International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 07 | July-2016 www.irjet.net p-ISSN: 2395-0072

# Security Technique for Enhancing Secrecy of Outsourced Data in Cloud

**Tina Sara Kenu[1], Mr. Arun R[2]**

*[1]MTech Cyber Security, Dept. of CSE, SNGCE, Kadayiruppu, Kerala, India.*
*[2]Asst. Prof. MTech Cyber Security, Dept. of CSE, SNGCE, Kadayiruppu, Kerala, India.*

---

**Abstract -** *Database outsourcing is a recent and important manifestation of this trend. The notion of database outsourcing enables the data owner to delegate the database management to a cloud service provider (CSP) that provides various database services to different users. Recently, plenty of research work has been done on the primitive of outsourced database. However, it seems that no existing solutions can perfectly secure a generic database and support the property of correctness of the results, especially in the case when the dishonest CSP intentionally returns an empty result to the user. Here presents an effective encrypted outsourced generic database which ensures the confidentiality of the sensitive data and the correctness of the search result even if the dishonest CSP returns empty result. For making the sensitive data more secure that is stored in the outsourced generic database, a framework is introduced which implement the concept of negative database on generic database(EAV model)for enhancing security. It consists of various set of algorithms which manipulate input data and store it in the database with some counterfeit data. Negative database provides an extra layer of security. In addition, for checking the correctness of the search result a mentioned above, a new verifiable auditing scheme for this outsourced database is introduced together with this. Thus the advantages of negative database approach on generic database and the verifiability auditing scheme is demonstrated and emphasized.*

***Keywords***: Database outsourcing, Negative Database, Generic Database, Database Security, Auditing, Database encryption**.**

## 1. INTRODUCTION

The continued growth of the internet and the other technologies has gained a tremendous trend towards outsourcing data management and information technology needs to other providers. By outsourcing, organizations can concentrate on their core tasks and operate other business applications via the Internet, rather than incurring substantial hardware, software and personnel costs involved in maintain applications.

Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in third-party data centers [1].It enables convenient and on-demand network access to a centralized pool of configurable computing resources and has a plenty of benefits for real-world applications such as on-demand self service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing, outsourcing, etc. In the outsourcing computation paradigm, the resource constraint clients can outsource the expensive computing and storage into the cloud service provider (CSP) and enjoy unlimited computing and storage services in a pay-per-use manner [2].

The database outsourcing paradigm poses numerous research challenges which influence the overall performance, usability, and scalability. One of the foremost challenges is the security of the stored data. A client stores its data at an external, potentially not trusted, database service provider. It is essential to provide adequate security measures to protect the stored data from both malicious outsider attacks and the database service provider itself. That is why database security should be ensured for the data that is stored in the outsourced database. Databases in general usually store sensitive data, such as medical records, credit card numbers and government top secret information's. For this reason, the database administrators should ensure proper protection of sensitive data. To achieve this, an unambiguous and consistent security policy must be followed, employing a number of mechanisms in order to create different layers of security. To be short, database security can be described by the following properties [3]:

- Confidentiality: Information should not be accessible to unauthorized users.
- Integrity: Data cannot be corrupted; only authorized users should be able to modify the data
- Availability: Authorized users should be able to access the data reliably at all times

The last line of defense for the database management is data encryption. Even if a malicious user manages to circumvent all the above mechanism he/she will need to decrypt the data. Data encryption can provide very strong security for data at rest but there are many factors to be considered before encrypting the database.

- Should the data be encrypted by the DBMS or by the application that created them?
- Who can decrypt the data?
- Do all data need to be encrypted

All of the above points depict serious decisions one should consider during the design phase of a database and the applications using it.

Hucigeumeus et al [3] first implicitly introduced the notion of database outsourcing. In outsourced database

(ODB) model, the data owner to reduce the heavy database maintenance cost. In addition, the data owner performs the database encryption operations and uploads the encrypted database with corresponding indices to the CSP. The CSP is responsible for providing all necessary resources and service to users. The users can issue various g=query requests to the CSP and receive the corresponding results from CSP.

