Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 10, October-2019

Cloud Computing and Secure Query Processing

Ameya Tathavadkar
Department of Information Technology
K.J. Somaiya College of Engineering,
Vidyavihar Mumbai, India

***Abstract*:- Cloud computing [3] enables utilization of great amount of "computing resources"[3] to process client's data efficiently and allows clients to store large amount of data. In order to make use of such services, in many cases information is subcontracted to Cloud Service Providers (CSPs). Although, subcontracting important information to the CSPs [3] could give rise to a catastrophe. So, any kind of data has to be kept safe by using encryption techniques. Basic encryption methods are unproductive because such methods require decryption process [3]. In order to tackle this problem, a "fully homomorphic encryption (FHE)" [3] scheme is needed because it allows processing of data in an encrypted form. The main issue with regards to different "FHE" [1] schemes suggested is that of efficiency. "Privacy Homomorphism (PH)" [1] technique can also be used to provide great security features to client.**

Keywords— Privacy, Homomorphic encryption, Security, Cloud Computing, Encrypted data, Decryption data.

INTRODUCTION

One of the upcoming technique to remotely access, store and process data is the use of cloud technology. Cloud Services Providers (CSPs) main purpose is to satisfy clients' needs like huge storage spaces along with powerful computing resources for storing and processing data efficiently. To completely make use of the facilities provided by the CSPs, information must be subcontracted to them. But it is possible that subcontracting such information to a third party like a CSP could result in leakage of information or illicit use of data. So, it is necessary to protect data with the help of encryption techniques.

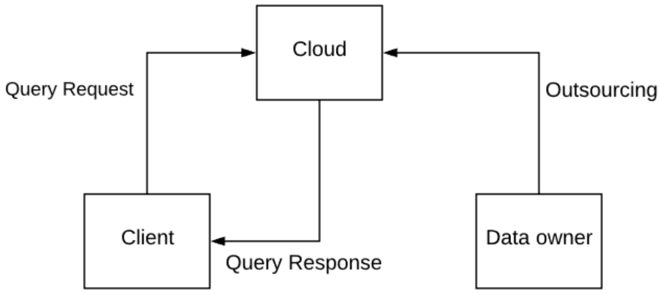

Figure 1 Query Processing Model for Cloud

Since the data is in encrypted form before applying any processing the CSP needs to decrypt the data by using the correct decrypting method. But such an approach is not recommended since decrypting our data on cloud may disclose some sensitive information like personal information or social security numbers etc. to the CSPs. It could also reveal the information to other cloud clients because it is possible that vulnerabilities may exist in the environment. The solution to this problem is to use homomorphic encryption [1] schemes as they allow us to process data even without decrypting it.

I. FEATURES OF CLOUD COMPUTING

Cloud Computing revolutionized the manner in which tech giants stored their data .We take a look at some of the characteristics of cloud computing-

1. **Elasticity:** It allowed the infrastructure to scale dynamically on demand.
2. **Adaptability:** Cloud [4] offered a sense of automatization which will allow it to manage itself. Human interference required to manage it should be minimum.
3. **High Availability:** Working on duplicated information in various data sites, the Cloud [4] had to be reliable and "not sensitive to the failure of an instance or a data center" [4].
4. **Cost Cutting:** Payment per usage implies that the payment is made on the basis of the services utilized by the user.

II. SECURITY ISSUES FOR CLOUD COMPUTING

Outsourcing data to CSPs always comes with a certain risk. When we allow CSPs to process our data it also implies that they become responsible for the safety and security of our information. This is one of the major causes the making the security specialists anxious. Factors such as reductions in costs, quick re-provisioning of resources and easy maintenance are some of the advantages of cloud computing. Let us have a look at a few of the issues that can arise-

1. **Integrity of data-**
   This is one of the crucial factors for all storage institutes. Usually, this implies protecting data from unlawful access and changes. Because of the huge number of people and possible access sockets in a cloud database, permission is pivotal in making sure that only approved users can interact with data. Methods such as "RAID-like strategies" [4] and "digital signature" [4] can be used in order to obtain data integrity.
2. **Data Confidentiality-**
   In order for users to allow access to their sensitive data confidentiality is necessary. It is not possible to eradicate the internal threat so the clients are hesitant to trust CSP's, and thus find it very risky to store their important data on cloud.
3. **Data Availability**
   Data availability means: when mishaps such as "hard disk damage" [4] and "network failures" [4] occur, the extent up to which the customer's information can be retrieved and how the customers

can cross-check their information by different methods instead of relying on the assurance by the CSP only.

