International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 3, Issue 6, November-December 2014 **ISSN 2278-6856**

BUILDING ENCRYPTED DATA OVER MAS BY ACHIEVING DATA CONFIDENTIALITY

[1]V.MUTHURANI , [2]A.LOURDESMARY

[1]M.E (COMPUTER SCIENCE AND ENGINEERING)
SCAD COLLEGE OF ENGINEERING AND TECHNOLOGY, CHERANMAHADEVI

[2]ASSOCIATE PROFESSOR OF CSE DEPARTMENT
SCAD COLLEGE OF ENGINEERING AND TECHNOLOGY, CHERANMAHADEVI

Abstract

In a cloud arena, inserting important information below untrusted third parties, risks the confidentiality of data. Guaranteeing confidentiality within the Database as a service (DBaaS) paradigm remains a problem. Therefore to resolve that Confidential Concurrent to Secure DBaas is projected because the initial resolution to produce availability, security, accessibility and reliability while not exposing unencrypted information to the cloud provider. It additionally permits multiple, freelance and regionally distributed clients to execute synchronal operations on encrypted and preserve information confidentiality and consistency at the consumer and cloud level. It eliminate any intermediate server between the cloud consumer and also the cloud provider.To realize this, Confidential concurrent to Secure DBaaS integrates existing cryptographic schemes, isolation mechanisms and management of encrypted information on the untrusted cloud information.

IndexTerms:-Cloud, Security, confidentiality, SecureDBaas, Access Control

1. INTRODUCTION

Cloud computing refers to the delivery of computing resources over the web rather than keeping information on your own drive or change applications for your desires, you use a service over the internet at other location to store your text or use its applications. Doing thus could give rise to sure privacy implications. In a cloud paradigm, wherever important information is placed in infrastructures of untrusted third parties, guaranteeing information confidentiality is of overriding importance. This requirement imposes clear information management choices original plain information should be accessible solely by trusty parties that don't embrace cloud providers, intermediates and Internet; in any untrusted context, information should be encrypted. Satisfying the goals has completely different levels of complexity looking on the kind of cloud service. There are many solutions for the storage as a service area, whereas guaranteeing confidentiality within the information as a service (DBaaS) is still open research area remains associate open analysis space. During this context, we propose SecureDBaaS because the resolution that permits cloud tenants to take full advantage of DBaaS qualities such as availability, security, reliability, responsibility and elasticity, measurability, without exposing unencrypted information to the cloud provider. The design method was motivated by a three goal to permit different, freelance, and regionally distributed consumer to execute simultaneous operations on encrypted information, as well as SQL statements that modify the database structure; to preserve information confidentiality and consistency at the consumer and cloud level, to eliminate any intermediate proxy between the cloud consumer and the cloud provider. The chance of blending accessibility, security and quality of a typical cloud DBaaS with information confidentiality is water tight through a example of SecureDBaaS that supports the execution of synchronic and freelance operations to the remote encrypted information from many geographically distributed consumers as in any unencrypted DBaaS setup to realize these goals. SecureDBaaS integrates existing science schemes, isolation methods, and novel strategies for management of encrypted knowledge on the untrusted cloud information.

This paper contains a theoretical discussion regarding solutions for knowledge consistency issues due to synchronic and freelance consumer accesses to encrypted information throughout this context, we have a tendency to cannot apply fully homomorphic schemes tributable to their excessive procedure quality. The SecureDBaaS style is ready-made to cloud platforms and does not introduce any intermediate proxy or broker server between the buyer and cojointly the cloud provider. Eliminating any positive intermediate server permits SecureDBaaS to realize a similar accessibility, dependability, and snap levels of a cloud DBaaS alternative proposals supported intermediate proxy were thought of infeasible for a cloud-based answer as a results of any proxy represents one purpose of failure and a system bottleneck that limits the foremost edges (e.g., measurability, accessibility, and elasticity) of a info service deployed on a cloud platform in contrast to SecureDBaaS, architectures relying on a positive intermediate proxy do not support the foremost typical cloud scenario wherever geographically unfold consumers can at a similar time issue read/write operations and organization modifications to a cloud information. A large set of experiments supported real cloud platforms demonstrate that SecureDBaaS is instantly applicable to any package as a results of it wants no modification to the cloud information services totally different studies where the planned design is subject to the TPC-C commonplace benchmark for various numbers of purchasers and

