send data while private key is used for decryption. The main issue of this technique is the cost of encryption and decryption process [14].

There are various encryption techniques that used by client-side to provide data confidentiality. These techniques used to encrypt data before storing them in the cloud databases and to encrypt queries that need to be performed over encrypted data. Some of these encryption techniques are: *Random encryption schema (RND), Deterministic encryption schema (DET), Homomorphic encryption schema (HOM), Order preserving encryption schema (OPE), Format preserving encryption schema (FPE), Column based encryption schema, and Search encryption schema.*

3. PROPOSED SYSTEM

The proposed system is designed to solve the problem of executing queries over encrypted data in cloud databases efficiently. The main challenges are regarding security in term of data integrity and data confidentiality, and performance in term of communication cost and computational cost. The aims of this system is to reduce the communication cost by using Agent Oriented architecture style and enhance performance by reducing the computational cost while performing computational queries. The motivation of the proposed system is to allow users to perform wide range of queries efficiently over encrypted data in cloud databases.

Assumptions:

- Cloud DB server is not trusted.

- All authenticated clients are trusted.

- Clients are allowed to perform read-only operations such as select.

- DB owner is the only one who is allowed to create tables, alter tables, drop tables, insert, update, and delete records from the cloud database.

The proposed system is shown in figure 1. It is based on agent oriented architecture style. This style has been chosen to benefit from the feature of reducing the communication cost between clients and cloud DBMS and the asynchronous computing feature.

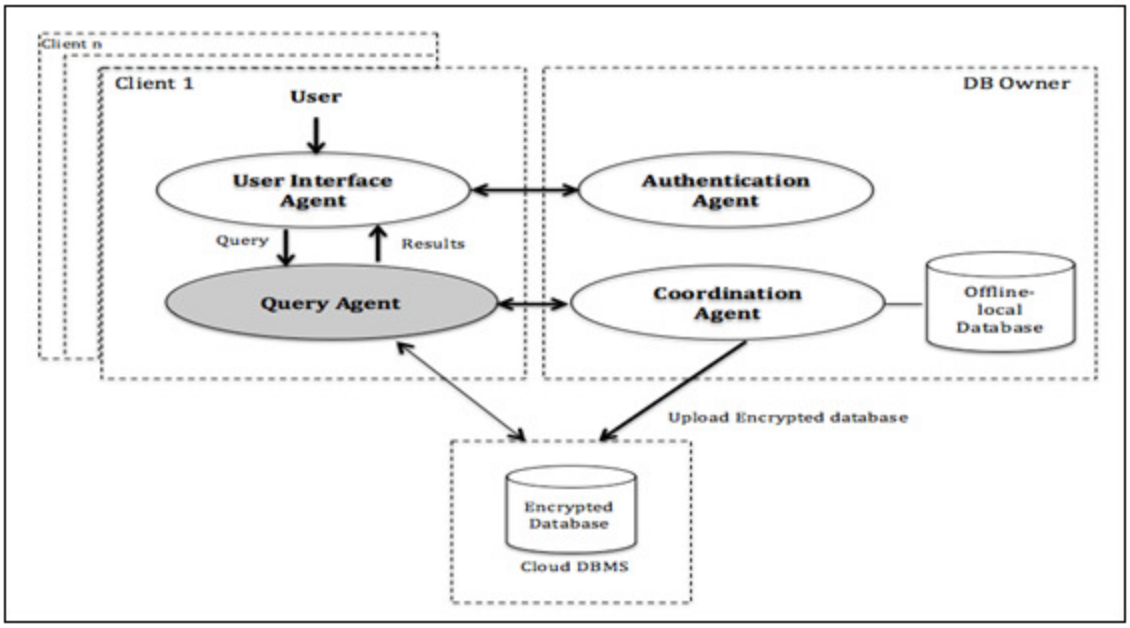


Figure 1: Proposed system

The proposed system consists of the following agents:

User Interface agent:

It is a static agent that is responsible of two main functions. First, it gets the user account information from the user after they log in to the system then determines if he/she is an authenticated user. After a user account has been verified, user can issue a query as a plain text.

Authentication Agent:

It is a static agent. The functionality of this agent is to determine if a user is authenticated or not by checking his/her account information.

Query Agent:

It is a mobile agent. It is responsible of the following functions:

- Encrypting query using the public key that has been published by DB owner.

- Sending the encrypted query to the cloud DBMS to be executed.

- After getting the encrypted results from the cloud DB server, Query agent travels to coordination agent to get the results decrypted.

- After the coordination agent decrypts the results, the results will be forwarded to the query agent.

- It forwards the decrypted result back to the user interface agent.

Coordination Agent:

This agent is responsible of the following functions:

- Publishing public keys for all clients.