4. **Data privacy**

"Privacy is the right of an individual or group to hide themselves or information about them and thereby reveal it selectively" [4]. The main issues faced in the field of data privacy are-

- How to ensure that user is in complete control of their information and that there are no illicit activities regarding the sensitive information
- How to make sure that replicated data is consistent at all sites, where replicating data at different sites is the preferred.
- Whose responsibility is it to ensure that all the legal constraints are in place for sensitive data?
- Exactly till which degree are the CSPs involved regarding the processing of data which can be correctly recognized, determined and checked.

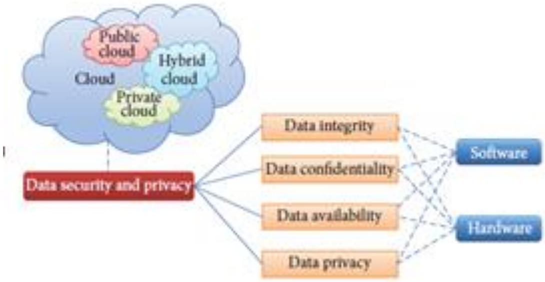

Figure 2 Data Security and Privacy

The possibility or occurrence of such risks is reduced if we allow the CSP to perform processing of the data without having to decrypt it which can be done by using the cryptosystems based on "Homomorphic Encryption" [4].

III. RELATED WORK

In a common model, non-trusted outsourcing servers are used in order to store and manage any type of data and then data owners grant access to trusted users to query the data. The first approach tries to reduce the exposure of precise information. For this the most accepted approach in recent research papers is encryption. Various SQL operations like MIN, MAX, and COUNT and so on can be rewritten to work on encrypted data by using "order preserving encryption scheme (OPES)" [2] which allows indexes to be built directly on the cipher text. But OPES doesn't allow operations like AVG and SUM to be performed. For these operations to work we need to decrypt the data first. In order to provide anonymity, confidentiality and privacy during private information retrieval (PIR) query anonymization is used to blend the users query with noise. One of the proposed solution to protect data and user privacy given in [5] is based on secure traversal framework and "privacy homomorphism" [2] based encryption scheme and "secure protocols for processing k-nearest-neighbor queries (kNN) on R-tree index."[2]

IV. APPLICATION OF HOMOMORPHIC ENCRYPTION IN CLOUD COMPUTING

"Homomorphic Encryption systems are systems which can perform operations on encrypted data without decryption. An encryption scheme is called homomorphic over an operation if it supports the following equation-

$E\ (m_1)*\ E\ (m_2) = E\ (m_1*m_2),\ \forall m_1,\ m_2 \in M$

Where E is the encryption algorithm and M is the set of all possible messages"[1] .

When we decrypt the outcome of any operation, it matches with the result of the operation on the raw information.

Consider the following example where-

-"E_k is an encryption algorithm with key k.

- D_k is a decryption algorithm.

-n, m" [1] are the two numbers to be added/multiplied

Then-

1) "$D_k\ (E_k\ (n) \times E_k\ (m)) = n \times m$

2) $D_k\ (E_k\ (n) + E_k\ (m)) = n+m$" [1]

The first property is called multiplicative "homomorphic encryption" [1] while the latter is additive "homomorphic encryption" [1]. When both cases are fulfilled concurrently only then an algorithm is fully homomorphic.

We shall now see different scenarios where homomorphic encryption is used-

Consider a case of performing a search and retrieve operation on encrypted data. Consider a system of data management of a hosting data service [4], shown below which consists of different entities like "data owner", "data user" and a "storage server" [4].

