Distributed, Concurrent, And Independent Access To Encrypted Cloud Databases

[1]DIVYA SAMA, [2]MR.K.CHAITHANYA

1.Pg Scholar, Sri Venkateswara Engineering College, Suryapet,Telangana.INDIA.

2. M.Tech,Asst.Professor, Sri Venkateswara Engineering College, Suryapet,Telangana.INDIA

ABSTRACT:

Placing critical data in the hands of a cloud provider should come with the guarantee of security and availability for data at rest, in motion, and in use. Several alternatives exist for storage services, while data confidentiality solutions for the database as a service paradigm are still immature. The proposed novel architecture integrates cloud database services with data confidentiality and the possibility of executing concurrent operations on encrypted data. This is the novel solution supporting geographically distributed clients to connect directly to an encrypted cloud database, and to execute concurrent and independent operations including those modifying the database structure. The proposed architecture has the further advantage of eliminating intermediate proxies that limit the elasticity, availability, and scalability properties that are intrinsic in cloud-based solutions. The efficacy of the proposed architecture is evaluated through theoretical analyses and extensive experimental results based on a prototype implementation subject to the TPC-Standard benchmark for different numbers of clients and network latencies.

1. INTRODUCTION:

In a cloud context, where critical information is placed on infrastructures of untrusted third parties ensuring data confidentiality is of paramount importance. This requirement imposes clear data management choices: original plain data must be accessible only by trust expertise that does not include cloud providers, intermediaries, and Internet; in any untrusted context, data must be encrypted. Satisfying these goals has different levels of complexity depending on the type of cloud service. There are several solutions ensuring confidentiality for the storage as a service paradigm while guaranteeing confidentiality in the database as a service (DBaaS) paradigm is still an open research area. This context proposes Secure DBaaS as the novel solution that allows cloud tenants to take full advantage of DBaaS qualities, such as availability, reliability, and elastic scalability, without exposing unencrypted

data to the cloud provider. The architecture design was motivated by a threefold goal: to allow multiple, independent, and geographically distributed clients to execute concurrent operations on encrypted data, including SQL statements that modify the database structure; to preserve data confidentiality and consistency at the client and cloud level; to eliminate any intermediate server between the cloud client and the cloud provider. The possibility of combining availability, elasticity, and scalability of a typical cloud DBaaS with data confidentiality is demonstrated through a prototype of Secure DBaaS that supports the execution of concurrent and independent operations to the remote encrypted database from many geographically distributed clients as in any unencrypted DBaaS setup. To achieve these goals, Secure DBaaS integrates existing cryptographic schemes, isolation mechanisms, and novel strategies for management of encrypted metadata on the untrusted cloud database. This paper contains a theoretical discussion about solutions for data consistency issues due to concurrent and independent client accesses to encrypted data. In this context, it does not apply fully homomorphic encryption schemes because of their excessive computational complexity. The SecureDBaaS architecture is tailored to cloud platforms and does not introduce any intermediary proxy or broker server between the client and the cloud provider Workloads including modifications to the database structure are also supported by Secure DBaaS, but at the price of overheads that seem acceptable to achieve the desired level of data confidentiality. The motivation of these results is that network latencies, which are typical of cloud scenarios, tend to mask the performance costs of data encryption on response time. This paper is organized as follows, section 1 discusses the introduction, and section 2 describes related work. Section 3 details the system design and implementation. Section 4 presents the performance evaluations of our system design. Finally, section 5 presents some concluding remark.

2. RELATED WORK " A View of Cloud Computing" M. Armbrust [1], has developed with innovative ideas for new Internet services no longer require the large capital outlays in hardware to deploy their service or the human expense to operate it. Cloud Computing will grow, so developers should take it into account. Moreover: 1. Applications Software needs to both scale down rapidly as well as scale up, which is a new requirement. Such software also needs a pay-for-use licensing model to match

needs of Cloud Computing. 2. Infrastructure Software needs to be aware that it is no longer running on bare metal but on VMs. Moreover, billing needs to build in from the start. 3. Hardware Systems should be designed at the scale of a container (at least a dozen racks), which will be is the minimum purchase size. "SPORC: Group Collaboration Using Untrusted Cloud Resources" A.J. Feldman, W.P. Zeller, M.J. Freedman, and E.W. Felten [2], have described Cloud-based services are an attractive deployment model for user-facing applications like word processing and calendaring. In SPORC, a server observes only encrypted data and cannot deviate from correct execution without being detected. SPORC allows concurrent, low-latency editing of shared state, permits disconnected operation, and supports dynamic access control even in the presence of concurrency Acknowledgments. "Secure Untrusted Data Repository (SUNDR)" J. Li, M. Krohn, D. Mazie` res, and D. Shasha, [3] have proposed SUNDR is a network file system designed to store data securely on untrusted servers. SUNDR's protocol achieves a property called fork consistency, which guarantees that clients can detect any integrity or consistency failures as long as they see each other's file modifications. Measurements of our implementation show

performance that is usually close to and sometimes better than the popular NFS file system. "Depot: Cloud Storage with Minimal Trust" P. Mahajan, S. Setty, S. Lee, A. Clement, L. Alvisi, M. Dahlin, and M. Walfish [4] have described the design, implementation, and evaluation of Depot, a cloud storage system that minimizes trust assumptions. Depot began with an attempt to explore a radical point in the design space for cloud storage: trust no one. "Providing Database as a Service" H. Hacigu¨ mu¨ s, B. Iyer, and S. Mehrotra [5], have proposed a new paradigm for data management in which a third party service provider hosts "database as a service" providing its customers seamless mechanisms to create, store, and access their databases at the host site. The authors introduced NetDB2, an internet-based database service built on top of DB2 that provides users with tools for application development, creating and loading tables, and performing queries and transactions. "Fully Homomorphic Encryption Using Ideal Lattices" C. Gentry [6], has proposed a fully homomorphism encryption scheme – i.e., a scheme that allows one to evaluate circuits over encrypted data without being able to decrypt. The circuit privacy of E2 immediately implies the (leveled) circuit privacy of our (leveled) fully

