RESEARCHERID
THOMSON REUTERS

[ICRTCET-2018]

ISSN 2348 – 8034
Impact Factor- 5.070

GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

PRIVACY PROTECTION FOR SQL QUERIES WITH SECURE TWO CLOUD DATABASE SCHEME

Jisna M P*[1] & Manoj V. Thomas[2]

*[1&2]Dept. of Computer Science and Engineering Vimal Jyothi Engineering College Chemperi, Kannur

ABSTRACT

With the fast development of distributed computing clients are asked for to outsource their information, for greater flexibility and cost benefits. Privacy protection for such data is a challenging issue, so that user encrypt their data before outsourcing into the cloud which removes the problems of plain text keyword search attack. Many of the existing encryption schemes provide sufficient functionality to the SQL queries. But most of them mainly concentrates on particular type of SQL queries, not all and The majority of the strategies are powerless against protection leakage to cloud server. In this paper, we put forward a secure cloud database scheme which provides privacy protection to both numerical and string based queries. Numerical data is strongly protected since it is vulnerable to statistical properties an access pattern and string data is also protected using attribute based encryption method. Also it support operations such as SUM/AVG on encrypted data.

I. INTRODUCTION

Cloud computing is one of the growing fields in this era of computing that can provide profitable and flexible storage and processing capabilities for individuals, hospitals, industries etc. Data users are spurred to outsource their information to the server as opposed to paying much venture for programming or equipment to deal with the data. The advantages are financial and administrative benefits. However, there are major things that must be considered by anyone who migrates to cloud computing. One such challenge is privacy.

In data outsourcing, we store the sensitive data on a remote site that is not under the control of the client will make the hazard. Moreover, the information demand of the client can be uncovered to the semi-trust server to disregard the privacy of the user. The cloud may attend their best to steal the closet information for their benefits. What is more terrible is that the cloud could share such valuable information to the adversaries for revenue, which is an inadmissible chance[1]. Hence, secrecy of data and user privacy must be a serious concern.

To ensure secrecy of data, the data owner encrypts the database once transferred it to the server. We are also aware that the privacy of a user is portrayed by the query he requested and its outcome. It is vital to ensure the queries and its result from the unapproved parties as well as semi-honest cloud to protect the user privacy.

Some of the existing methods such as CryptDB[2] satisfy the properties such as confidentiality of data and hide the query pattern. To eliminate redundancy and ensure confidentiality, CryptDB utilizes onions of encryptions and also passive and active attacks between the client and DBMS are addressed. Some methods use set of cryptographic tools to ensure pri-vacy such as [3][11]. So that privacy issues cannot resolve thoroughly. Most of the problems with the private data can be resolved if the data in the cloud are stored and processed in encrypted form because any operations need to do will be executed on encrypted data. So that the cloud server deals with the encrypted data only. So such system has effective data protection.

The primary commitments of this paper are as follows: We extend the work done by Kaiping Xue.et al [1]. In this paper, authors presented a two cloud secure database scheme which provides privacy protection for numerical range queries as well as address the secrecy leakage in statistical properties. Here we would like to extend this work by considering the privacy protection for the numeric, string data and SQL queries. In addition, we support aggregate operations such as SUM/AVG.

II. RELATED WORK

We discusses various works done in security of out-sourced data. Most of the users mainly uploads databases and documents. So we are here classifies the survey into two categories. Here gives the survey of security methods for the query processing on encrypted data and keyword search over the encrypted documents.

A. Secure SQL query processing on encrypted data

There are some existing approaches that focus on providing security for query processing over cloud database. Authors in [4] propose a database scheme for processing the SQL queries over the encrypted database. This will helps to keep data secure and ensure privacy of user. In this paper, all the relational operators are implemented by doing the selection operator over the server-side database and thus consequently keeps the server from perceiving the type of query pattern . But the performance of the system will be affected if the no of requested values is more.

