INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 2)
Available online at: www.ijariit.com

Two clouds secure database for numerically related SQL range queries with privacy preservation

Sruti S
sruti.senthil@gmail.com
RMK Engineering College, Kavaraipettai, Tamil Nadu

Shruthi S
shruthimani17@gmail.com
RMK Engineering College, Kavaraipettai, Tamil Nadu

Varsha S
varsha31.7.vp@gmail.com
RMK Engineering College, Kavaraipettai, Tamil Nadu

Radhika S
sra.cse@rmkec.ac.in
RMK Engineering College, Kavaraipettai, Tamil Nadu

ABSTRACT

In the present scenario, businesses and people are outsourcing database to minimize their effort and to obtain a low cost service. In order to provide the sufficient functionality for SQL queries, many secure database schemes have been proposed. However, such schemes are vulnerable to privacy leakage to cloud server. For numerical range inquiry (>, <, and so forth.) these neglect to give adequate security insurance. A portion of the difficulties faced are privacy leakage of statistical attributes, access patterns and so on. Likewise increased number of queries will release more information to the cloud server. We have studied some of these research works and analyzed the best possible ways to come to the desired level of privacy preservation in the case of cloud computing. We have proposed a two-cloud architecture for secure database, with a series of intersection protocols that provide privacy preservation to various numeric-related range queries. Security analysis shows that privacy of numerical information is strongly protected against cloud providers in our proposed scheme.

Keywords: *Database, Range query, Privacy preserving, Cloud computing.*

1. INTRODUCTION

In the present circumstances, it can be seen that the cloud has taken the control over the IT business with its innumerable advantages. Cloud computing is alluded to as SaaS (Software as a Service) since it renders the applications as administrations over the Web and the hardware and systems software in the data centers that offer those administrations. The hardware of data centre and software is called a cloud. Today the clouds can be open/public and in addition private. Private clouds are associated to the inner datacenters of a business or other association, not made accessible to the overall public. Cloud computing in this manner can be compressed as a blend of SaaS and utility computing, booting out the data centre (little + medium estimated).

Security is the chief concern of the cloud computing. Cloud clients confront security dangers both from outside and inside the cloud. Due to the privacy concerns, the cloud service provider is assumed semi-trust (honest-but-curious.), it becomes a critical issue to put sensitive service into the cloud, so encryption or obfuscation are needed before outsourcing sensitive data - such as database system.

A cloud client, such as an IT enterprise, wants to outsource its database to the cloud, which contains valuable and sensitive information (e.g. transaction records, account information, disease information), and then access to the database (e.g. SELECT, UPDATE, etc.). Due to the assumption that cloud provider is honest-but curious, the cloud might try his/her best to obtain private information for his/her own benefits. Even worse, the cloud could forward such sensitive information to the business competitors for profit, which is an unacceptable operating risk.

Shielding the information from the server itself is the pro of the main issues related with it. The server will by definition control the "bottom layer" of the product stack, which adequately goes around most known security methods.

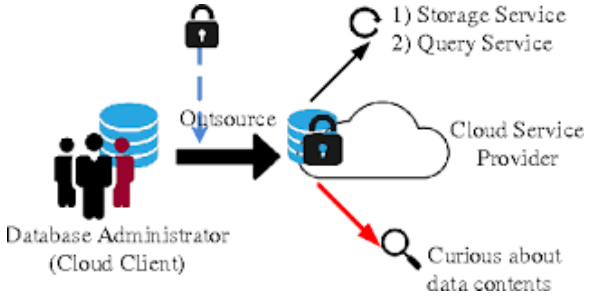


Fig.1.Outsourced database, service and privacy risk

One straightforward approach to mitigate the security risk of privacy leakage is to encrypt the private data and hide the query/access patterns. CryptDB is used for such purpose.

CryptDB, a framework that gives confidentiality to applications that utilize database administration frameworks. CryptDB permits to perform queries over encrypted data, likewise the SQL's very much characterized set of operators, and queries over encrypted data.