network latencies show that the performance of coinciding scan and write operations not modifying the SecureDBaaS data structure is love that of unencrypted cloud information. Workloads similarly as modifications to the data structure square measure supported by SecureDBaaS, however at the worth of overheads that seem acceptable to achieve the specified level of data confidentiality. The motivation of those results is that network latencies, that square measure typical of cloud eventualities, tend to mask the performance costs of knowledge secret writing on latency. The final conclusions of this paper square measure is very important as a result of for the first time they demonstrate the relevancy of secret writing to cloud information services in terms of feasibleness and performance.

2. LITERATURE SURVEY

L.Ferretti proposed the architecture that avoids any intermediary component, thus achieving availability and scalability comparable to that of unencrypted cloud database services2. The advantages are Guarantees data consistency in scenarios in which independent clients concurrently execute SQL queries, and the structure of the database can be modified.Reduced isolation levels for multi-version systems have never been characterized before despite being implemented in several products and its drawbacks are Concurrent modifications of the database structure are supported but at the price of higher overhead and stricter transaction isolation levels.In this paper, we presents CryptDB as the intermediate server between the client and server to provide confidentiality for application that uses DBMSs.CryptDB's approach is to execute queries over encrypted data and the key that SQL uses a well-defined set as operators3.The advantages are CryptDB prevents the DBA from learning private data.CryptDB ensures the confidentiality of logged out users data.Their drawbacks are Throughput penalty occurs while tracing the database using MYSQL servers is seems to be modest or unassumable.Include intermediate server.Single point failure and bottleneck problem. This paper overcomes the problem between a client and the server while processing the clients query request. In this paper we suggests, using multiple service providers in order to store data. This process may uses the decomposition algorithm in which the columns of a database can be split across the server. This algorithm should satisfy the following:

- Privacy constraints should not be violated
- Workload should be reduced

His advantage is to reduce heavy network traffic and the encryption and decryption cost. Needs less memory space4, drawbacks: hardware requirements are high when compared to the past solutions. In this paper, depot clients do not have to trust, that is assume, that depot servers operate correctly. Depot is built on three key ideas:

- Reduce misbehaviour to concurrency
- Enforce Fork-Join-Casual consistency
- Layer other storage properties over FJC

The merits are tolerate the fault, good availability and latency overhead5, it does not consider confidentiality and dynamic access controller. Server compromise to encrypt sensitive data which are run on client side only, not on server side. It is not trust worthy. In this paper, we challenge two benefits 1. Operational Transformation-Framework for executing lock-free concurrent operations. 2. Fork Consistency- Interacting with an untrusted server6. To Preserve Consistency and to provide check point mechanism how clients can detect and recover from malicious forks. Drawback of this paper is Single operations on a single shared document. SPORC is not designed for the environment is designed with a large number of updates. Do not permit large garbage collection frequently.Do not support computation on encrypted data.