Baby et.al [5] propose a strategy for enhancing the execution of three kinds of inquiries, for example, non-aggregate, aggregate and user-defined queries over the encrypted data put away at the server. Non-aggregate queries(similarity queries and range queries) produces a characteristic index value which will be considered when user puts a query request whereas in the case of aggregate query operations such as sum/avg, comes about are acquired specifically from the encrypted attribute. For user-defined queries, an attribute is added to the record that shows how as often as possible the tuple is being queried. So for this type of queries, presence of the decrypted result will be identified by checking the frequency of tuple against the threshold. This method has good performance since it can execute multiple operations. But additional overhead in decrypting and storing on temporary database.

Samanthula et.al proposes [6] a secure system that support processing of queries on encrypted data and, while in the meantime ensuring the secrecy of information and protection of user's input query more. Here they use homomorphic encryption and garbled circuit technique to enhance the performance. Wong et.al [7]presents an encryption strategy that backings inter-operable operations, which permits wide range of SQL queries to be handled by the server on enciphered data. But the approach is constrained to integer values.

Authors in [9] provides a framework that provide data privacy by inserting null values in the database and secrecy views so that any queries about the views of db may not reveals significant information. Method proposed by Wong et.al [10] introduced the SCONEDB model, which is developed based on problems with KNN computations. In this scheme, it uses scalar product preserving encryption for defeat the attackers differently at different cost with the knowledge about the behavior of the user and the attacker on encrypted databases. Authors in [12] proposes a security plan that provide confidentiality of data, and secure query processing over cloud NoSQL databases.

B. Keyword search over encrypted data

Existing solutions such as [13]-[15] focus on improv-ing the searching over encrypted data by allowing multiple keyword search . Authors in [13] propose a scheme that support fuzzy matching based on algorithmic design. The algorithm is based on hashing and bloom filter mechanism and use euclidean distance to find out similarity of keywords. This method removes the need of dictionary and effectively supports multiple keyword fuzzy search. But the approach solve single letter mistakes and will not consider the most often occurring spelling mistakes and out of ordering. Zhangjie et.al [14] propose a scheme that handles common spelling mistakes. It consider keyword weight as a factor while check-ing similarity. Technique by [15] choose the better similarity measure based on greedy approach and TF/IDF model is used for index and query generation which provides confidentiality to index and queries. This is more efficient approach than since it solves its limitations. Authors in [18] and [16] proposes single keyword search over encrypted cloud data based on rank .

III. PRELIMINARIES

This section discussed about the background theories. In this work we use Paillier encryption and attribute based encrytion with AES. In this scheme we concentrates on the privacy of numeric and string related queries.

A. Paillier Cryptographic Algorithm

There are several fully homomorphic encryption techniques that support arbitrary computations. Paillier cryptosystem is one such scheme [19]which is partially homomorphic(addition homomorphic). i.e, in the event that two whole numbers i and j are encoded with a same key k will be indicated as $E_k(i)$ and $E_k(j)$, then the operation exists such that,

$$E_k(i) \qquad E_k(j) = E_k(i + j)$$

The algorithm goes through three stages: key generation, encryption and decryption.

A. Key generation:

Choose two large prime numbers k, l randomly and independently of each other and compute n=k*l. Compute μ

B. λ^{-1}mod n. The public key (PK) is n, and the private key (SK) is taken as (λ, μ).

- Encryption:

Let x be the plain-text. Firstly, select a number $r \in z_n{}^{*}{}_2$

randomly and cipher-text can be computed as, c= E (x; r) = (n + 1)$^x r^n$mod n^2 −→ 1

- Decryption:

The cipher-text can be decrypted as follows;

x= $\underline{c^{\lambda}\text{mod}} \qquad \underline{n^2{-}1}$ $\cdot\ \mu$mod n−→ 2

n

B. Attribute based encryption with AES

Attribute-based encryption is a type of public-key en-cryption in which the messages are encrypts/decrypts based on attributes. In this paper the fields and rows of a table is considered as attributes.

Each attribute is encrypted with any one of AES mode. There are for the most part five standard Modes of Operation: Electronic Code Book (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB), and Counter (CTR)[21]. Every mode is utilized in view of the coveted characteristics. But we are utilizing AES-CTR and AES-OFB mode in view of the measure of the trait.