homomorphism encryption scheme. "Crypt: Protecting Confidentiality with Encrypted Query Processing" R.A. Popa, C.M.S. Redfield, N. Zeldovich, and H. Balakrishnan [7], have described, Crypt is a system that provides practical and provable confidentiality in the face of these attacks for applications backed by SQL databases. It works by executing SQL queries over encrypted data using a collection of efficient SQL-aware encryption schemes. "Efficiency and Security Trade-Off in Supporting Range Queries on Encrypted Databases" J. Li and E. Omiecinski [8], had discussed concerns about protecting sensitive information of data and queries from adversaries in the DAS model. Data and queries need to been cryptic, while the database service provider should be able to efficiently answer queries based on encrypted data and queries. "Distributing Data for Secure Database Services," V.Ganapathy, D. Thomas, T. Feder, H. Garcia-Molina, and R.Motwani [9] have proposed, the advent of database services has resulted in privacy concerns on the part of the client storing data with third party database service providers. This paper provide algorithms for (1) distributing data: our results include hardness of approximation results and hence a heuristic greedy algorithm for the distribution problem (2) partitioning the query at the

client to queries for the servers is done by a bottom up state based algorithm. Finally the results at the servers are integrated to obtain the answer at the client. "How to Share a Secret," A. Shamir[10], has described how to divide data D into n pieces in such a way that D is easily reconstruct able from any k pieces, but even complete knowledge of k - 1 pieces reveals absolutely no information about D. This technique enables the construction of robust key management schemes for cryptographic systems that can function securely and reliably even when misfortunes destroy half the pieces and security breaches expose all but one of the remaining pieces.

3. METHODOLOGY This paper propose a novel architecture that integrates cloud database services with data confidentiality and the possibility of executing concurrent operations on encrypted data. This is the novel solution supporting geographically distributed clients to connect directly to an encrypted cloud database, and to execute concurrent and independent operations including those modifying the database structure. The proposed architecture has the further advantage of eliminating intermediate proxies that limit the elasticity, availability, and scalability properties that are intrinsic in cloud-based solutions. Secure

DBaaS provides several original features that differentiate it from previous work in the field of security for remote database services.

4.CONCLUSION This paper proposes an innovative architecture that guarantees confidentiality of data stored in public cloud databases. Proposed solution does not rely on an intermediate proxy that is considered a single point of failure and a bottleneck limiting availability and scalability of typical cloud database services. A large part of the research includes solutions to support concurrent SQL operations (including statements modifying the database structure) on encrypted data issued by heterogeneous and possibly geographically dispersed clients. The proposed architecture does not require modifications to the cloud database, and it is immediately applicable to existing cloud DBaaS such as the experimented PostgreSQL plus Cloud Database, Windows Azure, and Expound. There are no theoretical and practical limits to extend our solution to other platforms and to include new encryption algorithms. It is worth serving that experimental results based on the TPC-C standard benchmark show that the performance impact of data encryption on response time becomes negligible because it is masked by network latencies

that are typical of cloud scenarios. In particular, concurrent read and write operations that do not modify the structure of the encrypted database cause overhead. Dynamic scenarios characterized by (possibly) concurrent modifications of the database structure are supported, but at the price of high computational costs. These performance results open the space to future improvements that we are investigating.

References

[1] M. Armbrust et al., "A View of Cloud Computing," Comm. of the ACM, vol. 53, no. 4, pp. 50-58, 2010.

[2] A.J. Feldman, W.P. Zeller, M.J. Freedman, and E.W. Felten, "SPORC: Group Collaboration Using Untrusted Cloud Resources," Proc. Ninth USENIX Conf. Operating Systems Design and Implementation, Oct. 2010.

[3] J. Li, M. Krohn, D. Mazie` res, and D. Shasha, "Secure Untrusted Data Repository (SUNDR)," Proc. Sixth USENIX Conf. Operating Systems Design and Implementation, Oct. 2004.

[4] P. Mahajan, S. Setty, S. Lee, A. Clement, L. Alvisi, M. Dahlin, and M. Walfish, "Depot: Cloud Storage with Minimal Trust",

ACMTrans. Computer Systems, vol. 29, no. 4, article12, 2011.

[5] H. Hacigu¨ mu¨ s¸, B. Iyer, and S. Mehrotra, "Providing Database as a Service", Proc. 18th IEEE Int'l Conf. Data Eng.Feb.2002.

[6] C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattices", Proc. 41st Ann. ACM Symp. Theory of Computing May 2009.

7] R.A. Popa, C.M.S. Redfield, N. Zeldovich, and H. Balakrishnan, "CryptDB: Protecting Confidentiality with Encrypted Query Processing," Proc. 23rd ACM Symp. Operating Systems Principles, Oct. 2011.

[8] J. Li and E. Omiecinski, "Efficiency and Security TradeOff in Supporting Range Queries on Encrypted Databases," Proc. 19thAnn. IFIP WG 11.3 Working Conf. Data and Applications Security, Aug. 2005.

[9] V.Ganapathy, D. Thomas, T. Feder, H. Garcia-Molina, and R.Motwani, "Distributing Data for Secure Database Services," Proc. Fourth ACM Int'l Workshop Privacy and Anonymity in the Information Soc., Mar. 2011.

[10] A. Shamir, "How to Share a Secret", Comm. of the ACM, vol. 22, no. 11, pp. 612-613, 1979.