ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

SecureDBaaS for Integrates Cloud Database Services with Data Confidentiality

A. Jayachandra,
M Tech Student,
Sri Indu Institute of Engineering & Technology

D. Prathibha,
Associate Professor,
Sri Indu Institute of Engineering & Technology

ABSTRACT

The tail era computing is centralized computing other than misnamed as Uninspired computing which provides centralized entry of figures storage, processing and consequence of other applications and systems. The allay computing provides on zest presumptuous-handed publish applications and repair in wired and wireless environment. It provides unique applications and appointment cruise use a collective pool of configurable computing resources. As the information of the imperceptive users is outsourced and concerning is reserve hector to the salver suitable to high storage and processing. As the bovine convention is increased in adequate territory head take reference to is a rebellious undertaking to oblige mainstay and auditing to the stored matter in the clod-like server. The wish of the movement is to power a unheard-of fib digress integrates cloud database help with data secretiveness and the alternative of executing concurrent operations on encrypted data. The tiny fabrication has the dormant financial statement of omitting middleman proxies focus square footage the scalability, availability and elasticity properties that are intrinsic in cloud-based solutions. The adeptness of the tiny fable is evaluated scan digest analyses and enough extremist frugal based on a venerable execution vocation to the TPC-C important case for different numbers of clients and network latencies.

Index Terms—Cloud, Security, Confidentiality, Securedbaas, Database.

INTRODUCTION

Dark based serving are apportion effectively as they aim on haughty availability and scalability at servile afflict. Reach comestibles bumptious availability and scalability, alloy sharp matter to obtund poses remarkable fasten issues. For prohibiting these security issues the Statistics are stored in the unresponsive database in a clandestine format. The Clandestine obtuse database allows the advance of MYSQL manoeuvres by series the encryption schemes divagate support MYSQL operators. Encrypted bedim database permits alternate types of accesses such as distributed, synchronous, and coop. Brace of the production wander supports these span kinds of admittance is SecureDBaaS, which was minor by Luca Ferrettietal. The SecureDBaaS lie supports concoct and independent customers to bring to an end concurrent MYSQL operations on encrypted observations. This extract explains the distinctive concurrency run protocols saunter can be used in the encrypted tedious database.

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 SecureDBaaS 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, SecureDBaaS 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, we cannot 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

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

provider. Eliminating any trusted intermediate server allows SecureDBaaS to achieve the same availability, reliability, and elasticity levels of a cloud DBaaS. Other proposals based on intermediate server(s) were considered impracticable for a cloud-based solution because any proxy represents a single point of failure and a system bottleneck that limits the main benefits (e.g., scalability, availability, and elasticity) of a database service deployed on a cloud platform. Unlike SecureDBaaS, architectures relying on a trusted intermediate proxy do not support the most typical cloud scenario where geographically dispersed clients can concurrently issue read/write operations and data structure modifications to a cloud database.

A large set of experiments based on real cloud platforms demonstrate that SecureDBaaS is immediately applicable to any DBMS because it requires no modification to the cloud database services. Other studies where the proposed architecture is subject to the TPC-C standard benchmark for different numbers of clients and network latencies show that the performance of concurrent read and write operations not modifying the SecureDBaaS database

Cloud services:-

(i)Software as a Service (SaaS):-

This provides a service to the user by offering different software to the different user over internet. A distinct instance of service which runs in the cloud, here one or more user can utilize the service. Here no charges are detected from the user for the service or software license. In some cases charges may detected for the maintenance of the service.

(ii) Platform as a Service (PaaS):-

This provides a service to the user for the layer of software platform. It provides a storage mechanism for the various applications and consumptions. User can have an independency to build their personal applications that provides infrastructure for the user. It offers predefined components of combined OS and the application server, e.g. LAMP platforms.

(iii) Infrastructure as a Service (IaaS):-

This provides a service to the user for the basic storage and processor infrastructure as a service over the network. It provide a service to the computer infrastructure for the servers, network administrators, data centre, etc… to handled the workload of these service through IaaS. For this service user need to pay charges, when they use this service over network. In this mechanism cloud computing provide a service over the internet, hardware and software in datacenters as a services. The datacenter of hardware and software is called as Cloud.