- Storing private key in a local offline database.

- Decrypting any encrypted results that are coming from Query agent.

To support wide range of queries, coordination agent in the DB owner side and query agent in the client side use a proposed encryption schema. It consists of two main encryption algorithms: **Algorithm #1** for database creation and **Algorithm #2** for query execution. Database owner encrypts the whole database by using Algorithm #1 before storing it in cloud database server. Each authorized client uses Algorithm #2 to encrypt their queries before sending them to the cloud database server. Each client needs to have an account to use the system. Database owner has two agents: Authentication agent and coordination agent. As stated previously, Authentication agent checks if the logged client is who he/she claims to be. The Coordination agent is the main agent in the database owner side. It is responsible of getting the encrypted results of a query from Query agent then it uses a private key that is stored in a local- offline database to decrypt the results and forward it back to the Query agent. Figure 2 shows a sequence diagram of the proposed system.

Database creation encryption algorithm is shown in figure 3. It is used only once by the database owner to encrypt all tables in a database before storing them in the cloud database server. The main concept of this algorithm is to encrypt each column based on its data type. This mechanism allows applying operations over some columns in a database.

On the other hands, client who issues a query performs query execution algorithm that is shown in figure 4. This algorithm allows each client to encrypt his/her query using the public key that has been published by database owner.

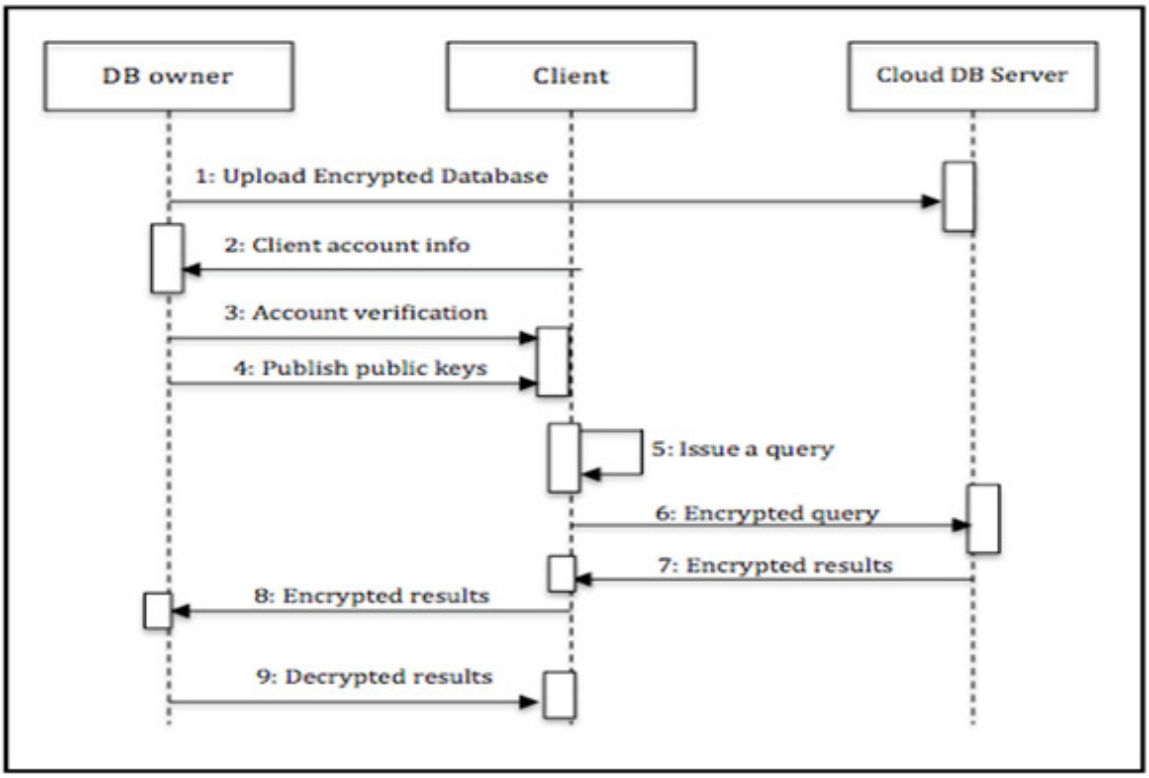


Figure 2: Sequence diagram of QinCloud

Algorithm #1: Database Creation (Setup)

Input: Database schema *S*, $S=\{R_1, R_2, \ldots\ldots, R_n\}$ where *R* is a relation in a database and *n* is the number of the relations in S, public key *k*

Output: Encrypted database schema, $S_E = \{R_{1E}, R_{2E}, \ldots\ldots, R_{nE}\}$ where R_E is the encrypted form of R.