A homomorphic public key encryption scheme (E) has four algorithms KeyGen, EncryptE, DecryptE, and an additional algorithm EvaluateE that takes as input the public key pk, a circuit C from a permitted set CE of circuits, and a tuple of cipher texts it outputs a cipher text. To achieve self-sustaining process and its demerits are excessive computational complexity7. In this paper, we show that homomorphic encryption scheme is insecure by invent a cipher text only attacks. A homomorphic encryption function allows manipulation of two or more cipher text to produce a new cipher text corresponding to some arithmetic function of the two respective plaintext, without having any information about the plaintext or the encryption/ decryption keys. Optimization and generalization that extend subset of SQL. Using proxy server represents a single point of failure and bottleneck problem13. The disadvantages are include intermediate server, Concurrent issue or read/write operations and a data structure modifications to a cloud database. In this paper, we will adapt a prefix preserving encryption scheme to create the index. We mainly discuss interval matching or exact matching as query conditions. Interval matching is defined as Boolean function f [a, b] (x), which returns true if and only if x€[a, b]. The merits are Optimization and generalization that extend subset of SQL. The content of B+ tree is not visible to an untrusted database service provider11. Drawbacks are Cloud service provider can compromise and view the data. Concurrent issue or read/write operations and a data structure modifications to a cloud database. Lot of data processing has to occur on client machines. SUNDR is a network file system designed to store secure data on untrusted servers. SUNDR protect all the file systems through cryptographically so that clients can detect any unauthorized attempts to change files. Even data can be damaged the data can recover the file systems data from untrusted clients file caches. Do not support computation in encrypt data. User cryptography is to provide privacy and integrity10. Hash-based method for database encryption is proposed Indexing information attached to the encrypted database which can be used by the server to select the data from the database. The advantages are Server can select a data to be returned in response to a query without the need of disclosing the database

content9. The drawbacks are it is only suitable for selected queries. In this paper, a new approach for data management in which a third party service provider hosts "Database as a Service" providing its mechanisms to create, store, and access their databases. Three challenges are Data privacy, Performance, and User Interface to improve performance of the query processing. Demerits: Privacy is weak. Whenever a data item that belongs to a block is required, the trusted proxy needs to retrieve the whole block, to decrypt it and to filter unnecessary data that belong to the same block8. In this paper, we use Database as a service as a service model provides to create, store, modify and retrieve data. There are two challenge: 1. Data Privacy, 2. Security. The index value of each remote table attribute value is the bucket number to which the corresponding Plain value. Using proxy server represents a single point of failure and bottleneck problem12. The disadvantages are concurrent issue or read/write operations and a data structure modification to a cloud database. Inefficient when multiclient access the database. Difficult to manage the bucket number and attribute value in database. In this paper, guaranteeing confidentiality in the database as a service is still an open research area. Confidential concurrent to secure DBaaS is proposed as the first solution that allows cloud tenants to take full advantage of DBaaS qualities such as availability and reliability without exposing unencrypted data to the cloud provider1. The advantages are to execute integrity concurrent operations on encrypted data to preserve data at the client and cloud level. Cannot apply fully homomorphic encryption scheme because of their excessive computational complexity. The data integration from multiple data sources has been important problem15. Here we are using third party to protect the data and improve privacy. To reduce the high computational cost due to encryption/decryption and to speed up the query processing time. It is impossible to execute this type of query with traditional cryptographic solutions. In this paper data outsourcing allow users and organizations to access external services from distributed services14. The encryption is of two layers 1. Inner Layer- provided by owner for initial protection. 2. Outer layer- provided by reflect policy modifications. The advantage is compared with a solution requiring to re-send a novel encrypted version of the resource is typically huge and arbitrarily large. The different disadvantages like more incentives utilization. Cloud servers are not providing long term service distribution. Present cloud servers are not provides strong assurance for data integrity. Original plain text information should be accessible solely by sure parties that do not embrace cloud providers, intermediates, and Internet; in any untrusted context, information should be encrypted. To satisfy these goals are completely different levels of quality looking on the kind of cloud service. There are many solutions making certain confidentiality for the storage as a service paradigm, whereas guaranteeing confidentiality within the info as a service (DBaaS) paradigm continues to associate open analysis space. We have a tendency to propose a unique design that integrates cloud information services with information confidentiality and also the chance of execution synchronal operations on encrypted information. This is often the primary resolution supporting geographically distributed consumers to attach associate encrypted cloud information and to execute synchronal and freelance operations as well as those modifying the info structure. The projected design has any advantage of eliminating intermediate proxies that limit the physical property, handiness and measurability properties that are intrinsic in cloud-based solutions. Secure DBaaS provides many original options that differentiate it from previous add the sphere of security for remote information services.

