International Journal of Computer and Communication Technology

Volume 5 | Issue 1 Article 11

January 2014

A Shared-Disk Database Approach towards Securing Data in the Cloud

Debi Prasad Mishra
School of Computer Engineering, KIIT University, Bhubaneswar, Orissa, India, dpmishra.07@gmail.com

Ranjita Mishra
School of Computer Engineering, KIIT University, Bhubaneswar, Orissa, India, ranjita_mishra86@yahoo.co.in

Animesh Tripathy
School of Computer Engineering, KIIT University, Bhubaneswar, Orissa, India, animesh_tripathy@kiit.ac.in

Recommended Citation
Mishra, Debi Prasad; Mishra, Ranjita; and Tripathy, Animesh (2014) "A Shared-Disk Database Approach towards Securing Data in the Cloud," *International Journal of Computer and Communication Technology*: Vol. 5 : Iss. 1 , Article 11.
DOI: 10.47893/IJCCT.2014.1221
Available at: https://www.interscience.in/ijcct/vol5/iss1/11

A Shared-Disk Database Approach towards Securing Data in the Cloud

Debi Prasad Mishra{dpmishra.07@gmail.com}, ***Ranjita Mishra***{ranjita_mishra86@yahoo.co.in}, ***Animesh Tripathy***{animesh_tripathy@kiit.ac.in}
School of Computer Engineering, KIIT University,
Bhubaneswar, Orissa, India

Abstract- The increasing popularity of cloud computing in distributed computing environment may have positive as well as negative effects on the data security of service consumers. This paper highlights some major security issues existing in current cloud computing environments. The primary issue to be dealt with when talking about security in a cloud is protection of the data. The idea is to construct a privacy preserving repository where data sharing services can update and control the access and limit the usage of their shared data, instead of submitting data to central authorities, and, hence, the shared-disk database architecture will promote data sharing and privacy of data. This paper aims at simultaneously achieving data confidentiality while still keeping the harmonizing relations intact in the cloud. Our proposed scheme enables the data owner to delegate most of computation intensive tasks to cloud servers without disclosing data contents or user access privilege information.

Keywords- cloud, privacy, repository, integration, context-aware

I.INTRODUCTION

The impact during a major change in case of Information Technology industries is not always crystal clear and therefore a balanced wait and see attitude is perfect for an organization. However, some of the changes offer cost benefits, improvement of operations which in turn may offer significant strategic advantage. The benefits and the strategic advancement in these areas happen to be directly proportional to each other. The rise in these two factors will greatly impact the bottom line. The term cloud computing is known to many but a few have actually understood in depth of how it can be beneficial to them. For an organization, in order to jump to the cloud, it is of utmost importance to understand what, why, how and from whom. Since all cloud providers are not the same, many companies, for this reason are concerned over the protection of data as their entire data is being controlled by a central authority. In such a case, companies may encrypt their data before even storing it in the cloud which in turn may lead some to offer greater data security and confidentiality than these companies. But, the fact to be considered is the level of security offered by various cloud providers which raises the security issue since all providers will not offer the same level of security. Generally speaking, security is usually improved by keeping the data in one centralized location.

A. Cloud Overview

Cloud computing is an Internet-based computing, whereby shared resources, software, and information are provided to computers and other devices on demand. It is an archetype shift following the shift from mainframe to client–server. Details are extracted from the users, who no longer have need for expertise in, or control over, the technology infrastructure “in the cloud” that supports them. The term “cloud” is used as a metaphor for the Internet, based on the cloud drawing used in the past to represent the telephone network, and later to depict the Internet in computer network diagrams as an abstraction of the underlying infrastructure it represents. It is seen to have essential characteristics of self-service, on-demand, location-independent, measured access to shared, and elastic resources over the network. That is: ubiquitous, metered access to scalable network services.

Many formal definitions have been proposed in both academia and industry, the one provided by U.S. NIST (National Institute of Standards and Technology) [1] appears to include key common elements widely used in the Cloud Computing community:

Cloud computing is a model for enabling convenient, on demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [2].

B. Cloud Issues

Cloud Computing is still in its infancy, current adoption is associated with numerous issues. In this section, we discuss some of the important issues associated with clouds.

Reliability: Stability of the data storage system is of important consideration in clouds. Generally, people worry about whether a cloud service provider is financially stable and whether their data storage system is trustworthy. Most cloud providers attempt to mollify this concern by using redundant storage techniques, but it is still possible that a service could crash or go out of business, leaving users with limited or no access to their data.

Data Backup: Cloud providers employ redundant servers and routine data backup processes, but some customers worry about being able to control their own backups. Many providers are now offering data dumps onto media or allowing users to back up their data through regular downloads.