When outsourcing the database to the cloud the following security challenges should be taken into account. First, the secrecy of outsourced data: the outsourced data contains some sensitive information that should not be exposed to the semi-trusted CSP. The CSP should not learn anything about what it is actually stored. The traditional encryption technique can provide a solution to this problem since it is very difficult to perform effective operations over the encrypted data. Second, the security challenge is the verifiability of results. The dishonest CSP may return an incorrect result in order to save the computation resources. The user can efficiently perform verifiable auditing for the result by the CSP [2].

## 2. BACKGROUND

We first give formal introductions and information's regarding database outsourcing and the security of the outsourced data. Now we will see background information regarding the proposed system.

### 2.1 Generic Database

Generic data models are generalization of conventional data models that we normally use in application databases. Generic database is based on basic EAV (Entity Attribute Value) model where all types of data can be stored in single table without worrying about which type of data can be stored and where it has to be stored. It is more general database that can be used for any purpose. Its application is widely seen in the health care systems where the structure of the table is not strictly defined. As to maintain flexibility of database we prefer to use a generic model as it reduces number of entities in a database, and thus preventing unstable situations like when we need to change the required fields in any existing database.

### 2.2 EAV(Entity Attribute Value) model

The EAV data model, consist of three parts: entities, attributes and values. In an EAV model, each attribute-value pair is a fact describing an entity, and a row in an EAV table stores a single fact.EAV tables are often described as "long and skinny". "long" refers to the number of rows, "skinny" to the few columns. The EAV models simplest form is three column tables; the entity column which describes the data item, for example a product. The attribute column that describes attributes for entities, and the value table contains the value for those attributes. EAV is also known as object-attribute-value model. The idea of EAV model originates from the concept of association lists as they are called LISP [4].Fig-1 Shows the conversation of a simple table to EAV model table.

| Simple Table | | |
|---|---|---|
| ID | Name | Disease |
| 1 | Renna | Malaria |
| 2 | Rohan | Sugar |

| EAV MODEL | | |
|---|---|---|
| Entity | Attribute | Value |
| 1 | Name | Reena |
| 1 | Disease | Malaria |
| 2 | Name | Rohan |
| 2 | Disease | Sugar |

Fig-1: EAV Mode

### 2.3 Negative Database

In database security, a negative database is database that saves attributes that cannot be associated with a certain entry. In a normal (positive) database, a scrutiny of a single record provides the meaningful information. Whereas, a similar approach for a negative database reveals very little meaningful information about the original data. A negative Database (NDB) can be defined as a database that contains huge amount of data consisting of counterfeit data along with the real data. Anup Patel [5] describes a negative database as representation of original data field with addition of some erroneous data. The main purpose of using negative database along with the positive database is to provide another layer of security to a database along with existing layers like encryption etc[4].The negative database can protect the database from unauthorized user at frontend, shows false results for unauthorized users, using database at back end and secure data transfer.

## 3. PROPOSED SYSTEM

The proposed system presents an effective encrypted outsourced generic database which ensures the confidentiality of the sensitive data and the correctness of the search result even if the dishonest Cloud Service Provider (CSP) intentionally returns empty result. For making the sensitive data more secure that is stored in the outsourced generic database, a framework is introduced which implement the concept of negative database on this for enhancing security. This framework consists of various set of algorithms which manipulate the input data and store it in the database with some counterfeit data. This populated database is referred as negative database. This provides an extra layer of security and make outsourced generic database more secure. In addition, for checking the correctness of the search result as mentioned above, a new verifiable auditing scheme for this outsourced database is introduced together.

### 3.1 System Model

The proposed system consists of four entities for performing the required functions: the data owner, users, cloud service provider and the trusted third party. Fig-2 show the system model.

- Data Owner

The data owner refers to the entity who wants to outsource database to the CSP. The data owner performs the database encryption operations on the sensitive data that is needed to be secured and uploads the encrypted database with the negative data concept .Moreover ,makes a verifiable auditing scheme to check the correctness of the search result when the CSP returns a null answer to the user.