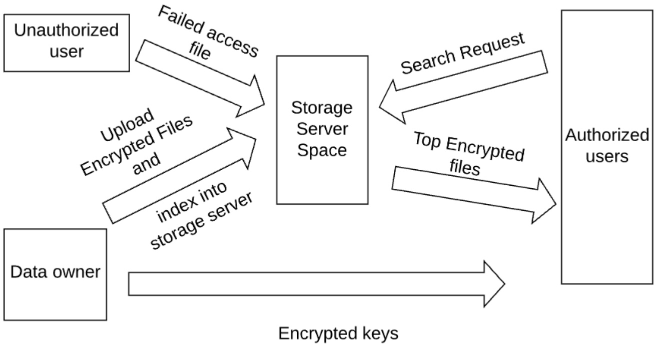

Figure 3 Search and retrieval process

The client who owns the data has a large number of data files. For greater flexibility such holders are advised to subcontract the information from local scope to global scope. To protect data files, they are encrypted before uploading them globally. So allowing search and retrieval [4] on any type encoded information is of supreme importance. Let's say that "the data owner has a collection of n files say, $C = \{f_1, f_2...f_n\}$ which may be of extension .txt, .doc and .pdf" [2]. In order to protect the information from unlawful users we can use the following-.

1) **Secure Privacy Homomorphism (PH)**

PH is a type of encryption method "which maps a set of operations on plain text to another set of operations on cipher text" [2].

1. Process of encryption

It is the process of changing "plain text into a cipher text" [2] with the use of "public and private keys" [2]. Given A is a clear text which uses private keys to convert A into cipher text. The query is sent to the cloud by the client and then owner sends encrypted key index to client.

A=query

"E (i) = encrypted index key" [2]

Steps-

```
1) "Start.
2) Take any number & multiply by 13 and store that answer.
3) Convert that answer into data type String and store into string variable fs.
4) Initialize integer array ak [ ] of size 10
5) Initialize index counter variable to zero.
int ak_ind=0;
6) For int i=0 to i<fs.length
Integer k =
Integer.parseInt (fs.valueOf (fs.charAt (i))) +1;
If ((k==10) and (ak_ind>0))" [2]
        ak [ak_ind-1] =ak [ak_ind-1] +1;
        ak [ak_ind] =0;
        Increment ak_ind by 1;
Else
        ak [ak_ind]=k;
        Increment ak_ind by 1;
"End IF
End For
7) For int j=0 to j<ak_ind
enc_val=enc_val.concat (ak[j].toString ());
End For
8) Return String s1.
9) Stop" [2]
```

*2. **Process of decryption***

This is the process in which cipher text is converted into normal message by using public as well as private keys. In fig 3, data owner sends the decryption index key E^{-1} (i) to the data cloud for future distance decryption.

"E^{-1} (i) = decrypted index key.

Steps-

```
1) Start.
2) Take String s1.
3) Initialize int prime=13.
4) Convert String s1 into integer and store it in integer variable dc.
5) Initialize int dec=0;
6) dec=dc/prime.
7) Stop."[2]
```

1) Fully Homomorphic Encryption:

For any type of processing or calculation on data which is saved on the cloud, it is advised to opt for the "fully Homomorphic encryption" [5] which can carry out any procedure on encoded information without decoding. The use of fully Homomorphic encryption [5] is a crucial part of Cloud Computing security. It also allows us to subcontract the processing on sensitive information to the Cloud, as we need not worry about privacy breech due to decryption.

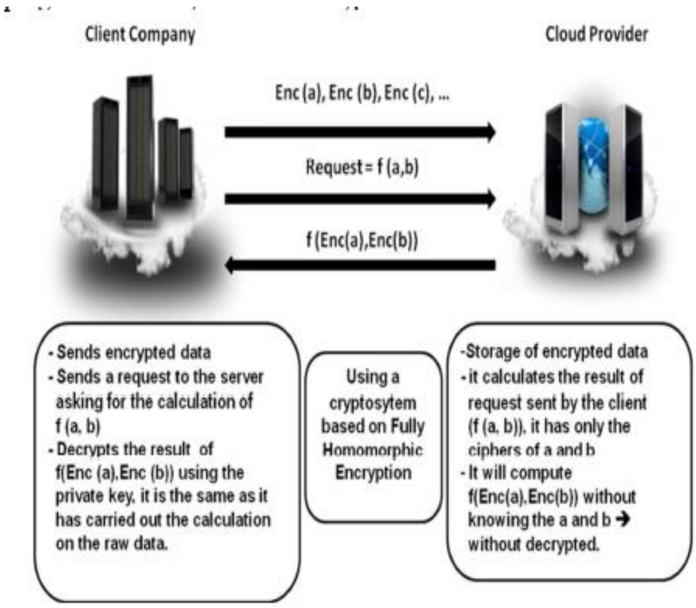
Client Company
Cloud Provider
Enc (a), Enc (b), Enc (c), ...
Request= f (a,b)
f (Enc(a),Enc(b))
- Sends encrypted data
- Sends a request to the server asking for the calculation of f (a, b)
- Decrypts the result of f(Enc (a),Enc (b)) using the private key, it is the same as it has carried out the calculation on the raw data.
Using a cryptosytem based on Fully Homomorphic Encryption
-Storage of encrypted data
- it calculates the result of request sent by the client (f (a, b)), it has only the ciphers of a and b
- It will compute f(Enc(a),Enc(b)) without knowing the a and b → without decrypted.