Privacy: The Cloud model has been criticized by privacy advocates for the greater ease in which the companies hosting the Cloud services control and monitor communication and data stored between the user and the host company lawfully or unlawfully. There have been efforts to “harmonize” the legal environment by deploying local infrastructure and allowing customers to select “availability zones.”

Security: Cloud service providers employ data storage and transmission encryption, user authentication, and authorization. Many clients worry about the vulnerability of remote data to criminals and hackers. Cloud providers are enormously sensitive to this issue and apply substantial resources to mitigate this problem.

Ownership: Once data has been relegated to the cloud, some worry about losing their rights or being unable to protect the rights of their customers. Many cloud providers address this issue with well-skilled user-sided agreements. According to the agreement, users would be wise to seek advice from their favorite legal representative.

Availability and Performance: Business organizations are worried about acceptable levels of availability and performance of applications hosted in the cloud.

Legal: There are certain points of concern for a cloud provider and a client receiving the service like location of the cloud provider, location of infrastructure, physical location of the data and outsourcing of the cloud provider’s services etc.

I. RELATED WORK

Since this paper is concerned about managing the privacy of data which is sent to a database in the cloud, in this section, general approaches to privacy management for data repositories have been placed, for which various techniques have been developed to ensure that stored data is accessed in a privacy compliant way. Some mechanisms and solutions have been built to encrypt confidential data when it is stored in data repositories, for example solutions using translucent databases [16]. Most of these solutions focus on confidentiality and access control aspects, and have little flexibility in providing policy-driven mechanisms encompassing aspects beyond authentication and authorization. Bertino and Ferrari [17], describe access control policy-based encryption mechanisms for Extensible Markup Language (XML) documents. Bertino and Ferrari describe mechanisms for fine-grained encryption of parts of XML documents, in which decryption keys can either be granted to data receivers or collected from LDAP servers, based on data receivers’ credentials. Hippocratic databases include mechanisms for preserving the privacy of the data they manage. Their proposed architecture is based on the concept of associating privacy metadata (i.e. privacy policies) to data stored in data repositories, along with mechanisms to enforce privacy. The drawback of this approach is that it might require substantial changes to current data repository architectures and, therefore, might take a long time and require substantial investment to succeed. In addition, this approach does not take into account that the management of privacy spans across the database boundaries: such management has to be carried out within a broader context within cloud computing. Although now withdrawn from production, IBM Tivoli Privacy Manage provided mechanisms for defining fine-grained privacy policies and associating them with data. The privacy policies contain authorization constraints along with constraints on contextual information and intent. This approach addressed the privacy management problem purely from an access control perspective within a single enterprise. It did not include additional aspects relevant for privacy management within cloud computing such as trust management and dealing with ongoing privacy obligations dictated by legislation and enterprises’ guidelines. An alternative approach is based on an adaptive privacy management system where data are retrieved from standard data repositories, and parts of these data are encrypted and associated with privacy policies. With the popular central warehouse solution [5], all data is collected by a central authority or a shared-disk database [15] which can execute all queries. Although existing privacy preserving query processing approaches, such as [6], [7], [8] can evaluate a query on randomized data, none of them can handle a series of queries, where some queries need other queries’ results as inputs. In [9], [10], a symmetric searchable encryption scheme and an asymmetric searchable encryption scheme are proposed to store users’ data in a third party. The wrapper development and optimization have been extensively studied in [12], [13], [11] which intuitively decompose the query plan and construct a query plan graph. This aims to make use of current data repository technologies and reduce to the minimum the impact on them, in terms of required changes:

interactions with data repositories can still happen but in a way that confidential data is protected and contextually released, in a fine-grained way, based on the fulfillment of associated privacy policies.

The organization of our work is as follows. Section III focuses on data privacy security issues in the cloud and the advantage of protecting data. Section IV briefly explains the shared-disk database architecture pertaining to our construction. Section V illustrates the thought towards approaching the construction. In Section VI, the proposed scheme has been presented in terms of its security and performance. Section VII presents the system architecture. Section VIII provides concluding remarks for the paper.