IV. PROPOSED SYSTEM

In this section, first we gives an introduction about the overview of the system and then present the query intersection protocol to realize the processing of the numeric and string related query with privacy preservation on outsourced cloud database.

A. Overview of the proposed system

We extend the work done by Kaiping Xue et.al.[1] by providing aggregate operators such as SUM/AVG on encrypted data and support string-based query . Our proposed framework incorporates a database owner and two non-intriguing clouds. In our system, the database owner can be actualized as end user and the two cloud (allude to Cloud I and Cloud II) on the server-side, give the storage and the calculation benefit. The two cloud cooperate to react each numeric range query request from the customer. To lead a safe database scheme, information is enciphered as it is uploaded to be put away in cloud I, and the secret keys are put away in cloud II. For security purpose, these two clouds are thought to be non-overlapping with one another so that both of them knows only part of knowledge. But in the case of string related query, the cloud which stores the encrypted table is enough to process the query since we use consider SQL LIKE operator only, i.e, There is no need to decrypt the table.

We use paillier cryptosystem to encrypt the numeric related data and AES-CTR, AES-OFB mode encryption for string related data. Here we add the SUM/AVG operations on encrypted data since we use paillier cryptosystem which is addition homomorphic.

B. Privacy concerns

We additionally accept the clouds to be semi-fair: On one hand, both of the two cloud will gives the correct data in the interaction of our protocol(genuine); then again, the cloud may attempt to release the private data from the information that they procedure(inquisitive)[1]. So our work is based on the assumption that two clouds I and II are non overlapping in nature. We consider the issues related to data and statistical properties of query.

*Data: The privacy concerns of data contents include a column name and item values. So they are protected against an attacker by encrypting the column name and item values so that adversaries cannot easily get the original values and column names.

* Statistical properties: Repeated query request may leaks statistical properties[1] and access patterns. So we are focus on preventing such attacks by hiding the query patterns and preventing the repeated query request by using token based scheme.

C. Design

Architecture of the proposed system are shown below. In this scheme, client and cloud service providers are two participants. Client needs to upload their database to cloud server which contains confidential information. So in light of this plan as appeared in fig 1, for guaranteeing security, information are scrambled and put away in one cloud(cloud I). Here numeric data are encrypted using Paillier and string data are encrypted using AES and key for numeric data are stored in other cloud (cloud II). Key for AES are known to client only. Each cloud knows only partial information, so that no one can obtain any private information.

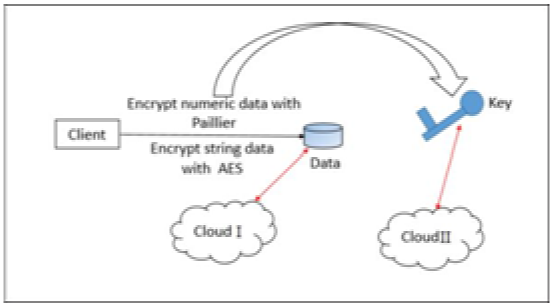
Encrypt numeric data with Paillier
Key
Client
Encrypt string data with AES
Data
Cloud I
CloudII

Fig. 1. Architecture for the proposed scheme

ISSN 2348 – 8034
Impact Factor- 5.070

The query logic is shown in fig 2.

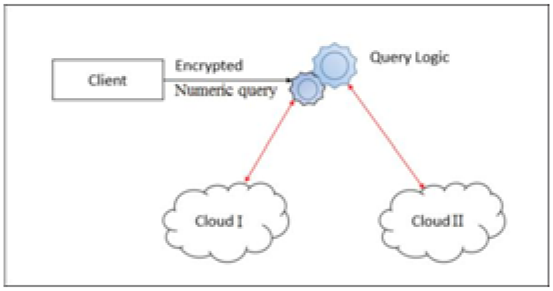

Fig. 2. Query Logic

Two cloud cooperates together to process the queries whereas in the case where string appeared in WHERE clause of a query, cloud I itself is enough to process the query. ie, there is no need to decrypt the data, because in this case, we consider only LIKE operator so that it can find out the matching result from encrypted data which will be depicted in fig 3. Also, our proposed scheme compute SUM/AVG on encrypted data by using the properties of Paillier.

