International Journal of Information and Computation Technology.
ISSN 0974-2239 Volume 3, Number 9 (2013), pp. 977-984
© International Research Publications House
http://www. irphouse.com /ijict.htm

Secure and Efficient Data Storage in Multi-clouds

Veena Khandelwal

CSE, University College of Engineering, Rajasthan Technical University, Kota, Rajasthan, India.

Abstract

Any information is valuable as long as it has related data. If related data are not put together, the information is meaningless as unrelated data has no value. The mapped information is required only by authenticated users. So there is no necessity to store related information together. If the relations of a database are fragmented into chunks and these chunks are stored at different cloud service providers who provide Database as a Service then it could prevent from any privacy breach and the data stored will be secure. It would also reduce the data transfer costs as the entire data is not always required, for e.g. during updates queries. Also, instead of storage of chunks at a single CSP, if each chunk or fragment is stored at multiple CSPs it ensures availability and also permits concurrent access. Additionally, it would prevent financial loss during cloud outages and also prevent data lock-in. Replicating data chunks at multiple clouds situated at geographically different locations would also have an additional decrease in response time.

Keywords: Multi cloud; security; privacy constraints; Horizontal Fragmentation; Vertical fragmentation; availability; DBaaS.

1. Introduction

Relational database management systems (RDBMSs) are an integral and indispensable component in most computing environments today, and their importance is unlikely to diminish [1]. With the advent of hosted cloud computing and storage, the opportunity to offer a DBMS as an outsourced service is gaining momentum, as witnessed by [2] Amazon's RDS, Microsoft's SQL Azure, Rackspace's Cloud Databases, Google

Compute Engine's Google Cloud SQL, StormDB's StormDB Database, SalesForce's Database.com and Savvis's Symphony.

Such a database-as-a-service (DBaaS) is attractive for two reasons. First, due to economies of scale, the hardware and energy costs incurred by users are likely to be much lower when they are paying for a share of a service rather than running everything themselves. Second, the costs incurred in a well-designed DBaaS will be proportional to actual usage ("pay-per-use")—this applies to both software licensing and administrative costs. The latter are often a significant expense because of the specialized expertise required to extract good performance from commodity DBMSs. By centralizing and automating many database management tasks, a DBaaS can substantially reduce operational costs and perform well.

Since CSPs are separate market entities, data integrity and privacy issues are the more critical ones that need to be addressed. Even though CSPs have standard regulations and powerful infrastructure to ensure data privacy and provide better availability, the reports on privacy breach and service outage have been apparent in the last few years. Also the political influence might become an issue with the availability of services. From the customer's point of view, relying on solo CSP is not very promising.

Client privacy is a tentative issue as all clients do not have the same demand regarding privacy. Some are satisfied with the current policy while others are concerned about their privacy. The proposed system is designed preferably for the clients belonging to the second category for whom privacy is a great concern. The client may not afford the luxury of maintaining private data storage, while they are interested in spending a little more money on maintaining their privacy and availability. Especially companies dealing with financial, educational, health, legal, banking are prominent targets and leaking information of such companies can do significant harm to customers and may sometimes lead to national catastrophe.

The proposed approach will provide the cloud computing users with a decision model that provides better security by distributing data over multiple CSP's in such a way that none of the CSPs can successfully retrieve meaningful information from the data pieces allocated to their servers. Also, it provides the user with better assurance of availability of data by maintaining redundancy in data distribution. In this case, if the service provider suffers a service outage or goes bankrupt, the users still can access his data by retrieving it from another CSP.

2. Related Works

Carlo Curino, Evan Jones,Yang Zhang, Eugene Wu in [3] "Relational Cloud: The Case for a Database Service" in order to allow workloads to scale across multiple computing nodes, divide data into partitions that maximize transaction/query performance. They have developed a new graph-based data partitioning algorithm for transaction-oriented workloads that groups data items according to their frequency of co-access within transactions/queries.

G. Aggarwal, M. Bawa, P. Ganesan, H. Garcia-Molina, K. Kenthapadi, R. Motwani, U. Srivastava, D. Thomas, Y. Xu in [4] "Two Can Keep a Secret: A Distributed Architecture for Secure Database Services" perform efficient partitioning of data using privacy constraints on distributed database.

Subashini, S. and V. Kavitha in [5] "A Metadata Based Storage Model for Securing Data in Cloud Environm-ent" in order to eliminate the disadvantage of storing all data of a client to the same provider, split data into chunks and distribute them among multiple cloud providers.

Ms. P. R. Bhuyar, Dr. A.D. Gawande, Prof. A.B.Deshmukh in [6] "Horizontal Fragmentation Techniques in Distributed Database" fragment a relation horizontally according to locality of precedence of its attributes.