- User

The user is an entity that wants to access database service, who may have limited storage capacity and computing power. Moreover, the user may be corrupted by the dishonest server when they request for file. The user will be getting a null result but he actual knows that the file has been uploaded to the cloud and the CSP is misbehaving.

- Cloud Service Provider(CSP)

The CSP is responsible for storing the encrypted data, file and provides the various database services to the user (e.g., retrieval).

- Trusted Third Party

The trusted third party works as an arbitration center to deal with the dispute between the user and the CSP. The trusted third party is updated each time by the data owner when a file is being uploaded into the CSP after encryption.

This party checks whether the file is actually present in the CSP when the CSP intentionally returns an empty result when the user request for that file. For simplicity, we assume that the authentication between data owner and the user is appropriately completed.

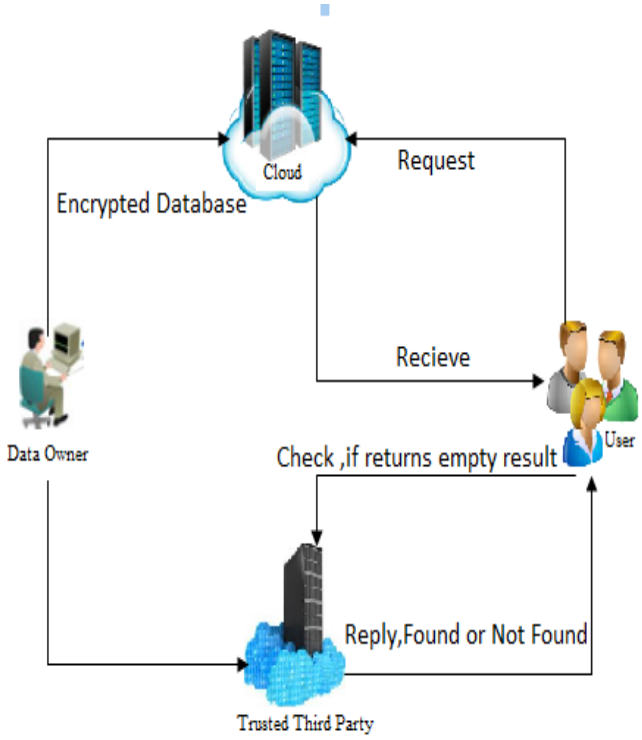


Fig-2: System Model

### 3.2 Securing the Outsourced Generic Database.

It is very important to make the sensitive data stored in the generic database secure. For that we pass the sensitive data through three level of security, an encryption, base32 encoding and negative data creation step. The actual data which is considered to be sensitive is first passed through an encryption algorithm, to generate a cipher text for a given value of sensitive data. Then to make that more secure the base32 encoding algorithm is used as the next step. After that negative data creation algorithm is performed where a counterfeit data along with the real data stored to the database. It provides an additional layer of security as given by the author Gaurav Dubey et.al [4].The Fig-3 shows the algorithm that enhances the security

1. Input=sensitive data
2. Encrypt []= RSA _PublicKey(input)
4. Let cipher_text= Base32_Encode (Encrypt); // converting encrypted text to base 32 to add security.
5. Get first 8 character of the cipher_text and pass it to Break_data () and Negative_data_creation () functions.
    5.1. Get the cipher_text input from the previous section and pass it through Break_data ().
    5.2. Break_data()
        a. Assume a number "n" i.e. the number of value fields in the modified EAV model.
        b. Convert the cipher_text to the length that is multiple of "n" by appending zeros in the start.
        c. Split the cipher_text in "n" groups, each group as cipher_val[][] having n rows and (length/n) columns.
    5.3. Negative_data_creation()
        a. For each row of cipher_val[][] from (i=0) to (i<n) is to be stored in a different value field of a modified EAV generic model.
            i. Take a variable string Str="";
            ii. Take an integer p=0;
            iii. For each (p=0) to (p<(length/n))
                1. gr |= generated_chars (k);
                //Generate "k" random characters.
                2. Str=str+gr+cipher_val[i][p];
                //concatenate
            iv. gr:= generated_chars(k);
            v. str=str+gr;
        b. "n" encrypted strings with some negative data is got, where the useful data is only at the position at multiple of "k+1" and the remaining data is negative.
    5.4. Store these n strings as negative data to the negative database in different "n" fields.
6. Store the remaining of the cipher_text that got from step 4 to another one of the value field and while retrieval concatenate this with the 8 character of the cipher_text,

Fig-3: Database Encryption and Negative Data Creation Algorithm.