Proposed system is made out of Table Creation and Query Protocol. The method of Query interaction Protocol comprises of four sections: Query Request, Item Send, Index Send, and Query Response which is clarified in fig 4.

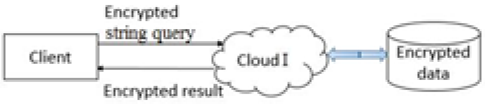

Fig. 3. String related Query

Client want to outsource the database to cloud. He/she first create the table and encrypted table is then sent to cloud I and secret key for numeric data is shared with cloud II. For retrieving the data, client needs to generates a sql query. This query request is forward to cloud I by the query protocol. Cloud then find out the columns corresponds to the request, but cannot identify the items that matches query pattern since

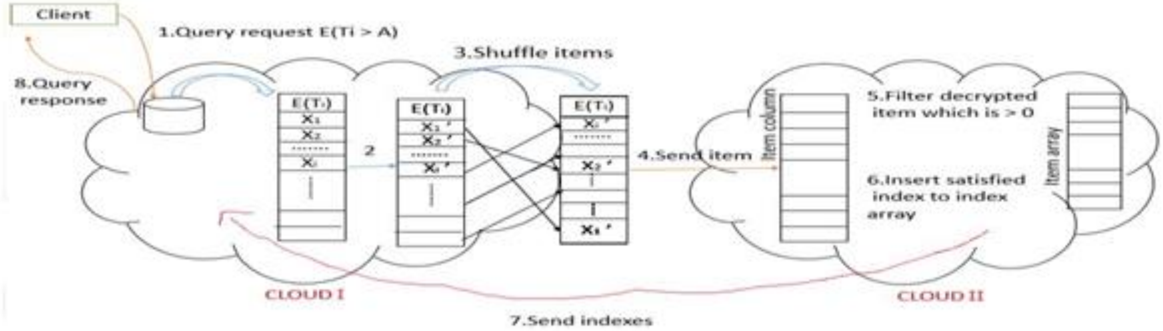

Fig. 4. Query protocols(Each step is marked 1,2,3..

cloud I does not knows the decryption key. To maintain the privacy concerns, cloud A converts each items into another form based on Paiilier encryption and shuffles it so that cloud cannot identify the exact operator in the query request. After receiving data items and query from cloud I, Cloud II decrypt the items with secret key and identifies the matching items and responds with the corresponding index value. On receiving index, cloud I find out exact row items and responds with the result to the client. Two cloud together cooperates each other to respond with the query request. This query intersection protocol will be explained in detail in subsection D.

D. Query Intersection protocol

Numeric related query is processed as per [1].

1.Table creation:
Client needs to generate the table and encrypt it before outsourcing it on cloud
1) The client encrypt each column of the table T_i with an arbitrarily chose symmetric key k. Enciphered column is meant as E(T_i) and symmetric key is safely kept by client.
2) For every numeric related item x, client generates a public-private key pair(PK/SK) for Paillier cryptosystem. Encryption is carried out in this way;
X=E(x,PK)
3) For string related item z, encryption is carried out by using AES encryption. Encrypted table is transfered to cloud I along with public key and secret key is shared to cloud II.

2.Query Request:
(i) String related Query: The case where string appears in WHERE CLAUSE, Client generate the query as "SELECT * FROM table WHERE name LIKE a%". This plain text query is converted to an encrypted format. Here there is no need to decrypt the data, it can be find out the matching item from encrypted text since we consider LIKE operator only(fig 4). (ii) Numeric Query: Client generate the query as "SELECT *

FROM table WHERE T_l <a". This plain text query is converted to an encrypted format.
1) Encrypt the column name and range boundary value:Encrypt the column name with symmetric key k and limit esteem With public key PK.
2) Generate the token
3) Send the Query Request:Encrypted query is sends to cloud A with signed token.