Himel Dev, Tanmoy Sen, Madhusudan Basak and Mohammed Eunus Ali in [7] "An Approach to Protect the Privacy of Cloud Data from Data Mining Based Attacks" inside the Cloud Data Distributor provide each chunk a unique *virtual id* and this *id* is used to identify the chunk within the Cloud Data Distributor and Cloud Providers. This virtualization conceals the identity of a client from the provider.

3. Data Storage Unit

File as a unit of storage. Replications of the entire file at multiple CSP's is beneficial if the file does not contain sensitive data and the queries require all the data. If the file is stored at only one CSP and is not replicated at more than one CSP, a single CSP will get a high volume of remote data accesses. Storing at multiple CSP ensures the availability of data as well as permits concurrent access.

Chunks of file as a unit of storage. Users require only a subset or a fragment of a file and the locality of access is defined on those fragments. Chunk storage permits a number of users to execute concurrently since the users will access different portions of a file. Parallel execution of a single query is also possible. Fragments of a file is usually the appropriate unit of storage. They aim to improve security, reliability, storage costs, update costs and communication costs.

If the data in the file falls under the category 'Normal' and the queries do not require all the data, chunk storage of data can be considered, where the data can be fragmented depending on the type or queries to be executed be it Horizontal or Vertical or Hybrid fragmentation. But if the data to be stored is 'Sensitive', then simple Horizontal, Vertical or Hybrid fragmentation would not provide the required security of data.

4. Data Storage Model

Consider the data is stored at a single Cloud Database as a Service provider. Then there is a single point of failure which will affect data availability. Availability is also an important issue if he runs out of business. Cloud service customers cannot rely on single CSP to ensure storage of vital data. If the database is stored at two DBaaS

providers, there are chances that the two CSPs can act together secretly to achieve a fraudulent purpose and exchange the part of the data with each other and reconstruct the whole data.

In our approach, the client does not have to trust the administrators of any cloud service providers to guarantee privacy. So long as an adversary does not gain access to all the data, data privacy is fully protected. If the client were to obtain database services from different vendors, the chances of an adversary breaking into all the service providers, is greatly reduced. Furthermore, the insider attacks at any one of the cloud service providers do not compromise the security of the system as a whole.

If database security is taken care of by the customer, it also helps the cloud service provider by limiting their liability in case of break-ins into their system. If the service provider is not able to find any valuable information from the contents of the database, nor will the outsider. Existing proposals for secure database service are based on encryption. Although, these attempts are good at securing data in the cloud, they cause large overheads in query processing. Weak encryption algorithms that allow efficient queries leak far too much information and thus do not preserve privacy. On the other hand strong encryption algorithms often necessitate resorting to Plan A for queries, fetching the entire database from the servers which is simply too expensive. Despite increasing processor speeds encryption and decryption are not exactly cheap. New approach is to allow the client to partition its data across three or more logically independent cloud storage systems.

5. Data Privacy

Each file has a privacy level: 'Normal' , 'Sensitive' or 'Critical' [5]. The data which has low value to cloud service providers or attackers and can be allowed to be stored as public data is considered as 'Normal'. The data which is having high value is considered as 'Critical' and the data which has value when mapped with other data is considered as 'Sensitive'. The data which maps 'Sensitive' or 'Critical' data to 'Normal' data is also considered as 'Sensitive'.

The steps to ensure data privacy consists of Categorization, Fragmentation, Distribution, and Replication. Categorize user data as 'Normal', 'Sensitive' or 'Critical'. Split user data into chunks based on the categorization and provide these chunks to CSPs providing Database as a Service. Fragmentation of data is performed in such a fashion so as to ensure that the exposure of the contents of anyone database does not result in a violation of privacy. The presence of three or more cloud service providers enable efficient semantic attribute decomposition, or attribute encoding of sensitive attributes. For example, we can store telephone number by segregating area code at one CSP and telephone number at another CSP. The presence of multiple cloud service providers also enable the storage of many attribute values in unencrypted form. Typically the exposure of a set of attribute values corresponding to a tuple may result in privacy violation while the exposure of only some subsets of it may be harmless. For example individual's name and his credit card number may be a serious privacy

violation. However, exposing the name alone or the credit card number alone may not be a big deal. In such cases we may place individual's name in one CSP while storing his credit card number in another avoiding having to encrypt either attribute. A consequence is that queries involving both names and credit card number may be executed far more efficiently than if the attributes had been encrypted.

Distribution is done according to the sensitivity of data and the reliability of CSP. Reliability is defined in terms of reputation and reliability of the CSP. Distribution restricts an attacker from having access to sufficient number of chunks of data and thus prevents successful extraction of valuable information.