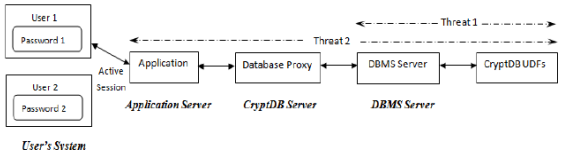


Fig.2. Crypt DB architecture

CryptDB tends to the hazard of an inquisitive database administrator (DBA) who endeavors to learn private information (e.g., health books, financial articulations, individual data) by keeping an eye on the DBMS server by keeping the DBA from learning private information. It uses a few instruments to accomplish this security functionality. One of the devices being the Order preserving encryption (OPE) is generally utilized as a part of databases to process SQL Queries over encrypted information. It permits to perform order operations on ciphertext like the plaintext for e.g. data server can fabricate index and sort the encrypted information like the plaintext. Regardless of going to the security reason well, despite everything, it uncovers the order of the ciphertext, the derivation of the statistical properties such as the data distribution and the access pattern. Therefore the objective of security protection of the outsourced information to a cloud server is refined by partitioning the sensitive knowledge into two parts and store them in two non-colluding clouds. Moreover a secure database service architecture is acknowledged by utilizing two non-colluding clouds in which the information learning and query rationale is divided into two clouds. Henceforth, perceiving just a single cloud can't help uncover private data. Other than a progression of intersection protocols to give numeric-related SQL range queries with privacy preservation is additionally executed and it won't uncover order related data to any of the two non-colluding clouds.

1.1 MOTIVATION

Privacy is most vital factor in cloud and modern day data storage services. Many creators took a shot at security protection, yet private data can't be fully protected by some technique. All the enterprises and organizations has numerous private information, they don't impart the information about this to anybody. If any of the information is leaked the organization's misfortune is sure shot. With the goal that we are turning on protection of the sensitive information. Present day innovation additionally takes a shot at privacy preservation in the cloud servers.

2. RELATED WORK

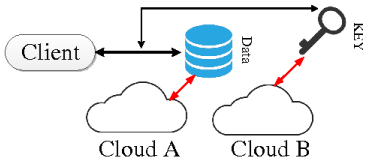
Fuzzy searchable encryption deal with the issue that search keywords allows small-scaled distinction in character/numeric level. Specifically for numerical keywords, the query predicate can get numerical records within a range.

John Daugman, and Piotr Zielinski have proposed a fast search algorithm for a large fuzzy database that stores iris codes or data with a comparative binary structure. The hazy nature of iris codes and their high dimensionality is handled by the novel procedure, Beacon Guided Search (BGS), which does so by dispersing a large number of "beacons" in the search blank. BGS is considerably quicker than the present ES with an immaterial loss of precision. It takes substantially less memory and it doesn't rely upon caching data in storage, in this way murdering the requirement for complex storage administration. The preprocessing is basic and brisk. It holds up to 30% bit errors in the query and also up to seven cyclic rotations. The abundance memory put is little and promptly affordable– it bolsters dynamic upkeep, empowering simple ordering of new books. Yin Yang, Hongwei Li, Mi Wen, Hongwei Luo, and Rongxing LuSS, proposed a ranked range query (RRQ) scheme, which can bolster both range query and ranked search. Established on the homomorphic Paillier cryptosystem, we utilize two super-increasing sequences to total multidimensional keywords. The first one is used to total one purchaser's or vendors multidimensional keywords to a collected number. The second one is connected to make a synopsis number by amassing the accumulated quantities of all sellers. Security investigation exhibits that RRQ can accomplish confidentiality of keywords, confirmation, information trustworthiness and query privacy. In any case, in the meantime more intricate pre-filtering rules, for example, "and", "or", "not" isn't finished by RRQ strategy. R.A.Popa, C. Redfield, N.Zeldovich and H.Balakrishnan proposed CryptDB, a framework to defend the private information in databases from firstly the inquisitive cloud server itself and secondly the application server's bargains. CryptDB fundamentally includes utilizing the range queries productively finished the encrypted information utilizing a novel SQL-aware encryption system. It limits the data uncovered to the untrusted database server. Regardless of satisfying the assignment of protection safeguarding, still a few information is