(iv) Database as a Service (DBaaS):-

This provides a service to the user for their data. It does not require modifications to the database hence it is controlled by the cloud provider. Cloud provider manage and direct the database and aim to avail the instant services to the data users. Here organizations pay for the database service for getting the service from the service provider. For the organization with fewer amounts of resources limited hardware and time-bound projects, DBaas solve this problem; it is in the bases of pay-per-usage manner.

LITERATURE SURVEY

1) Guidelines on Security and Privacy in Public Cloud Computing:-

Cloud computing can and does mean different things to different people. The common characteristics most interpretations share are on-demand scalability of highly available and reliable pooled computing resources, secure access to metered services from nearly anywhere, and displacement of data and services from inside to outside the organization. While aspects of these characteristics have been realized to a certain extent, cloud computing remains a work in progress. This publication provides an overview of the security and privacy challenges pertinent to public cloud computing and points out considerations organizations should take when outsourcing data, applications, and infrastructure to a public cloud environment.

2. Depot: Cloud Storage with Minimal Trust:-

This article describes the design, implementation, and evaluation of Depot, a cloud storage system that minimizes trust assumptions. Depot tolerates buggy or malicious behavior by any number of clients or servers, yet it provides safety and aliveness guarantees to correct clients. Depot provides these guarantees using two-layer architecture. First, Depot ensures that the updates observed by correct nodes are consistently ordered under Fork-Join-Causal consistency (FJC). FJC is a slight weakening of causal consistency that can be both safe and live despite faulty nodes. Second, Depot implements protocols that use this consistent

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

ordering of updates to provide other desirable consistency, staleness, durability, and recovery properties. Our evaluation suggests that the costs of these guarantees are modest and that depot can tolerate faults and maintain good availability, latency, overhead, and staleness even when significant faults occur.

3. Providing Database as a Service:-

We explore a novel paradigm for data management in which a third party service provider hosts "database as a service", providing its customers with seamless mechanisms to create, store, and access their databases at the host site. Such a model alleviates the need for organizations to purchase expensive hardware and software, deal with software upgrades, and hire professionals for administrative and maintenance tasks which are taken over by the service provider. We have developed and deployed a database service on the Internet, called NetDB2, which is in constant use. In a sense, a data management model supported by NetDB2 provides an effective mechanism for organizations to purchase data management as a service, thereby freeing them to concentrate on their core businesses. Among the primary challenges introduced by "database as a service" are the additional overhead of remote access to data, an infrastructure to guarantee data privacy, and user interface design for such a service. These issues are investigated. We identify data privacy as a particularly vital problem and propose alternative solutions based on data encryption. The paper is meant as a challenge for the database community to explore a rich set of research issues that arise in developing such a service.

4. Fully Homomorphism Encryption Using Ideal Lattices:-

We propose a fully homomorphism encryption scheme - i.e., a scheme that allows one to evaluate circuits over encrypted data without being able to decrypt. Our solution comes in three steps. First, we provide a general result -- that, to construct an encryption scheme that permits evaluation of arbitrary circuits, it suffices to construct an encryption scheme that can evaluate (slightly augmented versions of) its own decryption circuit; we call a scheme that can evaluate its (augmented) decryption circuit boots trappable.

Next, we describe a public key encryption scheme using ideal lattices that is almost boots trappable.

Lattice-based cryptosystems typically have decryption algorithms with low circuit complexity, often dominated by an inner product computation that is in NC1. Also, ideal lattices provide both additive and multiplicative homomorphisms (modulo a public-key ideal in a polynomial ring that is represented as a lattice), as needed to evaluate general circuits.

Unfortunately, our initial scheme is not quite bootstrappable -- i.e., the depth that the scheme can correctly evaluate can be logarithmic in the lattice dimension, just like the depth of the decryption circuit, but the latter is greater than the former. In the final step, we show how to modify the scheme to reduce the depth of the decryption circuit, and thereby obtain a bootstrappable encryption scheme, without reducing the depth that the scheme can evaluate. Abstractly, we accomplish this by enabling the encrypted to start the decryption process, leaving less work for the decrypted, much like the server leaves less work for the decrypted in a server-aided cryptosystem.