3. PROPOSED SYSTEM

It guarantees data confidentiality by permitting a cloud information server to execute coinciding SQL operations over encrypted data. It provides identical accessibility, security, elasticity and measurability of the first cloud DBaaS as a result of it doesn't need any intermediate proxy. Multiple consumers, probably regionally distributed, will access the same time and severally a cloud information service. It does not need a trusty broker or a trusty server as a result of tenant data and information keep by the cloud information area unit forever encrypted.

3.1 SYSTEM ARCHITECTURE

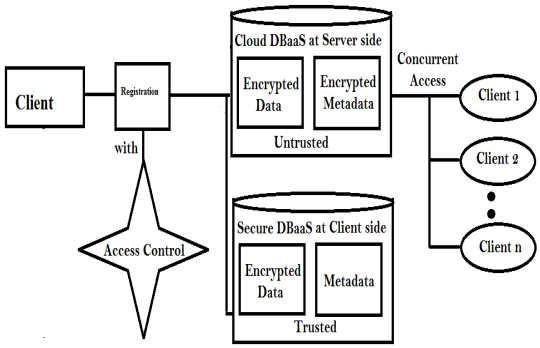


Fig 3.1 system architecture

4. RELATED WORK

Secure DBaaS distributes correct options that differentiate it from previous sector of security for remote information services are:

- Register the number of authorities needed for the number of attributes.
- Each authority is based on the condition applied.
- An access tree is constructed according to the owner's wish.
- Monotonic access structure is followed.
- The attribute set are randomly chosen and placed separately for every user.
- Here each users are distributed with different attributes generated randomly from owners attribute set.
- Users and owners attribute set are to be managed as GAL and GID
- .Both the thing should be built by authority.
- If the access control is satisfied then the data stored can be decrypted.

- The data uploaded will be stored separately at both the sides in SDBaaS and Cloud DBaaS.
- Client can have concurrent access to Cloud DBaaS only not to Secure DBaaS. So the data he has uploaded will not be compromised and would be indirectly under his own control by the help of his authority.

5. MODULE DESCRIPTION

5.1 REGISTRATION PHASE WITH ACCESS CONTROL MECHANISM

Cloud Owner and its remote user would be registered with monotonic access control mechanism. A monotonic access structure is a structure where: given a universal set P, if a subset S! Of P satisfies the access structure, all subsets S! of P which contain S! Satisfy the access structure.

5.2 SECURE DATABASE AS A SERVICE(S-DBAAS)

SecureDBaaS supports the execution of concurrent and independent operations to the remote encrypted database from many geographically distributed clients as in any unencrypted DBaaS setup. It allows cloud tenants to take full advantage of DBaaS qualities, such as availability, security, reliability, elasticity and scalability without exposing unencrypted data to the cloud provider SecureDBaaS adopts multiple cryptographic techniques and isolation mechanism to transform plain text data into encrypted tenant data and encrypted tenant data structures

5.3 MANAGEMENT OF DATA AND METADATA

Encrypted tenant data are stored through secure tables into the cloud database. To allow easily seen execution of SQL statements, each plain text data is transformed into a secure table because the cloud database is untrusted. Metadata generated by SecureDBaaS contain all the information necessary to manage SQL statements over the encrypted database in a way transparent to the user.

There are two types of such a Database metadata are related to the whole database. Table metadata contains all information that is necessary to encrypt and decrypt data of associated secure table.