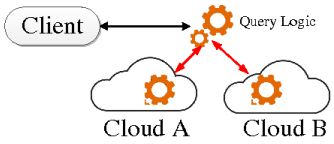
uncovered in the process. Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, Yirong Xu proposed Order Preserving Encryption for Numeric Data[11] that enables any comparison operation to be straightforwardly connected on encrypted information. Query results produced are sound (no false hits) and complete (no false drops). OPES (Order Preserving Encryption Scheme) enables comparison operations to be specifically connected on encrypted information, without decrypting the operands. Accordingly, balance and range inquiries and also the MAX, MIN, and COUNT, GROUP BY and ORDER BY queries can be specifically prepared over encrypted data. OPES results are correct and don't contain false positives , a value in a column can be modified or a new value can be inserted in a column without requiring changes in the encryption of other values and it can be effortlessly incorporated with existing database frameworks. Encryption of non-numeric information, for example, factor length strings aren't finished by OPES. Also while applying SUM or AVG to a group the values should be decrypted. Raluca Ada Popa, Frank H. Li, Nickolai Zeldovich, proposed "An Ideal-Security Protocol for Order-Preserving Encoding", which accomplishes perfect security. The fundamental method utilized is variable/mutable ciphertexts,which implies, the ciphertexts for few plaintext values change and its demonstrated that impermanent ciphertexts are required for perfect security. mOPE is superior to anything OPE scheme by 1-2 requests of extent. The same-time OPE security (sTOPE) executes such that only the order of items present in the database is known. mOPE and stOPE utilize Merkle hashing to secure clients against a malevolent server. Although leak the order information of the data in plaintext. Furthermore the prototype issues only single query at a time where more finegrain ordering is possible. J.-M. Bohli, N. Gruschka, M. Jensen, L. L. Iacono, and N. Marnau, proposed the Security and privacy-enhancing multi cloud architectures [8], This paper works as an overview paper where creators talked about the security in open cloud and multiple cloud. Also the high potential for security prospects in cloud computing have been discussed. Homomorphic encryption and secure multiparty calculation protocols to be exceptionally encouraging regarding both technical security and regulatory compliance. However there is no single ideal way to deal with cultivate both security and legal compliance in an omni applicable way. The confinements of these methodologies just originate from their restricted applicability and high multifaceted nature being used. M. A. AlZain, E. Pardede, B. Soh, and J. A. Thom, proposed the Cloud computing security from single to multi-clouds, It indicates security in single cloud and multiple clouds [13]. Additionally demonstrates some limitation and points of interest in security in cloud computing. Single clouds work on three phases SaaS,PaaS,IaaS. Clients and business organizations don't lose their private data because of vindictive attacker in the cloud. It has a high capacity to diminish security chances that influence the cloud computing client. Find conceivable to conceivable confinement. However the service availability is still a disappointment and also there is a loss of administration accessibility.

3. SYSTEM ARCHITECTURE

Our proposed secure database system includes a database administrator, and two non-colluding clouds. In this model, the database administrator can be implemented on a client's side from the perspective of cloud service. The two clouds (refer to Cloud A and Cloud B), as the server's side, provide the storage and the computation service. The two clouds work together to respond each query request from the client/authorized users (availability). For privacy concerns, these two clouds are assumed to be non-colluding with each other, and they will follow the intersection protocols to preserve privacy of data and queries (privacy).In our scheme, the knowledge of stored database and queries is partitioned into two parts, respectively stored in one cloud. The mechanism guarantees that knowing either of these two parts cannot obtain any useful privacy information. To conduct a secure database, data are encrypted and outsourced to be stored in one cloud (Cloud A), and the private keys are stored in the other one (Cloud B).