6. Architecture

Architecture as shown in Figure 1. consists of trusted client as well as three or more cloud service providers that provide Database as a Service. The Database as a Service providers provide reliable content storage and data management but are not trusted by the clients to preserve content privacy. The client does not store any persistent data but stores a mapping table describing the storage of various fragments location, their names etc. However the client has access to cheap hardware providing processing power as well as temporary storage and functionality in terms of offering a DBMS frontend, reformulating and optimizing queries and post processing query results, all of which are fairly cheap and can be performed using inexpensive hardware. The client executes queries by transmitting appropriate sub queries to each database and then piecing together the result obtained from the Cloud service providers at the client side.

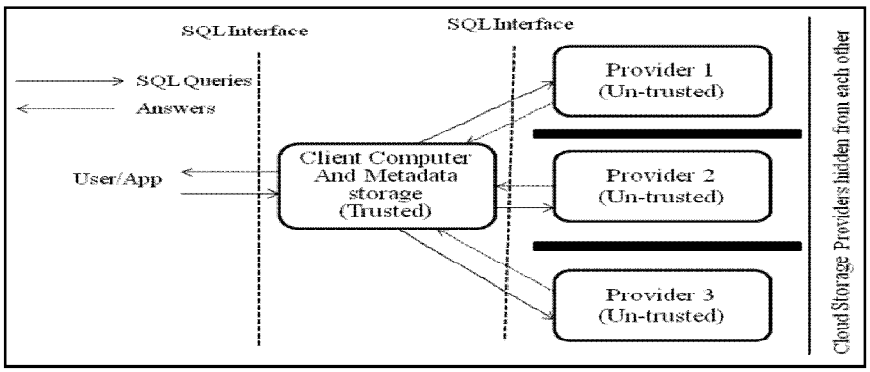

Figure 1: Multi Cloud Storage Architecture.

7. Relational Decomposition

Data Fragmentation and Distribution among multiple CSPs is performed for ensuring security and availability of data in cloud. There are different techniques to partition a relation $R = (A_{1,}, A_2, A_3, ...A_n)$. Traditional decomposition methods are Horizontal,

Vertical and Hybrid Fragmentation [7]. Horizontal Fragmentation partitions a relation along tuples. Vertical Fragmentation partitions a relation along attributes and a Mixed/Hybrid Fragmentation is a combination of Horizontal and Vertical Fragmentations. The fragments should be constructed such that they fulfill Completeness, Reconstruction and Disjointness properties.

Horizontal fragmentation is done based on the selection conditions in the queries to reduce the amount of data during transfers. Horizontal fragmentation has limited use in enabling privacy preserving but it can be of great use in reducing communication costs. Whenever Reads or Writes or Delete operations are performed, they are not always on the entire relation. Horizontal fragmentation is done according to the workload behavior of the queries.

Vertical fragmentation requires key attributes to be present in the sub relations. Key attributes may themselves be sensitive information. A single attribute may become a privacy constraint. In that case, it cannot be stored in open. It can be stored either by encoding the attribute or by storing the hash of the attribute, or by performing semantic attribute decomposition [4]. If it is a primary key attribute, in such a case, introduce a unique tuple ID. We can generate random numbers as tupleIDs ensuring that tupleIDs not already exist. Vertical fragmentation may require semantic attribute decomposition where an attribute A is split into two attributes $A1$ and $A2$. Attribute $A1$ is stored in one of the sub relation and attribute $A2$ is stored in another. For example while storing credit card number issuer identification number is stored in one of the sub relation and individual account identifier and check digit is stored other. Semantic Attribute decomposition will also benefit selection queries based on individual account number or queries that perform aggregation when grouping by issuer identification number could benefit greatly from $A2$ attribute. In absence of $A2$, if credit card numbers were encrypted, query processing becomes more expensive. Attribute encoding can also be used for attributes that need to be kept private. For e.g. salary. Encode salary s as s_1 and s_2 where $s_1 = s + r$ and $s_2 = r$. Store s_1 and s_2 in separate sub relations.

7.1 Specifying the Privacy Constraints

Privacy requirements on a relational schema R are specified as a set or privacy constraints[4]. Each privacy constraint is listed as a set of attributes which alone or together may have some value. The decomposition of relation R should be such that for each privacy constraint on a relational schema R, all the attributes of a privacy constraint should not be a part of any sub relational schema. Some of the attributes of a constraint may be open, some may be encoded and some may be semantically decomposed but all the attributes of a privacy constraint cannot be together in any sub relation schema.

Consider a database in a bank consisting of user information along side with the credit card information.

- A Customer Table {CustomerId (Primary Key (PK)), CustomerName, CustomerAddress, CustomerPhone, CustomerDOB}

- A Membership Table {CustomerId (Primary & Foreign Key (FK)), Pwd, PwdQuestion, PwdAnswer}

Identify the privacy constraints on each table and then perform vertical fragmentation.