3.Item Send:
Cloud I first identifies the column E(T_i). It goes through this following phases previously sending the things to cloud II:
1) Number comparison:For each item $X_j = T_{ij}$ in the column, Cloud I selects a positive integer r_j randomly and j individually, where $0 \leq j \leq rj$ and computes,
$x_j = \frac{x_j}{A} \; {}^{r_j} \cdot E(- {}_j \cdot PK) \longrightarrow 5$
Items shuffling: Cloud I shuffles the items in column L to produce the new column L and securely stores the mapping.

4.Index send:
On receiving the items, cloud II firstly authenticate the got token by look up its expiry time. Then cloud II decrypts each item and if $x_J > 0$, index j is inserted into new index array L . Along with this index cloud II add some dummy indexes for maintaining privacy.

5.Query response:
Cloud I find out the matching index from the original column for each each index j in the index column L based on the index mapping information table. As per the mapped index j

Cloud I sends the relating columns in the table, as the query reaction, to the client.
Subsequent to accepting the response, the customer can deci-pher the items with SK to acquire the required information, and ousts sham things that does not fulfill the question predicate.

E. Encryption for string related data

The database is secured by scrambling each field (con-taining string information) of a database table. Here the field and a row of a database table is allude as attribute and record respectively. Each attribute is encoded with AES in either of two modes based on the length. counter mode (AES-CTR) is used for attribute having 128 bit length or less and output feedback mode (AES-OFB) is used for other cases[20]

Here we follows the attribute based encryption by Chen[20]. The encryption is carried out as follows. Two modes of AES worked based on the length of the attribute. AES-CTR works well with attribute of 128 bit length or less. The CTR functionencrypts the l-bit long attribute using the 128-bit attribute seed and database key. Seed can be formed based on the structure of a logical schema. Each attribute can be uniquely identified since we are using the parameters such as (databaseID; tableID; rowID; columnID) as identifier which is spatially unique across different databases and tables and also use a global incremental counter to each record to provide temporal uniqueness. Encryption is finished by XOR-ing the first l msb of encryption pad with the data. AES-OFB function is used when the attribute longer than 128 bit. Here the function uses a series of 128 bit encryption pad for encryption. The most significant l bits of the connected encryption pad $p^0p^1......p^m$ is then used to encrypt the attribute

data. Decryption is done by XOR-ing the cipher text with the same encryption pad. AES-CTR and AES-OFB both use AES encryption but in diverse operation modes.

F. Aggregate Query operation

Here we use Paillier cryptosystem for numeric data encryption. Since Paillier is addition homomorphic. So we can execute the aggregate query on the database without decrypting the data. By using the property of Paillier, we can easily process the aggregate queries such as SUM/AVG. In the case of SUM query, summation can be done on encrypted data and this outcome will be decrypted to give the real sum which is given to the client. This result is same as when finding out sum after decrypting each data. Hence, this will reduce the computation cost by minimizing the cost for decryption of data on the client side. The SUM can be calculated as follow: C(p) . C(q) = p + q

where C(p) C(q) denotes the cipher-text of corresponding plain-text values p and q. Here the result is same if we find out the sum after decrypting the cipher-text because of the Paillier property .

V. RESULT ANALYSIS

To validate the proposed approach, we will first analyze the performance status of our system against the existing system [1], and then evaluate the execution time for SUM/AVG query processing. To verify the performance benefits, compare proposed system with existing schemes based on the properties they provides.

A. Experimental Setup

The recommended system is implemented using JAVA 1.8 and Netbeans 8 in a core i3 processor and 4-GB memory. Here cloud environment is established by AWS(Amazon Web Services).

B. Evaluation of result

Table I comparison result

Number of records	Existing scheme	Proposed scheme
5	2042	2200
10	4747	5000

15	7262	7524
20	8835	8934
25	10157	11000
30	18520	19100

Here we compare our scheme with the existing scheme[1] against the processing time. Figure 5 proves the comparison result with respect to the processing time. Our scheme has slight improvement with the existing scheme, but it also considers non numeric data(string) also.Compared with existing, this scheme has additional benefits such as it supports aggregate queries such as SUM/AVG. So that this slight variation for range queries is a good achievement.