(a) Knowledge Partition of Stored Data



(b) Knowledge Partition of Query Logic

Fig.3. Two Cloud Database and Architecture

For each query, the corresponding knowledge includes the data contents and the relative processing logic. We utilize a prototype of knowledge partition, dividing application logic into two parts, which is firstly proposed by Bohli et al. in. The application logic, as a secret knowledge, is partitioned into two parts, each of which is only known to one cloud. Intuitively, this two-cloud architecture increases some complexity to some extent, and we will analyze and point out that this overhead is acceptable.

4. PRELIMINARIES AND DEFINITIONS

A. Paillier Cryptographic Algorithm:

There are various cryptographic techniques to support numeric-related operations (e.g. addition, multiplication, XOR) upon the encryption field. Paillier cryptosystem [41] is one of the most popular techniques that provides addition homomorphic, which means:

if two integers a and b are encrypted with a same key k into two ciphertexts (be denoted as Ek(a) and Ek(b)), there exists an operation (refer to as "⊗"), such that Ek(a)⊗ Ek(b)= Ek(a+b). Paillier cryptographic algorithm is composed of the following phases: key generation, encryption and decryption. ◦ Key generation. Two large and independent prime numbers p and q are randomly selected. Then we compute n = p·q and μ=λ−1 mod n, whereλ is the least common multiple of p and q, and commonly λ = lcm(p −1,q −1). The public key (PK) isn, and the private key (SK) is(λ,μ). ◦ Encryption. Let m be the integer to be encrypted. Firstly, we select a random number ∈ Z∗ n2, and then the ciphertext of m can be computed as follows: E(m;r)=(n+1)m ·rn mod n2.

- Decryption. Let the ciphertext c = E(m;r). The plaintext m can be recovered as follows: m =(cλ mod n2) −1 n ·μ mod n.
- Paillier cryptosystem holds additive homomorphic in group Z+ n , which corresponds to the multiplication operation in the encryption field in Zn2. The following equation illustrates the homomorphic property of Paillier cryptosystem. E(m1;r1)· E(m2;r2) = (n+1)m1rn 1 ·(n+1)m2rn 2 = (n+1)m1+m2(r1 ·r2)n = E(m1+m2;r1 ·r2)
- Another property can be be summarized as follows: Em2(m1;r1) = ((n+1)m1rn 1)m2 = (n+1)m1·m2(r1m2)n = E(m1 ·m2;rm2 1).
- As the random number r does not affect the result of decryption in Paillier encryption, Eq. (4) can be seen as the product of m1 and m2 in the encryption field. In the rest of this paper, we use E(m, PK) to denote the encryption result of the plaintext m with PK, and D(X, SK) to denote the decryption result. We use capital letters like "X" to denote encrypted results (ciphertext), and lowercase letters like "x" to denote unencrypted results (plaintext). The random number r ∈ Z∗ n is omitted in the discussion of our scheme. For number comparison, the sign of an plaintext number in Paillier cryptosystem is defined as follows: each participated plaintext integer x is assumed to be x < n/2. Then for clarity, the sign of x is defined to be positive if 0 < x < n/2, and the sign is defined to be negative if x > n/2. As a result, the arithmetic subtraction of arbitrary two integers (xi−xj) will not exceed the threshold n/2 ifxi > xj, and the subtraction will exceed n/2 ifxi < xj.

B. Numeric-Related SQL Queries:

The Structured Query Language (SQL) is a specified purpose programming language, which is used to manage data in a relational database system, which has become a standard of the ANSI and ISO in 1986 [42] and 1987 [43] respectively. A query operation can request arbitrary data with a statement to describe the desired data. The requested data can be several columns of one or more tables in the database, and it can also be aggregated results from the original data (such as sum, average, and count of the datum.). To obtain the desired data, the query contains some statements to describe the requirement, e.g.somenumeric-related (">,""<," "=," "BETWEEN," etc.). For clarity, we refer to those query requests as numeric related SQL queries in the rest of the paper. Based on the introduced two-cloud architecture, we further propose a series of interaction protocols between the client and the two clouds, which can realize numeric-related SQL queries, and satisfy privacy requirements. It should be noted that, apart from the query operation, there are other SQL operations (e.g. update, insert) which modify the data. The privacy issue for such cases can be resolved with other existing approaches, such as ORAM (Oblivious RAM) which is beyond the scope of our paper. In this paper, we focus on implementing query operation with privacy preserving.