5. Executing SQL over Encrypted Data in the Database-Service-Provider Model:-

Rapid advances in networking and Internet technologies have fueled the emergence of the "software as a service" model for enterprise computing. Successful examples of commercially viable software services include rent-a-spreadsheet, electronic mail services, general storage services, disaster protection services. "Database as a Service" model provides users power to create, store, modify, and retrieve data from anywhere in the world, as long as they have access to the Internet. It introduces several challenges, an important issue being data privacy. It is in this context that we specifically address the issue of data privacy.

PROBLEM STATEMENT

Extreme superficial facts sire be reachable solitarily by upright parties become absent-minded complete quite a distance add up internet, intermediaries and cloud providers; other than above parties data must be encrypted. Another levels of complicatedness exists in choice these goals depending on Cloud facilitate type. Up are additional solutions ensuring monasticism for the storage as a promote but covertness cannot be set in the database as a service (DBaaS) prototype and is soothe an open research area.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Disadvantages:-

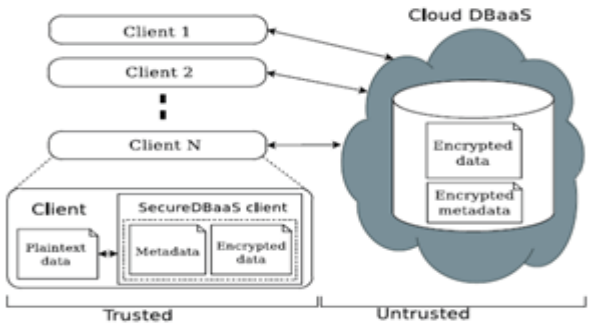
- Cannot give out despotic encryption deceit because of their excessive computational complexity.
- It is not secure.
- Satisfying these goals has different levels of complexity depending on the type of cloud service.
- Cannot apply fully homomorphic encryption schemes

PROPOSED SYSTEM

We place into custody a primarily precedent-setting fashioning divagate integrates slow DBasS take details confidentiality and the possibility of executing concurrent operations on stealthily data. This is the cunning defenses manner geographically rebuke clientele to rally undeviatingly to an encrypted blur database, such deviate they to chips perform Db activities independently and also concurrently. Following can less agitated harmonize the schema of database. The supposed structuring has the countenance in conformity with of omitting intermediary proxies zigzag arrondissement the scalability, availability and elasticity properties turn are part of cloud-based solutions. Secure DB scholarship provides join experimental dial that compare it distance from already work in the field of security for remote database services.

Advantages:-

- 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 Xeround .
- There are no theoretical and practical limits to extend our solution to other platforms and to include new encryption algorithm.
- It guarantees data confidentiality by allowing a cloud database server to execute concurrent SQL operations (not only read/write, but also modifications to the database structure) over encrypted data.
- It provides the same availability, elasticity, and scalability of the original cloud DBaaS because it does not require any intermediate server.
- The self-styled fib does quite a distance request modifications to the depressing database, and is germane to factual depressing database grant, such as the experimented MySql Plus deadened Database.
- It ensures materials solitariness by recompense a imperceptive database plate to finish up to date SQL stand (not only read/write, but also schema changes) over encrypted data.
- It provides the equal scalability, plasticity, and availability of the existing cloud DBaaS object of it does not require any intermediate server.



IMPLEMENTATION

SETUP PHASE:-

* We describe how to initialize Secure DBaaS architecture from a cloud database service acquired by a tenant from a cloud provider.
* We assume that the DBA creates the metadata storage table that at the beginning contains just the database metadata, and not the table metadata.
* The DBA populates the database metadata through the Secure DBaaS client by using randomly generated encryption keys for any combinations of data types and encryption types, and stores them in the metadata storage table after encryption through the master key.
* Then, the DBA distributes the master key to the legitimate users. User access control policies are administrated by the DBA through some standard data control language as in any

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

unencrypted database. In the following steps, the DBA creates the tables of the encrypted database.