### 3.3 Verifiable Auditing Scheme

There exist various motivations for clod service providers (CSP) to behave unfaithfully toward the cloud users for the benefits of possession.

Therefore, it is necessary to offer an efficient auditing service to check the correctness and availability of stored data. The proposed system involves a trusted third party which act as an arbitration center which performs verifiability auditing of the outsourced database that is when the cloud service provider returns an null result to user who request for the file. In this case we assume that the cloud service provider is semi honest-but-curious server that is; the CSP may cheat in search process for benefits. The third party can check whether the file actually present in the cloud even if the CSP intentionally returns an empty result. The concept of Bloom Filter is used by the trusted third party [7] that is used to check whether the data/file is actually present in the cloud if the CSP returns a null result saying that it is not found. The data owner each time updates the Bloom Filter when a file is being uploaded to the cloud. When the user checks for the file for retrieval, he/she can use the trusted third party to check whether the CSP is malicious or not. When the third party checks and delivers that the data/file is found which the CSP told not present, the user can conclude that the cloud is behaving as malicious for their benefits like to reduce computation cost etc. The auditing scheme for outsourced database can be done using hash function, bloom filter etc. The bloom filter uses the concept of hash function but involve more than one hash function. In hashing, the space efficiency is not good compared to bloom filter that is in hashing the hash table store bytes instead of bits that can consume more space.

- Bloom Filter.

A bloom filter consists of a bit array of m bits and k independent hash functions. An empty bloom filter is bit array of m bits, all set to 0.There must also be k different hash functions defined, each of which maps or hashes some set element to one of the m array positions with a uniform random distribution. Typically ,k is a constant ,much smaller than m, which is proportional to the number to be added; the precise choice of k and the constant of proportionality of m are determined by the intended false positive rate of the filter[6][7].

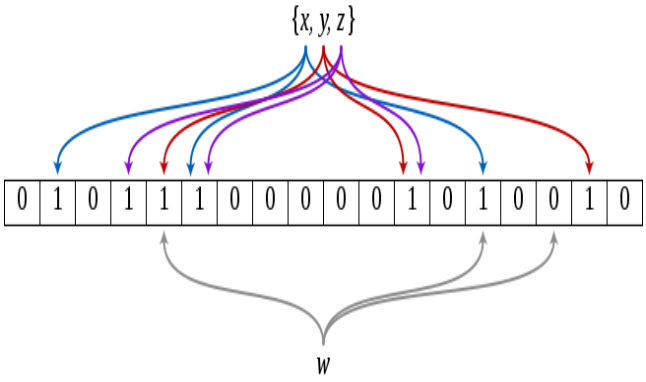


Fig-4. Example of Bloom Filter

An example of a bloom filter, representing the set x, y, z is shown in Fig-4.The coloured arrows shows the positions in the bit array that each set elements is mapped to. The element w is not in the set x, y, z; because it hashes to one bit-array position containing 0.For the above figure m=18 and k=3[8].So a standard bloom filter is a probabilistic data structure that can add element to asset and check if an element is in the set by telling "definitely not in the set "or "possibly in the set". This possibly in the set is exactly why it is called probabilistic. It means false positive are possible but false negatives are not possible.

## 4. SECURITY ANALYSIS

The system ensures the security of the data stored on the outsourced generic database and achieves verifiability of search result.