5. OUR PROPOSED TWO-CLOUD SCHEME

In this section, we firstly give an overview of our proposed two-cloud scheme, and then present the detailed interaction protocols to realize range query with privacy preservation on outsourced encrypted database.

A. Overview

In our scheme, two clouds (refer to Cloud A and Cloud B, respectively) have been assigned distinct tasks in the database system: Cloud A provides the main storage service and stores the encrypted database. Meanwhile, Cloud B executes the main computation task, to figure out whether each numerical record satisfies the client's query request with its own security key. With the assumption of no collusion between two clouds, the knowledge of application logic can be partitioned into two parts in our proposed scheme, where each one part is only known to one cloud. As we will analyze in this paper, one single part of knowledge cannot reveal privacy of the data and the query. Based on the two-cloud architecture, our scheme provides an approach to query numeric-related data with privacy preservation. The client can retrieve the desired data from the cloud, when the query predicates contain operators like ">," "<" and "BETWEEN" for one column, or even diverse condition combinations over one or more columns. For example, the client wants to retrieve items from the table, whose column Ti should be greater than a constant a (i.e., SELECT ∗ FROM table WHERE Ti > a). In our scheme, it is resolved by figuring out the sign of each value of (Ti(j)−a), in which j traverses all rows of the whole table. If the result is greater than 0, the relevant item satisfies the query predicate. These procedures are executed in the encryption field, so that the privacy is strongly preserved. Meanwhile, each column name Ti must be encrypted. Accordingly, if the operator is reversed, i.e., the predicate becomes "Ti < a," the corresponding operation is (a−Ti(j)). The remaining phases are similar as the above mentioned case. Meanwhile, if the predicate is "BETWEEN a and b"(SELECT * FROM table WHERE Ti BETWEEN a AND b), the result is the intersection of Ti > a and Ti < b. For the predicate "=a," it is treated as a special case of the operator "BETWEEN," where the retrieved items are intersection set Ti > a−1 and Ti < a+1. Additionally, the operator of COMBINATION is another one that combines predicates with boolean expression with ∨ and ∧. In Section V-B, we first present the intersection procedure of the first case(">"). Then in Section V-C we give some necessary introductions about "<," "BETWEEN," "=" and COMBINATION. The proposed mechanism can preserve the privacy of data and query requests against each of the two clouds. Specifically, Cloud A only knows the query request type and the final indexes, but due to dummy items appending, Cloud A cannot accurately understand the finally satisfied index set for each single request. Meanwhile, in order to prevent Cloud A from launching multiple specific-purpose query requests to deliberately seek more knowledge about the data, we introduce a token based scheme, which can restrict the number of items and the range of columns that Cloud A can only process. For Cloud B, it knows the satisfied indexes of each single request, but after the proposed operations, it does not know the relationship of the corresponding items.

Moreover, Cloud B can hardly distinguish whether two received columns are generated from one or more columns in the original database.

B. The Basic Scheme for Operator ">" As mentioned above, in our scheme, Cloud A permanently stores the client's encrypted database, and it also keeps the public keys related to the encrypted items in the database. Cloud B keeps the relevant private keys and undertakes the main task of computation. Our proposed scheme is composed of Table Creation and Query Protocol.

1) Table Creation: After the client rents the cloud service, he/she will outsource the database application to the cloud. To protect the private information, the following procedure is implemented before uploading to the cloud: For each column of the table (column in the table), the client randomly selects a symmetric key K, and then use it to encrypt each column name Ti ($1 \leq i \leq m$, where m is the total column number of the table). The encrypted name is denoted as E(Ti), assumed with equal length. The symmetric key K should be securely kept by the client. For each item (row in the table), its values in multiple columns should also be encrypted. In this paper we only take into consideration the numeric-related data. The client generates a public/private key pair for Pallier cryptosystem, denoted as PK and SK. For each numeric-related value x, the client uses PK to encrypt it as follows: X = E(x, PK), and the client should record the total item number of the table N. Then, the encrypted table is uploaded to Cloud A, as well as the public key PK. Meanwhile, the private key SK will be securely sent to Cloud B. Without loss of generality, we take only one table for example in this paper. For multiple tables in a database, table names can be encrypted in the same way that column names are encrypted.