III. SECURING DATA IN THE CLOUD

Security is still a serious issue of concern in the cloud. The main reason behind this is that both the customer data and the customer code lies in the provider premises. The following are some of the major issues related to security [3]:

- Server Access Security
- Database Access Security
- Internet Access Security
- Program Access Security
- Data Privacy Security

Of the above mentioned security issues we are focusing on the data privacy security issue. Privacy is the protection of transmitted data from passive attacks [4]. The objective is to ensure that the data required by customer is not being accessed by or not being disclosed to any unauthorized person in the cloud. The client has to assure themselves that the providers are executing their security responsibility. The general trend followed by customers for preserving privacy in cloud is as follows:

- Negotiate precise terms with their cloud provider
- Implement compensating controls where possible
- Assurance of the supplier regarding robustness of their service in preventing data from unauthorized access within a shared environment
- Establishment of an agreement particularly with regard to physical verification
- Get assurance from the cloud provider regarding security and compliance requirements

The responsibility for security is shared between the client and the service provider. To fight back against these issues a repository for preserving privacy of data in cloud has been proposed.

IV. ADVANTAGES OF SHARED-DISK DATABASE

In this paper, shared-disk database architecture is being presented for acceptance of integration requirements from clients to share data in the cloud and maintain their privacy, collect and integrate the appropriate data from data sharing services, and return the integration results to users. Our repository basically concentrates on the harmonizing operations and is beneficial in the following ways:

1) The data sharing services in the cloud possess the ability to update and control the access and usages of their shared data. That is, data can be updated when required and it can be inferred who is using the data and in what way.
2) The sharing of data in the cloud is done based on the need-to-share principle, which states that the dispatched information of the data is adequate to support clients' integration requirements, but carries no extra information of the data.
3) The repository is limited to gathering data from data sharing services and combining the data to satisfy users' requirements. The repository will contain no other information apart from that required to deliver the results to the user and it cannot use this data for other purposes.

V. AN INSPIRING EXAMPLE

Let us consider a commercial information system which collaborates with multiple organizations through sharing and integrating data. The organizations may include company research institutes, project databases, a company database, and a location database for the purpose of studying the developments of popular projects being carried out at different locations. For the sake of simplicity, we assume that the system only communicates with one business research database storing project names and corresponding project numbers, a project database storing the corresponding project numbers on which each employee works. It is assumed that a project can be worked upon by more than one employee, a company database storing all employees' project progress, and a location database storing the location of where each project is being carried out. The databases' schemas are listed below and the data are sited in Table1. This example may be expressed in terms of four SQL queries shown in Table 2, where Q1, Q2, and Q3 generate three temporary tables, Tmp1, Tmp2, and Tmp3, respectively, and the last query, Q4, outputs the final results. Now, we have with us the following database schemas:

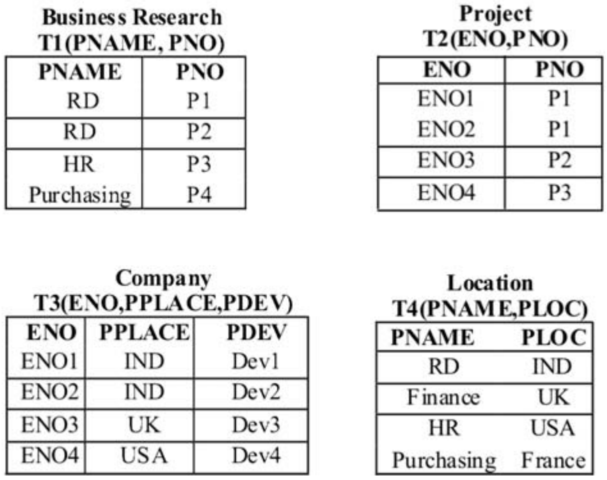
T1 (PNAME; PNO)

T2 (ENO; PNO)

T3 (ENO; PPLACE; PDEV)

T4 (PNAME; PLOC)

TABLE1 DATABASES IN THE EXAMPLE



Business Research
T1(PNAME, PNO)

PNAME	PNO
RD	P1
RD	P2
HR	P3
Purchasing	P4

Project
T2(ENO,PNO)

ENO	PNO
ENO1	P1
ENO2	P1
ENO3	P2
ENO4	P3

Company
T3(ENO,PPLACE,PDEV)

ENO	PPLACE	PDEV
ENO1	IND	Dev1
ENO2	IND	Dev2
ENO3	UK	Dev3
ENO4	USA	Dev4

Location
T4(PNAME,PLOC)

PNAME	PLOC
RD	IND
Finance	UK
HR	USA
Purchasing	France

In our paper, the repository has been restricted for collecting only the required information. Let us consider a simple query Q1→ Tmp1:

SELECT T1.PNO FROM T1 WHERE T1.PNAME="RD".