- Secure in terms of data confidentiality:

The construction ensures the security of data using the concept of negative database. The data that is considered sensitive is not only encrypted but goes through 3 levels of security in addition to RSA encryption and that include Base32 encoding and negative data creation. The unauthorized users will not get meaningful information from the data stored in the database after these steps. Thus ensures data confidentiality.

The execution time for different no of records was analyzed for RSA encryption alone and the 3 layers of security together. It shows that despite of providing extra layers of security other than encryption techniques, it is not giving much change in query execution time. Figure 4.shows the graph for execution time. Here x axis represents the no of records and y axis represent the time (ms).

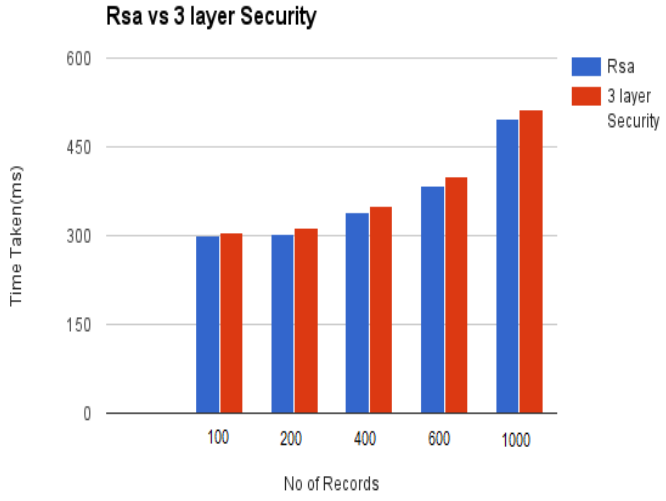


Fig-4: Execution time

- Achieve verifiability of search result:

The system also ensures a proof for the correctness of search result. If the result set is empty that is returned by the CSP for the user requests, the trusted third party is returned as proof for the user. The user checks the result with the third party and then can determine the correctness of the search result whether it is empty. The concept of Bloom filter helps the third party to keep updated with the things in the CSP. The Bloom filter has the advantages of space efficiency, minimum false positive and constant time than that of regular hash function. It have a strong space advantage over other data structures for representing sets such as tries, hash tables etc. A bloom filter with 1 percentage error and an optimal value of k, on the other hand, requires only about 9.6 bits per element regardless of the size of the elements.

## 5. CONCLUSION AND FUTURE WORK

In conclusion, data stored in database are considered very important therefore protecting them is equally important, So we cannot rely on a single security mechanism to protect this data like encryption, there should be deferent layers of mechanisms .Here presented an effective encrypted outsourced generic database which ensures the confidentiality of sensitive data through 3 layers of security and the correctness of the search result using auditing mechanism if the curious but semi honest CSP returns empty result. Here we are checking only the correctness of the search result. Thus as a part of future work we focus on providing a prevention mechanism if the cloud intentionally returns empty result.

## REFERENCES

1)Einar Mykletun,Maithili Narasimha,Gene Tsudik, ”Authentication and Integrity in Outsourced Database”.
2)Jianfeng Wang, Xiaofeng Chen, Xinyi Huang, Ilsun You,Yang Xiang.“Verifiable Auditing for Outsourced Database in Cloud Computing”.IEEE Transactions On Computers, Vol.64, No. 11, November 2015.
3)H.Hacigeumeus,B.Iyarand S.Mehrotra,"Providing database as a service” in Proc. 18th. Conf. Data Eng,2002, pp.29-38.
4) Gaurav Dubey ,Vikram Khurana ,Shelly Sachdeva.“Implementing Security Technique on Generic Database”.IEEE, 2015.
5) Anup Patel,Niveeta Sharma,Magdalini Eirinaki.”Negative database for data security”.ICC Proceedings of the 2009 International Conferene on Computing Engineering and Information.
6) B. Andrei and M. Michael.“Network applications of bloom filters: A survey”. Internet Math., vol. 1, no. 4, pp. 485-509.
7)Matei Ripeanu, Adriana Iamnitchi. “Bloom Filters â Short Tutorial”,
8) https://en.wikipedia.org/wiki/Bloom_filter