Figure 4 Process of FHE with respect to Cloud Computing

2) El Gamal Cryptosystem:

The El Gamal Cryptosystem is as follows-
Consider a prime number p and select α in such a way that α is a generator of Z_p*.Pick b and β where $\beta=\alpha^b$(mod p) and p, α, β are public while b is private. Let r ∈ Z_{p-1} be a secret number. Then,
$E_k(x, r) = (\alpha^r \bmod p, x\beta^r \text{modp})$

El Gamal Cryptosystem uses the "multiplicative homomorphic encryption propriety:

Let x1 and x2 be plaintexts. Then,
ek(x1, r1) ek(x2, r2)
$= (\alpha^{r1} \bmod p, x1\ \beta^{r1} \bmod p)\ (\alpha^{r2} \bmod p, x2\ \beta^{r2} \bmod p)$
$= (\alpha^{r1+r2} \bmod p, (x1\ x2)\ \beta^{r1+\ r2} \bmod p)$
$= e_k(x1+x2, r1+r2)$" [1]

After putting the plain text in the exponent we get-

"$ek(x, r) = (\alpha^r \bmod p, \alpha x\beta^r \bmod p)$
Then the homomorphism is additive:

$ek\ (x1, r1)\ ek(x2, r2) = (\alpha^{r1} \bmod p, \alpha^{x1}$
$\beta^{r1} \bmod p)\ (\alpha^{r2} \bmod p, \alpha^{x2}\ \beta^{r2} \bmod p)$
$= (\alpha^{r1+r2} \bmod p, \alpha^{x1+\ x2}\ \beta^{r1+\ r2}) \bmod p$
$= ek(x1+x2, r2+r2)$" [1]

V. CONCLUSION:

The concept of cloud computing[2] established on different homomorphic techniques is relatively new but enables us to carry out processing of data in the encrypted format .Thus confidentiality of data of the client is maintained. The main issues which hinder the progress of this field are "security and privacy issues" [4]. Decreasing information storage and processing price is a mandatory condition for all companies, while study of information is the most significant task in any of the organizations. This paper studies different techniques like El Gamal Cryptosystem, Fully Homomorphic encryptions, Secure Privacy Encryptions.

VI. REFERENCES

[1] Secure Cloud Computing Through Homomorphic Encryption Maha Tebaa,Said El Hajii

[2] Attribute Based Secure Query Processing in Cloud with Privacy Homomorphism Ms. Rupali S.Khachanel and Dr. Pradeep K.Deshmukh

[3] Secure Remote Data Processing in Cloud Computing Mohd Rizuan Baharon, Qi Shi, David Llewellyn-Jones, and Madjid Merabti.

[4] Sun, Yunchuan & Zhang, Junsheng & Xiong, Yongping & Zhu, Guangyu. (2014). Data Security and Privacy in Cloud Computing. International Journal of Distributed Sensor Networks. 2014. 1-9. 10.1155/2014/190903.

[5] HakanHacgm, BalaIyer, and SharadMehrotra.Efficient execution of aggregation queries over encrypted relational databases. In YoonJoon Lee, Jianzhong Li, Kyu-Young Whang, and Doheon Lee, editors, Database Systems forAdvanced Applications, volume 2973 of Lecture Notesin Computer Science, pages 125– 136. Springer Berlin Heidelberg, 2004.