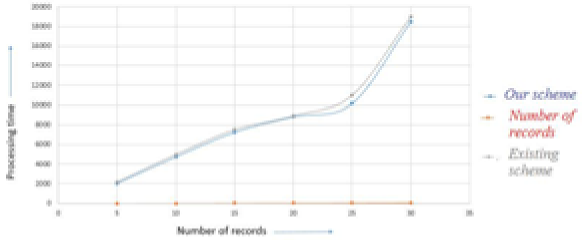

Fig. 5. Comaprison result of numeric range query

In figure 6, we can see the execution time required for aggregate operation(SUM/AVG). Here as no of records in the database increases, the execution time also increases slightly nor exponentially or linearly. As we can see in figure, it takes 316 milliseconds. That means according to our measurements, it takes only 316 sec for over 10,000 records which is good.

Table 2 shows different schemes and the attribute they are supporting. From this result we can understand that our scheme is better than other approaches. Here proposed scheme prevent statistical attacks, support both numeric and non numeric data and also perform aggregate and range queries over encrypted data. Even though it does not support all oper-ations, considering 3 attributes, our scheme has performance advantage.

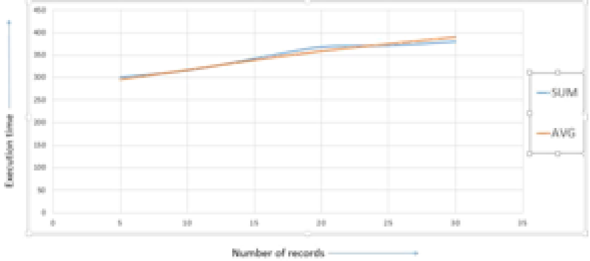

Fig. 6. Processing time for SUM/AVG

Table 2: Comparison of different schemes

Attribute Scheme	Privacy of data	Privacy of query & acess patterns	Aggregate query	Range query	Numeric data	String
Scheme 1 [4]	Yes	No	Yes	Yes	Yes	Yes
Scheme 2 [22]	Yes	No	Yes	Yes	Yes	No
Scheme 3 [5]	Yes	No	Yes	Yes	Yes	Yes
Scheme 4 [1]	Yes	Yes	No	Yes	Yes	No
Our scheme	Yes	Yes	Yes	Yes	Yes	Yes

VI. CONCLUSION AND FUTURE WORK

In this paper, we analyze the existing solutions for secure query processing in cloud database. Most of the works are securely processing queries but cannot address the security leakage by untrusted CSP, i.e, data will be available to unau-thorized users. We presented a two-cloud architecture with series of query protocols for the database which are outsourced to cloud with privacy protection. This protocol ensures the security for both numeric and string data, statistical properties, numeric range queries and string related queries. Here we use paillier encryption to protect numeric data and range queries as in [1], but we also provide operations such as SUM/AVG over encrypted data by using the addition homomorphic property of Paillier. For the processing of string related query, we consider only the LIKE operator. AES encryption is used here. Analysis of the results shows that our our scheme can meet the security conservation prerequisites.

In future, we will extend this work by enhancing the security for all complex sql queries and support all operations on encrypted data.

REFERENCES

1. *Xue, Kaiping, Shaohua Li, Jianan Hong, Yingjie Xue, Nenghai Yu, and Peilin Hong. ”Two-Cloud Secure Database for Numeric-Related SQL Range Queries With Privacy Preserving.”IEEE Transactions on Information Forensics and Security, 12, no. 7 (2017): 1596-1608.*
2. *Popa, Raluca Ada, Catherine Redfield, Nickolai Zeldovich, and Hari Balakrishnan. ”CryptDB: protecting confidentiality with encrypted query processing.” In Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles , pp. 85-100. ACM, 2011.*
3. *Popa, Raluca Ada, Frank H. Li, and Nickolai Zeldovich. ”An ideal-security protocol for order-preserving encoding.” In Security and Privacy (SP), 2013 IEEE Symposium on, pp. 463-477. IEEE, 2013.*
4. *Hue, T. B. P., D. N. Thuc, T. B. D. Thuy, Isao Echizen, and Sven Wohlgemuth. ”A User Privacy Protection Technique for Executing SQL over Encrypted Data in Database Outsourcing Service.” In Conference on e-Business, e-Services and e-Society, pp. 25-37. Springer, Berlin, Heidelberg, 2013.*
5. *Baby, Tinu, and Aswani Kumar Cherukuri. ”On query execution over encrypted data.” Security and Communication Networks 8, no. 2 (2015): 321-331.*