5.4 CONFIDENTIAL CONCURRENT ACCESS TO DBAAS (CCAD)

5.4.1 CONCURRENT SQL OPERATIONS

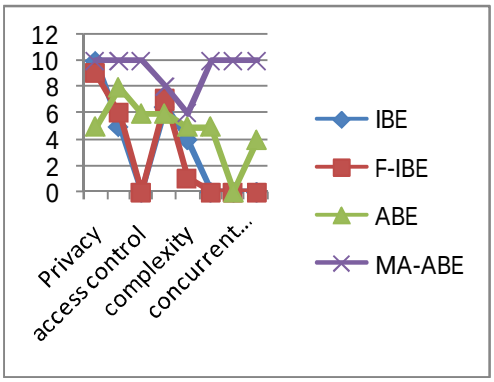
Support to the execution of SQL statements issued by multiple freelance (and presumably geographically distributed) consumers is one in every of the foremost necessary edges of SecureDBaaS with reference to progressive solutions. Our design should guarantee consistency among encrypted tenant knowledge and encrypted information as a result of corrupted or obsolete data would stop purchasers from decipherment encrypted tenant knowledge leading to permanent knowledge losses. An intensive analysis of the potential problems and solutions associated with synchronic SQL operations on encrypted tenant knowledge and data is contained in Appendix B, out there within the on-line supplemental material. Here, we have a tendency to comment the importance of characteristic two categories of statements that area unit supported by SecureDBaaS: SQL operations not inflicting modifications to the information structure, like browse, write, and update; operations involving alterations of the information structure through creation, removal and modification of information tables. Here, we've got an inclination to remark the importance of distinctive two classes of statements that unit supported by SecureDBaaS: SQL operations not inflicting modifications to the data structure, like scan, write, and update; operations involving alterations of the data structure through creation, removal, and modification of knowledge tables (data definition layer operators). In eventualities characterised by a static information structure, SecureDBaaS permits purchasers to issue synchronize SQL commands to the encrypted cloud information while not introducing any new consistency problems with relevance unencrypted databases. When information retrieval, a plain text SQL command is translated into one SQL command operative on encrypted tenant knowledge. As information does not need modification, a client can browse them once and cache them for additional uses in turn so rising performance. SecureDBaaS is that the first design that permit to synchronize and consistent accesses even once there are operations that may modify the information structure. In such cases, we've got to ensure the consistency of information through isolation levels that we tend to demonstrate will work for many victimization eventualities.

5.4.2 SEQUENTIAL SQL OPERATIONS

We describe the SQL operations in SecureDBaaS by considering associate degree initial easy situation within which we tend to assume that the cloud information is accessed by one consumer. Our goal is to focus on the most process steps therefore; we do not take into consideration performance optimizations and concurrency problems which will be mentioned but there within the on-line supplemental material. The first affiliation of the consumer with the cloud DBaaS is for authentication functions. SecureDBaaS depends on common place authentication and authorization mechanisms provided by the initial software system server. When the authentication, a user interacts with the cloud information through the SecureDBaaS consumer. SecureDBaaS analyzes the initial operation to spot that tables measure concerned and to retrieve their information from the cloud. The information are decrypted through the key and their data is employed to translate the initial plain SQL into a question that operates on the encrypted information. Instance operations contain neither plain text information (table and column names) nor plain text tenant data still, valid SQL operations that the SecureDBaaS consumer will issue to the cloud information. Translated operations are executed by the cloud information over the encrypted tenant knowledge. There is a one to one correspondence between plain text tables and encrypted tables, it's potential to stop a trust worthy information user from accessing or modifying some tenant knowledge by granting restricted privileges on some tables. User benefit is managed directly by the untrusted and encrypted cloud information. The results of the instance question that focuses encrypted tenant data and information are received by the SecureDBaaS