Steps:

1. For every relation R, I created R_E where:
   $R = \{a_1, a_2, \ldots\ldots, a_n\}$ and $R_E = \{Enc_k(a_1), Enc_k(a_2), \ldots\ldots, Enc_k(a_n)\}$
2. Each attribute ***a*** in ***R*** is evaluated based on its data type:

```
if( a is a numerical value) then
  {
   Enc_a= a *432 + 9403;// Applying the concept of OPE
   Insert Enc_a into RE;
  }
 else
  {
   Enc_a = AES (k , a);
   Insert Enc_a into RE;
  }
```

Figure 3: Database creation algorithm

Algorithm #2: Query Execution

Input: Query *Q*, Command file *cmdfile*, operations file *opfile*, and symbol file *symbfile*
Output: Encrypted Query *cipherQuery*

Steps:

1. Client issues a query *Q*.
2. Q is tokenized into n tokens.
3. **For** each token *i* **in** *Q* **do**

```
if ( i ∈ cmdfile OR i ∈ opfile OR i ∈ symbfile ) then
        cipherQuery.write(i);
else if(i is a numeric value) then
        {
          encryptedi = i  *432 + 9403;
          cipherQuery.write(encryptedi);
        }
else
        {
          x = encryptAES(k , i);
          cipherQuery.write(x);
        }

End of for
```

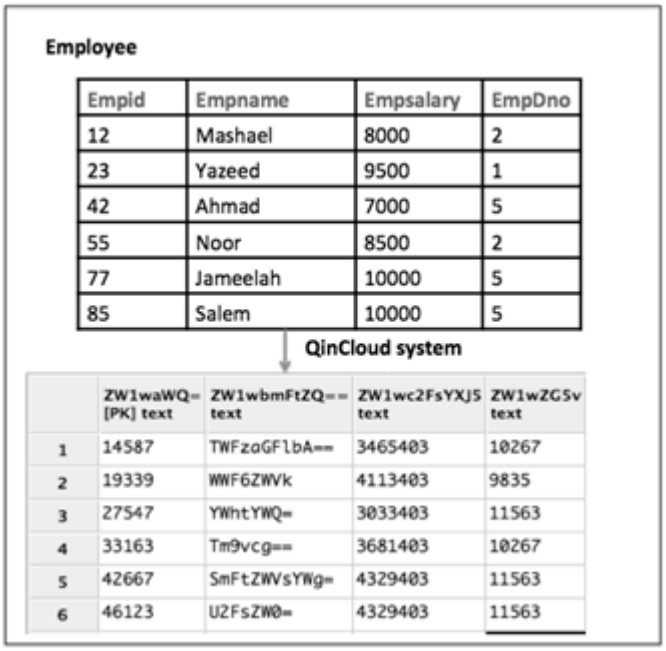
Figure 4: Query Execution algorithm

4. EVALUATION

The proposed architecture is built to enhance the performance of executing queries over encrypted data in cloud databases with preserving the data confidentiality. The benefits of this architecture are as follows:

1. It doesn't require any modifications to the cloud DBMS.
2. All computations are done on the cloud DBMS side.
3. Clients know nothing about how to decrypt the results of the query.
4. All public keys are hidden within the implementations but they are used by the system.
5. The proposed encryption schema uses multiple layer of encryption (AES, OPE and FPE).

To test the proposed system, a simple implementation of QinCloud has been designed using Java as programming language, postgreSQL JDBC Driver as a DBMS, Java cryptographic extension as an encryption framework, and finally Windows builder framework. Figure 5 shows a tested table before encryption and the its encrypted form.



Employee

Empid	Empname	Empsalary	EmpDno
12	Mashael	8000	2
23	Yazeed	9500	1
42	Ahmad	7000	5
55	Noor	8500	2
77	Jameelah	10000	5
85	Salem	10000	5

QinCloud system

	ZW1waWQ= [PK] text	ZW1wbmFtZQ== text	ZW1wc2FsYXJ5 text	ZW1wZG5v text
1	14587	TWFzaGFlbA==	3465403	10267
2	19339	WWF6ZWVk	4113403	9835
3	27547	YWhtYWQ=	3033403	11563
4	33163	Tm9vcg==	3681403	10267
5	42667	SmFtZWVsYWg=	4329403	11563
6	46123	U2FsZW0=	4329403	11563

Figure 5: Sample table

Figure 6 shows a comparison between the execution time of the encrypted query and unencrypted one. It is noticeable that the encrypted query takes a few more msec to be executed than the unencrypted on. However, in some cases such as count operation, the performance is constant since there is no need to deal with the data itself rather counting rows in both tables.