The obvious output in this case would be P1 and P2. Q1's result will be randomized by the repository and the result will be usable for Q2→Tmp2 :

SELECT T2.ENO FROM Tmp1,T2 WHERE Tmp1.PNO="T2.PNO" because the repository needs some extra information to execute queries as in this case Q1's result is needed by H(Tmp1;PNO) and H(T2;PNO). This simple hash solution can avoid the need for our repository to know Q1's results, but still keep the mapping relation between non RD projects and employees' ENOs. Since H(P3) does not appear in the Q1's hashed result {H(P1);H(P2)}, our repository can find that the employee with ENO4 is not working on RD. To protect the privacy of such information, the concept of Context-Aware Data Sharing [15] is used to randomize Q1's result. The context-awareness implies that when a business research institute shares its database T1 with our repository, it should know that its project number (PNO) data will be used to match the project number data from T2. While the simple hash solution only randomizes the items in Q1's result (i.e., P1; P2), our Context-Aware Data Sharing concept randomizes all PNOs in T1, but ensures that only P1 and P2 can be used to evaluate Q2. Hence, the mapping between non RD projects and ENOs are well protected. Similarly the hashed results are used to evaluate the following queries:

Q3→Tmp3:

SELECT T4.PLOC FROM T4 WHERE T4.PNAME="RD"

and

Q4: SELECT T3.PDEV FROM Tmp2, Tmp3, T3 WHERE T3.ENO=Tmp2.ENO AND T3.PPLACE=Tmp3.PLOC

In this paper, we will be focusing on the above example to illustrate how our repository is being used for studying the developments of various projects at different locations and how the data cannot disclose any additional information about the data of databases T1, T2, T3, and T4.

VI. OUTLINE OF OUR APPROACH

The goal of the approach is to build up a repository to facilitate the data integration and sharing across cloud along with preservation of data confidentiality. In this section, we will present the process of the data integration via our shared-disk database architecture REP. The process can be summarized as follows:

Step 1: The user sends his/her public key P_k and the requirements about data integration to our repository REP.

Step 2: The query plan wrapper of REP analyzes the user's integration requirements and converts them to a query plan graph G, and then decomposes G to a set of subgraphs (G_1;G_2; _ _ _;G_m) and sends the subgraphs to the query plan executor. Every sub graph G_i represents the context of one data sharing service for conducting context-aware data sharing.

Step 3: For every G_i, the query plan executor looks for the corresponding data sharing service S_i and sends G_i to S_i, which prepares the data using the Context-Aware Data Sharing concept and returns all randomized data to the query plan executor.

Step 4: The query plan executor integrates all returned data to execute the G and outputs the results FinalRes of user's request, which is encrypted with the user's public key P_k.

Step 5: REP sends Final Result to the user who then decrypts it with his/her secret key $S_{k.}$

VII. SYSTEM ARCHITECTURE AND ASSUMPTIONS

In continuing data integration systems, the concept of a central and trusted authority collecting all data from data sharing services and computing the integration results for users based on the data collected is usually not valid for data sharing services across various organizations. In our system, as in the Fig. 1 given below, our repository will collect only that data required for generating users' requests. For the analysed set of queries it is assumed that the proposed repository will construct the query plans for users' integration requirements, decompose the query plans, discover and fetch data from distributed data sharing services, assimilate all data together, and, finally, return the final results to users. Further, we assume that the repository is granted the access to the shared data by all data sharing services, and all shared data is well protected. Because the data sharing services use context-aware data sharing concept, the repository cannot learn any extra information from the inferential relations of the information it obtains

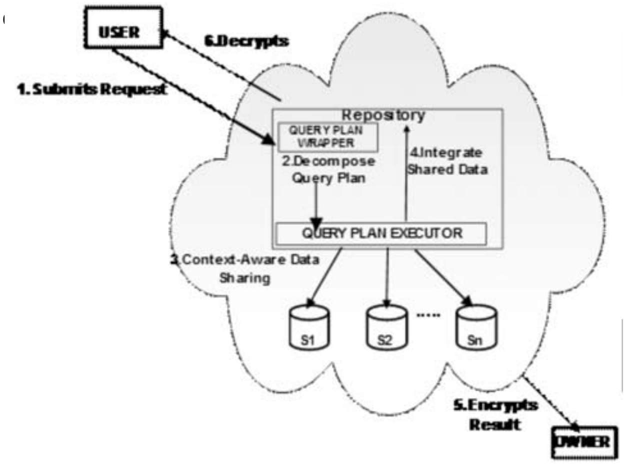


Fig. 1 The Shared-disk database approach for Securing Data in the Cloud

The repository consists of two components: the query plan wrapper and the query plan executor. The query plan wrapper is responsible for scrutinizing integration requirements and constructing query plans for the query plan executor. We assume that the query plan wrapper can select data sharing services and construct a query plan graph from users' integration requirements. The decomposition of the query plan graph into a set of small sub graphs for each data sharing service [11] is used to guide data sharing services to prepare shared data as shown in Fig. 2. The query plan executor is responsible for executing query plans to fetch data from data sharing services and producing the final results. In this paper, we are proposing a secure query plan executor which can execute query plans without additional information about the data of data sharing services.