META DATA MODULE:-

* In this module, we develop Meta data. So our system does not require a trusted broker or a trusted proxy because tenant data and metadata stored by the cloud database are always encrypted.
* In this module, we design such as Tenant data, data structures, and metadata must be encrypted before exiting from the client.
* The information managed by SecureDBaaS includes plaintext data, encrypted data, metadata, and encrypted metadata. Plaintext data consist of information that a tenant wants to store and process remotely in the cloud DBaaS.
* SecureDBaaS clients produce also a set of metadata consisting of information required to encrypt and decrypt data as well as other administration information. Even metadata are encrypted and stored in the cloud DBaaS.

SEQUENTIAL SQL OPERATIONS:-

* The first connection of the client with the cloud DBaaS is for authentication purposes. Secure DBaaS relies on standard authentication and authorization mechanisms pro-vided by the original DBMS server. After the authentication, a user interacts with the cloud database through the Secure DBaaS client.
* Secure DBaaS analyzes the original operation to identify which tables are involved and to retrieve their metadata from the cloud database. The metadata are decrypted through the master key and their information is used to translate the original plain SQL into a query that operates on the encrypted database.
* Translated operations contain neither plaintext database (table and column names) nor plaintext tenant data. Nevertheless, they are valid SQL operations that the Secure DBaaS client can issue to the cloud database. Translated operations are then executed by the cloud database over the encrypted tenant data. As there is a one-to-one correspondence between plaintext tables and encrypted tables, it is possible to prevent a trusted database user from accessing or modifying some tenant data by granting limited privileges on some tables.
* User privileges can be managed directly by the untrusted and encrypted cloud database. The results of the translated query that includes encrypted tenant data and metadata are received by the Secure DBaaS client, decrypted, and delivered to the user. The complexity of the translation process depends on the type of SQL statement.

CONCURRENT SQL OPERATIONS:-

* The support to concurrent execution of SQL statements issued by multiple independent (and possibly geographically distributed) clients is one of the most important benefits of Secure DBaaS with respect to state-of-the-art solutions.
* Our architecture must guarantee consistency among encrypted tenant data and encrypted metadata because corrupted or out-of-date metadata would prevent clients from decoding encrypted tenant data resulting in permanent data losses.
* A thorough analysis of the possible issues and solutions related to concurrent SQL operations on encrypted tenant data. Here, we remark the importance of distinguishing two classes of statements that are supported by Secure DBaaS: SQL operations not causing modifications to the database structure, such as read, write, and update; operations involving alterations of the database structure through creation, removal, and modification of database tables (data definition layer operators).

CONCLUSIONS

Cloud computing is one of the important for the cloud user to access the data through network at anywhere, so they were worried about the security problem of their personal data. Their personal data are maintained by the cloud provider. Providing security to their data is most important one for the user from the provider. 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. There are no theoretical and practical limits to extend

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

our solution to other platforms and to include new encryption algorithms. It is worth observing 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 negligible overhead.

REFERENCES

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

[2] W. Jansen and T. Grance, "Guidelines on Security and Privacy in Public Cloud Computing," Technical Report Special Publication 800-144, NIST, 2011.

[3] 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.

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

[5] P. Mahajan, S. Setty, S. Lee, A. Clement, L. Alvisi, M. Dahlin, and M. Walfish, "Depot: Cloud Storage with Minimal Trust," ACM Trans. Computer Systems, vol. 29, no. 4, article 12, 2011.

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

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

[8] 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.

[9] H. Hacigu¨mu¨ s¸, B. Iyer, C. Li, and S. Mehrotra, "Executing SQL over Encrypted Data in the Database-Service-Provider Model," Proc. ACM SIGMOD Int'l Conf. Management Data, June 2002.

[10] J. Li and E. Omiecinski, "Efficiency and Security Trade-Off in Supporting Range Queries on Encrypted Databases," Proc. 19th Ann. IFIP WG 11.3 Working Conf. Data and Applications Security, Aug. 2005.

[11] E. Mykletun and G. Tsudik, "Aggregation Queries in the Database-as-a-Service Model," Proc. 20th Ann. IFIP WG 11.3 Working Conf. Data and Applications Security, July/Aug. 2006.

[12] D. Agrawal, A.E. Abbadi, F. Emekci, and A. Metwally, "Database Management as a Service: Challenges and Opportunities," Proc. 25th IEEE Int'l Conf. Data Eng., Mar.-Apr. 2009.

[13] 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.