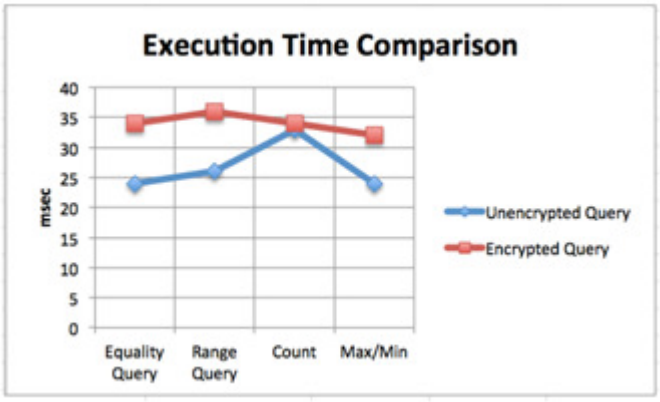


Figure 6: Execution time comparison

5. CONCLUSION AND FUTURE WORK

Recent researches discuss the problem of executing queries over encrypted data from two points of views: data confidentiality and performance. To enhance the performace, three factors need to be taken into considerations: communication cost ,computational cost, and encryption cost. The proposed system has been designed to enhance the performance by reducing the communication cost by using agent oriented architecture. To reduce the computtional cost, a proposed encryption

schema has been used to support equality, count, max, min and range queries and to reduce the response time of executing queries. The future works include:

- Use JADE framework to design the system as agents.
- Enhance the encryption schema to support SUM, AVE and MULT operations.
- Reduce the overhead of the decryption process.
- Use the 22 queries suggested in TPC-H benchmark to test the performance of the proposed system.

REFRENCES

[1] Ghazizadeh, Puya, Ravi Mukkamala, and Stephan Olariu. "Data Integrity Evaluation in Cloud Database-as-a-Service." IEEE Ninth World Congress on Services, 2013. Accessed 26 Nov. 2016.

[2] Ghobadi, Alireza, Roozbeh Karimi, Farnaz Heidari, and Masoud Samadi. "Cloud computing, Reliability and Security Issues." ICACT2014, 2014.

[3] H. Hacigu¨mu¨ s¸, B. Iyer, C. Li, and S. Mehrotra, “Executing SQL over Encrypted Data in the Database-Service- Provider Model,” Proc. ACM SIGMOD Int’l Conf. Management Data, June 2002.

[4] Joshi, V., & Patil, S. SecureDBaaS Architecture For Encrypted Cloud Database. International Journal of Computer Applications, 5(4), 2015

[5] Kumar, R. R., & Hussain, M. Query Execution over Encrypted Database. Second International Conference on Advances in Computing and Communication Engineering, 2015

[6] Luca Ferretti, Michele Colajanni, and MircoMarchetti: Distributed, Concurrent, and Independent Access to Encrypted Cloud Databases. IEEE Transactions On Parallel And Distributed Systems, Vol. 25, No. 2, February 2014.

[7] Munir, Kashif. "Security Model for Cloud Database as a Service (DBaaS)." IEEE, 2015.

[8] Raluca Ada Popa, Catherine M. S. Redeld, Nickolai Zeldovich, Hari Balakrishnan ”CryptDB: Protecting Confidentiality with Encrypted Query Processing”,Twenty-Third ACM Symposium on Operating Systems Prin- ciples,October 2011.

[9] Raluca Ada Popa, Frank H. Li, and Nickolai Zeldovich, ”An Ideal- Security Protocol for Order-Preserving Encoding”, In the Proceedings of 34th IEEE Symposium on Security and Privacy (IEEE S&P/Oakland) , May 2013.

[10] Refaie, R., Ahmed, A., Hamza, N., Al-monem, M., & Hefny5, H. A secure Algorithm for Executing Queries over Encrypted Data. IEEE, 2015.

[11] Sonali, J., & Patil, B. M. Integrating Encrypted Cloud Database Services using Query Processing. International Journal of Computer Applications, 148(12), 2016.

[12] Stephen Tu, M. Frans Kaashoek, Madden.S and Zeldovich.N. " Processing Analytical Queries over Encrypted Data". In Proc. of the 39th International Conference on Very Large Data Bases (VLDB), Riva del Garda, Italy, August 2013.

[13] Syed, Sadia, and M Ussenaiah. "The Rise of Bring Your Own Encryption (BYOE) for Secure Data Storage in Cloud Databases." IEEE Second International Conference on Multimedia Big Data, 2015. Accessed 28 Nov. 2016.

[14] Waghmare, Vivek, Kaveri Gojre, and Akshaya Watpade. "Approach to Enhancing Concurrent and Self-Reliant Access to Cloud Database: A Review." International Conference on Computational Intelligence and Communication Network, 2015. Accessed 29 Nov. 2016.