2) Query Request: When the client wants to retrieve some data from the outsourced database, he/she firstly generates a SQL query (e.g. "SELECT ∗ FROM table WHERE Ti > a"). After the plaintext query request is generated, it will be modified to an encrypted query following these steps: Encrypt the column name. The client computes the column name E(Ti) with the symmetric key K. Encrypt the range boundary value. The client encrypts the range boundary value a with the public key PK in Pallier cryptosystem. The encrypted boundary value is denoted as "A," as shown in Fig. 4. Generate the token. The client analyzes the query request and figures out how many columns are involved. Then, the client generates the corresponding token Sign(TNO||CN||N||T), whereTNO is the token serial number, and CN is the number of involved columns, N is the total item number in the table, and T is the current timestamp. All these data are signed by the client's private key SK. Send the query request.Then the client sends the encrypted query request to Cloud A as follows: SELECT ∗ FROM table WHERE E(Ti)>A, together with the signed token. Here, for the above SQL query, the specific column number CN is "1".

3) Item Send: Cloud A finds the column named E(Ti). Before sending the items to Cloud B, it implements the following three phases: Number Comparison (2 in Fig. 4). For each item X j =Tijin the column, Cloud A selects a random positive integer rj and j individually, where 0≤j < rj, and then computes: Xj = X j A rj · E(−j, PK). (5) With the additive homomorphic property of Paillier cryptosystem, the decryption result of Eq. (5) is equal to (xj − a)·rj −j. As the integer rj is positive and not too large, the values of (xj−a)·rj−j and xj−a have the same sign. All X j (j ∈ indexes of items in the column.) are stored in another temporary column (named L). Items shuffling (3 in Fig. 4). Cloud A further makes a random item shuffling in the column L to generate a new column L.To be noted, Cloud A should securely store the mapping of the items between the shuffled column L and the original column L in a new column M. Finally, Cloud A removes the column name E(Ti) from the column L,and sends it to Cloud B together with the token received from the client.

4) Index Send: After receiving the column and the token from Cloud A, Cloud B firstly verifies the legitimacy of the received token to make sure it hasn't expired and hasn't reused in a specific time interval. Then Cloud B checks the column from A to make sure that the column number and the item number are consistent with these corresponding values in the token. If the request is authorized, then Cloud B decrypts each item as follows: x j = D(Xj,SK), (6) where j belongs to the item indexes in the column. For each decrypted item x j,ifx j > 0, the index j is inserted into a new index array L . Additionally, from the aspect of privacy preservation, then Cloud B appends a certain number of dummy indexes and inserts them to the random positions of the new index array L. Finally, Cloud B returns the final index array L to Cloud A.Query Response: For each item j in the received index column L.Cloud A looks up the index mapping information column M, and gets its corresponding index j in the original column. According to the mapped index j, CloudA sends the corresponding rows in the table, as the query response, to the client. After receiving the response, the client can decrypt the items with SK to obtain the required data, and removes dummy items that does not satisfy the query predicate.

6. SECURITY ANALYSIS

In this section, we will focus on the privacy preservation in the outsourced query processes against two honest-but-curious clouds..

Security Proof Theorem 1: Cloud A cannot obtain any information from the user's query and the stored encrypted database as long as Paillier cryptosystem is semantically secure, and Cloud A and B are non-colluding. Proof:In these steps, since all the data received by Cloud A is encrypted and the computation steps are all performed in the ciphertext domain, and because of the semantic security of Paillier cryptosystem ,CloudA cannot deduce any private information from these three steps unless Cloud B colludes with it.

Theorem 2: Cloud B cannot infer any private information from Cloud A's input as long as blinding factors are properly generated, and Cloud A and B are non-colluding.