1) Customer table Constraints
   a) {CustomerPhone} is a sensitive information.
   b) {CustomerName and CustomerAddress}, {CustomerName and CustomerDOB}
   c) {CustomerAddress, CustomerPhone,Customer DOB}

CustomerPhone is a single privacy constraint and cannot be stored in clear. So it can be stored by semantically decomposing it into Area Code and Telephone number. The constraints specified in (b) and (c)can be addressed by vertical fragmentation of attributes. $R1$(CustomerId, CustomerName), $R2$(Customer Id, CustomerAddress, CustomerTelephoneAreaCode), $R3$(CustomerId, Customer TelephoneNo, CustomerDOB).

2) Membership table Constraints
   a) {Pwd} is a sensitive attribute.
   b) {PwdQuestion, PwdAnswer}

This table alone has no importance. But if the two cloud service providers collude, it has juicy information. So classify it as sensitive. Password is a sensitive attribute and cannot be stored in open. So store the hash value of the password. The constraints specified in (b) can be addressed by vertical fragmentation of attributes. $R1$ (CustomerId, Pwd#, PwdQuestion), $R2$ (CustomerId, Pwd# ,PwdAnswer).

8. Approach

Distribution and Replication. Cloud providers focus on delivering "3 Nines". This availability alone is not enough to meet SLAs of enterprise customers. High end applications require "4 Nines" availability. In order to ensure this high availability, after decomposition the client reformulates the queries and then replicates each decomposed relation (chunk) to at least two CSPs. Replication of each chunk is done at more than one cloud service provider so as to increase cloud availability from 3 nines i.e. 99.9 % to at least 4 nines i.e. 99.99 %.

If one of the chunk Storage Provider goes down, the other chunk Storage Provider will provide the data chunks that were stored on the failed server. The client also maintains a mapping table of the various relations, chunks names, sequence of chunks and storage locations. Each chunk is given a random name. So even if the CSPs collude with each other and exchange the part of the data with each other, they cannot reconstruct the whole data. Even if the adversary is able to find out some information from all the chunks, he is not aware of the proper order of the chunks in making the information have some value. So the data will be secure. Splitting data into smaller chunks restricts data mining attacks also to a great extent as they contain insufficient amount of data.

9. Conclusion

In this paper we have used categorization, fragmentation, distribution and replication techniques to ensure secure and efficient storage in clouds. Data fragmentation uses privacy constraints fragmentation along with horizontal and vertical fragmentation so that any information if in any case becomes available to either the Cloud Database as a Service Provider or to any outsider is of no value. So the data stored is secured. Also the data chunks are replicated at more than one service providers so as to ensure availability, allow concurrent access, restricts data mining attacks and reduce data transfer cost.

References

[1] MultiTenancy and Database as a Service (DBaaS) with ScaleArc Id. [Online] http://www.scalearc.com/ wordpress /wp-content /uploads/2013/01/ScaleArc_iDB_Whitepaper_Multi_Tenancy_And_DBaaS.pdf

[2] 10 of the most useful cloud databases. http://www.networkworld.com/news/2012/121912-cloud-databases-265224.html

[3] Carlo Curino, Evan P. C. Jones, Raluca Ada Popa, Nirmesh Malviya, Eugene Wu, Sam Madden, Hari Balakrishnan, Nickolai Zeldovich "Relational Cloud: The Case for a Database Service" in *Proceedings of 5th Biennial Conference on Innovative Data Systems Research,* Asilomar, CA, January 2011.

[4] G. Aggarwal, M. Bawa, P. Ganesan, H. Garcia-Molina, K. Kenthapadi, R. Motwani, U. Srivastava, D. Thomas, Y. Xu "Two Can Keep a Secret: A Distributed Architecture for Secure Database Services" in *Proceedings of Innovative Data Systems Research Conference* , 2005.

[5] Subashini, S. and V. Kavitha, "A Metadata Based Storage Model for Securing Data in Cloud Environment" in *proceeding of: 2011 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, CyberC 2011,* Beijing, China, October 10-12, 2011.

[6] Ms. P. R. Bhuyar, Dr. A.D. Gawande, Prof. A.B.Deshmukh "Horizontal Fragmentation Techniques in Distributed Database" in *International Journal of Scientific and Research Publications,* Volume 2, Issue 5, May 2012.

[7] Himel Dev, Tanmoy Sen, Madhusudan Basak and Mohammed Eunus Ali "An Approach to Protect the Privacy of Cloud Data from Data Mining Based Attacks" in *Proceedings of, the Third International Workshop on Data Intensive Computing in the Clouds DataCloud 2012,* Salt Lake City, UT.

[8] Abraham Silberschatz, Henry F. Korth, S. Sudarshan "Database system concepts " McGraw-Hill Higher Education, 2006.