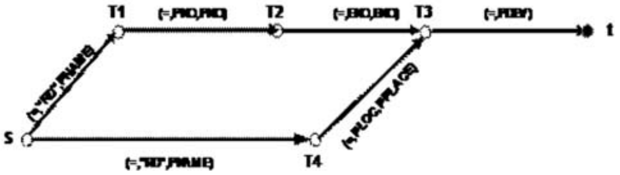


Fig. 2 The Query Plan Graph for the above example

VIII. CONCLUSION

In an emerging discipline, like cloud computing, security needs to be analyzed more frequently. With advancement in cloud technologies and increasing number of cloud users, data security dimensions will continuously increase. In our paper we are laying stress on the security issue in the cloud. The proposed scheme is probably secure under the standard security model [14]. In addition, our proposed scheme is able to support user accountability with minor extension. Whether you are assembling, managing or developing on a cloud computing platform, you need a cloud-compatible database. The shared-disk database architecture, on the other hand, does support elastic scalability. It also supports other cloud objectives such as lower costs for hardware, maintenance, tuning and support. It delivers high-availability in support of Service Level Agreements (SLAs). As with every tectonic shift in technology, there is a Darwinian ripple effect as we realize which technologies support these changes and which are relegated to legacy systems. Because of their compatibility, cloud computing will usher in an ascendance of the shared-disk database. The shared-disk database architecture delivers the critical capabilities required for a robust, cost-effective, and secure cloud security implementation.

IX. FUTURE WORK

In future we will be coming up with a secure development scheme of the hash function and its implementation in public key cryptography for maintaining the privacy of data in the cloud.

REFERENCES

[1] P. Mell and T. Grance, "*Draft nist working definition of cloud computing - v15,*" 21. Aug 2005, 2009.

[2] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "Above the clouds: A Berkeley view of cloud computing," *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28,* 2009.

[3] Rakshit, A. , et. Al, "*Cloud Security Issues*", 2009, IEEE International Conference on Services Computing

[4] Stallings, W., "*Cryptography and Network Security*", 3rd Edition, Pearson Education, New Delhi, 2003, p8.

[5] Kamber, et al., "Data Mining Concepts and Techniques", 2nd Edition, Elsevier, New Delhi, 2009, p12-13

[6] R. Agrawal, A.V. Evfimievski, and R. Srikant, "Information Sharing across Private Databases," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '03), pp. 86-97, 2003.

[7] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order-Preserving Encryption for Numeric Data," Proc. ACM Int'l Conf. Management of Data (SIGMOD '04), pp. 563-574, 2004.

[8] M. Bellare, A. Boldyreva, and A. O'Neill,"Deterministic and Efficiently Searchable Encryption," Advances in Cryptology (CRYPTO '07), pp. 535-552, 2007.

[9] D. Boneh, G.D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public Key Encryption with Keyword Search," Advances in Cryptology (EUROCRYPT '04), pp. 506-522, 2004.

[10] D.X. Song, D. Wagner, and A. Perrig,"Practical Techniques for Searches on Encrypted Data," Proc. IEEE Symp. Security and Privacy (S&P '00), pp. 44-55, 2000.

[11] Yau, Stephan S., et. al, "*A Privacy Preserving Repository for Data Integration across Data Sharing Services*", IEEE Transactions on Services Computing, Vol 1, No-3, July-September 2008

[12] S. Adali, K.S. Candan, Y. Papakonstantinou, and V.S. Subrahmanian, "Query Caching and Optimization in Distributed Mediator Systems," Proc. ACM Int'l Conf. Management of Data (SIGMOD '96), pp. 137-148, 1996.

[13] B. Yu, G. Li, K.R. Sollins, and A.K.H. Tung,"Effective Keyword- Based Selection of Relational Databases," Proc. ACM Int'l Conf. Management of Data (SIGMOD '07), pp. 139-150, 2007.

[14] Shucheng Yu., Cong Wang†, Kui Ren , and Wenjing Lou , "*Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing*", IEEE INFOCOM 2010 Proceedings

[15] Hogan, M., "*Cloud Computing and Databases*", Scale DB Inc., 2008

[16] Wayner P (2002) Translucent databases. Flyzone Press, Baltimore

[17] Bertino E, Ferrari E (2002) Secure and selective dissemination of XML documents. ACM Trans Inf Syst Secur 5(3):290–331