C. Privacy Preservation in Repeated Queries:

The clouds could collect more and more statistical information after receiving repeated query requests and generating the corresponding responses towards the database (e.g. Fig. 3). However, we will demonstrate that our scheme can reduce the privacy leakage greatly in this scenario. 1) For Cloud A: Repeated query requests will make Cloud A learn more and more about the privacy information, while in our scheme, this ability is restricted as follows. On one hand, many query requests are crossing over multiple

columns, and simple query requests are just a part of usual database query requests. In such a situation, Cloud A only receives the final index result (with dummies, optionally) from Cloud B filtered with multiple conditions, it cannot get the original comparison result of each one column. On the other hand, Cloud B responds Cloud A based on the token obtained from the client, and there have two ways to guarantee the security.

1) Each token contains the specific column number (CN) and the total item number in the table(N), which Cloud A must operate on exactly. Cloud A must send the result to Cloud B exactly with these two numbers without modification: If Cloud A increases or decreases CN or N, Cloud B will find that unmatched with the token, and if Cloud A replaces any item in these CN columns, it will take the risk of responding wrong result to client, which can be assumed not happening based on the assumption that semi trusted clouds are honest. 2) Each token has been signed with SK by client, Cloud A cannot modify any tokens or generate a new one, and every token contains a different serial number and timestamp, so Cloud A cannot conduct the replay attack.

2) For Cloud B: The name of each involved column is removed before sending to Cloud B, and meanwhile, different random integers are selected for each item in each query request by Cloud A. As a result, Cloud B cannot distinguish whether two previous query requests are on the same column,hence repeated queries cannot be utilized to increasing the accuracy of order guessing. Moreover, based on item shuffled, Cloud B cannot distinguish one same item from two previous queries, even though the plaintext SQL queries are identical.

7. CONCLUSIONS

In this paper, we have studied the various techniques and protocols associated with the privacy preservation of the outsourced data to the external cloud server. In order of the advance in this field some of the works include the fuzzy logic, range queries, order preserving encryption and multi cloud architecture. The fuzzy logic implemented the, Beacon Guided Search (BGS), which requires substantially less memory and no complex storage mechanism. The Range Queries operate by implementing the RRQ can accomplish confidentiality of keywords, confirmation, data integrity and query privacy. Then came the CryptDB which fundamentally includes utilizing the range queries productively finished the encrypted information utilizing a novel SQL-aware encryption system. However some data is still exposed to the cloud server. The order preserving encryption is one of the tools used by the CryptDB which enables comparison operations to be specifically connected on encrypted information, without decrypting the operands. But encryption of non-numeric information isn't possible with this tool. Later the multi-cloud architecture was introduced which introduced the idea of partitioning the sensitive information and query logic into two different non-colluding clouds which don't have the knowledge about each other. However this architecture doesn't hold true for queries such as SUM/AVG.

8. REFERENCES

[1] R. A. Popa, C. Redfield, N. Zeldovich, and H. Balakrishnan, "CryptDB: Protecting confidentiality with encrypted query processing".
[2] C. Curino et al. (2011). Relational Cloud: A Database-as-a-Service for the Cloud.
[3] D. Boneh, D. Gupta, I. Mironov, and A. Sahai, "Hosting services on an untrusted cloud," in Advances in Cryptology.
[4] X. Chen, J. Li, J. Weng, J. Ma, and W. Lou, "Verifiable computation over large database with incremental updates" .
[5] X. Chen, J. Li, X. Huang, J. Ma, and W. Lou, "New publicly verifiable databases with efficient updates".
[6] S. Benabbas, R. Gennaro, and Y. Vahlis, "Verifiable delegation of computation over large datasets".
[7] W. Li, K. Xue, Y. Xue, and J. Hong, "TMACS: A robust and verifiable threshold multi-authority access control system in public cloud storage".
[8] H. Kadhem, T. Amagasa, and H. Kitagawa, "MV-OPES: Multi valued order preserving encryption scheme: A novel scheme for encrypting integer value to many different values".