consumer, decrypted and delivered to the user. The quality of the interpretation method depends on the kind of SQL statement. In situations characterised by a static information structure, SecureDBaaS permits clients to issue coinciding SQL commands to the encrypted cloud information while not introducing any new consistency problems with relevance unencrypted databases. When metadata retrieval, a plain text SQL command is translated into one SQL command in operation on encrypted tenant knowledge. Data does not need modification, a consumer will browse them once and cache them for additional uses, so rising performance. SecureDBaaS is that the first design that permits coincident and consistent accesses even once there are operations which will modify the information structure. In such cases, we have got to ensure the consistency of knowledge and information through isolation levels, like the snapshot isolation, that we tend to demonstrate will work for many usage situations. Characterized by a similar secure sort we tend to limit potential consistency problems in some eventualities characterised by synchronous clients. As a example, the column share a similar secure sort. Hence reference the information, as diagrammatical by the dotted line, and use the encoding key related to their knowledge and encoding sorts. As they need a similar data and encoding sorts, will use a similar encoding key though no direct reference exists between them. The information already contain the encryption K related to the information and therefore the encoding forms of the 3 columns, as a result of the cryptography keys for all mixtures of information and encoding sorts are created within the data formatting part. Hence, K is employed because the encoding key of columns and derived in M1, M2, and M3.

7. PERFORMANCE ANALYSIS



IBE – Identity Based Encryption
F-IBE - Fuzzy Identity Based Encryption
ABE - Attribute Based Encryption
MA-ABE – Multi Authority Attribute Based Encryption

7. CONCLUSION

In this paper is proposed, Confidential Concurrent Access to DBaaS (CCAD) because the initial answer planned to that enable cloud tenants to require full advantage of DBaaS qualities like accessibility, security and reliability while not exposing unencrypted knowledge to the cloud provider. There associate any theoretical and sensible limits to increase our answer to alternative platforms to incorporate new cryptography algorithms. It additionally permits multiple, freelance and regionally distributed clients to execute concurrent operations on encrypted data, together with SQL statements that modify the information structure to preserve knowledge confidentiality and consistency at the consumer and cloud level to eliminate any intermediate server between the cloud consumer and therefore the cloud provider.

REFERENCES

[1] Luca Ferretti, Michele Colanjanni, and Micre Marchetti, "Distributed, Concurrent, and Independent Access to Encrypted Cloud Databses",IEEE Trans.Vol.25, No.2, Feb 2014.
[2] L. Ferretti, M. Colajanni, and M. Marchetti, "Supporting Securityand Consistency for Cloud Database", Proc. Fourth Int'l Symp.Cyberspace Safety and Security, Dec. 2012
[3] R.A. Popa, C.M.S. Redfield, N. Zeldovich, and H. Balakrishnan,"CryptDB: Protecting Confidentiality with Encrypted QueryProcessing," Proc. 23rd ACM Symp. Operating Systems Principles, Oct. 2011.
[4] 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 InformationSoc, Mar. 2011.
[5] 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, article 12, 2011.
[6] 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.
[7] C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattices,"Proc. 41st Ann. ACM Symp. Theory of Computing, May, 2009.
[8] H. Hacigu¨mu¨ s¸, B. Iyer, and S. Mehrotra, "Providing Database as a Service," Proc. 18th IEEE Int'l Conf. Data Eng., Feb. 2002.
[9] E. Damiani, S.D.C. Vimercati, S. Jajodia, S. Paraboschi, and P.Samarati, "Balancing Confidentiality and Efficiency in Untrusted Relational Dbmss," Proc. Tenth ACM Conf. Computer and Comm.Security, Oct. 2003.
[10] J. Li, M. Krohn, D. Mazie`res, and D. Shasha, "Secure Untrusted Data Repository (SUNDR)," Proc. Sixth USENIX Conf. OpeartingSystems Design and Implementation, Oct. 2004.
[11] J. Li and E. Omiecinski, "Efficiency and Security Trade-Off In Supporting Range Queries on Encrypted Databases," Proc. 19thAnn. IFIP WG 11.3 Working Conf. Data and Applications Security,Aug. 2005.
[12] 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.

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

[14] Fatih Emekci, Divyakant Agrawal, Amr El Abbadi. Aziz G¨ulbeden," Privacy Preserving Query Processing using Third Parties" University of California Santa Barbara,2006.

[15] Sabrina De Capitani di Vimercati, Sara oresti. Sushil Jajodia." Overencryption: Management of Access Control Evolution on Outsourced Data", ACM, 2007.