6. Samanthula, Bharath Kumar, Wei Jiang, and Elisa Bertino. ”Privacy-preserving complex query evaluation over semantically secure encrypted data.” In European Symposium on Research in Computer Security, pp. 400-418. Springer, Cham, 2014.
7. Wong, Wai Kit, Ben Kao, David Wai Lok Cheung, Rongbin Li, and Siu Ming Yiu. ”Secure query processing with data interoperability in a cloud database environment.” In Proceedings of the 2014 ACM SIGMOD international conference on Management of data, pp. 1395-1406. ACM, 2014.
8. Agrawal, Rakesh, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu. ”Order preserving encryption for numeric data.” In Proceedings of the 2004 ACM SIGMOD international conference on Management of data, pp. 563-574. ACM, 2004.
9. Bertossi, Leopoldo, and Lechen Li. ”Achieving data privacy through secrecy views and null-based virtual updates.” IEEE Transactions on Knowledge and Data Engineering 25, no. 5 (2013): 987-1000.
10. Wong, Wai Kit, David Wai-lok Cheung, Ben Kao, and Nikos Mamoulis. ”Secure knn computation on encrypted databases.” In Proceedings of the 2009 ACM SIGMOD International Conference on Management of data, pp. 139-152. ACM, 2009.
11. Liu, Zheli, Xiaofeng Chen, Jun Yang, Chunfu Jia, and Ilsun You. ”New order preserving encryption model for outsourced databases in cloud environments.” Journal of Network and Computer Applications 59 (2016): 198-207.
12. Ahmadian, Mohammad. ”Secure query processing in cloud NoSQL.” In Consumer Electronics (ICCE), 2017 IEEE International Conference on, pp. 90-93. IEEE, 2017.
13. Wang, Bing, Shucheng Yu, Wenjing Lou, and Y. Thomas Hou. ”Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud.” In INFOCOM, 2014 Proceedings IEEE, pp. 2112-2120. IEEE, 2014.
14. Fu, Zhangjie, Xinle Wu, Chaowen Guan, Xingming Sun, and Kui Ren. ”Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement.” IEEE Transactions on Information Forensics and Security 11, no. 12 (2016): 2706-2716.
15. Xia, Zhihua, Xinhui Wang, Xingming Sun, and Qian Wang. ”A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data.” IEEE transactions on parallel and distributed systems 27, no. 2 (2016): 340-352.
16. Xu, Peng, Hai Jin, Qianhong Wu, and Wei Wang. ”Public-key encryption with fuzzy keyword search: A provably secure scheme under keyword guessing attack.” IEEE Transactions on computers 62, no. 11 (2013): 2266-2277.
17. Hao, Feng, John Daugman, and Piotr Zielinski. ”A fast search algorithm for a large fuzzy database.” IEEE Transactions on Information Forensics and Security 3, no. 2 (2008): 203-212.
18. Wang, Cong, Ning Cao, Jin Li, Kui Ren, and Wenjing Lou. ”Secure ranked keyword search over encrypted cloud data.” In Distributed Com-puting Systems (ICDCS), 2010 IEEE 30th International Conference on, pp. 253-262. IEEE, 2010.
19. Paillier, Pascal. ”Public-key cryptosystems based on composite degree residuosity classes.” In International Conference on the Theory and Applications of Cryptographic Techniques, pp. 223-238. Springer, Berlin, Heidelberg, 1999.
20. Chen, Bony HK, Paul YS Cheung, Peter YK Cheung, and Yu-Kwong Kwok. ”Cypherdb: A novel architecture for outsourcing secure database processing.” IEEE Transactions on Cloud Computing (2015)
21. https://en.wikipedia.org
22. Gahi, Youssef and Guennoun, Mouhcine and El-Khatib, Khalil ”A secure database system using homomorphic encryption schemes” arXiv preprint arXiv:1512.03498 (2015).