International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 13 Issue 3 –MARCH 2015.

SECURITY IN CLOUD COMPUTING

Shweta Mahajan[#1] and Sachin Mahajan[*2]

#M.E,ComputerScience,JIT, Borawah, Bhopal,M.P, India
[*]M.E.,Computer ScienceJIT, Borawah, Bhopal,M.P, India

Abstract: In cloud computing, the data will be stored in storage provided by service providers. Service providers must have a viable way to protect their clients' data, especially to prevent the data from disclosure by unauthorized insiders. Storing the data in encrypted form is a common method of information privacy protection. If a cloud system is responsible for both tasks on storage and encryption/decryption of data, the system administrators may simultaneously obtain encrypted data and decryption keys. This allows them to access information without authorization and thus poses a risk to information privacy. To Overcome this Problem the Proposed System, after establishing "Independent Encryption / Decryption Services "in cloud computing environments, users of cloud computing services will use the services of at least two cloud computing service providers, one service provider to Encryption / Decryption and other Service Provider for Storage i.e. Without the decryption Key, there is no way for the Storage service to access the users Encrypted data. Within the Encryption / Decryption Service System there is no stored user data, thus eliminating the possibility that user data might be improperly disclosed. The Core Concept is consistent with division of management authority to reduce operational risk, thus avoiding the risk of wrongful disclosure of user data.

Index Words: Access control, authentication, attribute-based signatures, attribute-based encryption, cloud storage.

1. INTRODUCTION

As the name suggests, cloud computing is an emerging computing technology which uses the internet and central remote servers to maintain data and applications accordingly. With its ability to access personal files from any computer via internet without any installation has made cloud computing a vital factor in today's business activities. This so called technology opened the doors for much more efficient computing by centralizing storage, memory, processing and bandwidth. This new methodology of cloud computing can be broken down into three major segments. Those are namely:

- Applications
- Platforms
- Infrastructure

Each of the above mentioned segments serves a different purpose according to the requirement and offers different products mainly for businesses around the world. Cloud computing describes a new supplement, consumption, and delivery model for IT services based on Internet protocols, and it typically involves provisioning of dynamically scalable and often virtualized resources It is a product and consequence of the ease-of-access to remote computing sites provided by the Internet. This may take the form of web-based tools or applications that users can access and use through a web browser as if they were programs installed locally on their own computers. Parallels to this concept can be drawn with the electricity grid, where in end-users consume power without needing to understand the component devices or infrastructure required to provide the service.

The SecureDBaaS building design is customized to cloud stages and does not present any mediator intermediary alternately representative server between the customer and the cloud supplier. Taking out any trusted middle of the road server permits SecureDBaaS to attain to the same accessibility, dependability, and flexibility levels of a cloud DBaaS. Other recommendations (e.g., [8], [9], [10], [11]) based on middle of the road server(s) were viewed as impracticable for a cloud-based arrangement on the grounds that any intermediary speaks to a solitary purpose of disappointment and a framework bottleneck that restricts the principle profits (e.g., adaptability, accessibility, and flexibility) of a database administration conveyed on a cloud stage. Not at all like

SecureDBaaS, architectures depending on a trusted middle of the road intermediary don't help the most commonplace cloud situation where topographically scattered customers can simultaneously issue read/compose operations and information structure changes to a cloud database. A substantial set of examinations focused around genuine cloud stages show that SecureDBaaS is promptly relevant to any DBMS on the grounds that it obliges no change to the cloud database administrations. Different studies where the proposed building design is liable to the TPC-C standard benchmark for distinctive quantities of customers and system latencies demonstrate that the execution of simultaneous read and compose operations not altering the SecureDBaaS databasestructure is practically identical to that of decoded cloud database. Workloads including changes to the database structure are likewise upheld by SecureDBaaS, yet at the cost of overheads that appear satisfactory to accomplish the fancied level of information privacy. The inspiration of these results is that system latencies, which are commonplace of cloud situations, have a tendency to veil the execution expenses of information encryption on reaction time. The general conclusions of this paper are imperative in light of the fact that surprisingly they exhibit the materialness of encryption to cloud database benefits as far as practicality and execution.

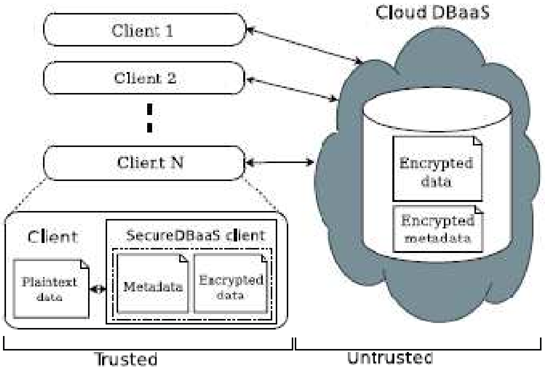


Fig. 1. SecureDBaaS architecture.

2. Related work

SecureDBaaS relates all the more nearly to works utilizing encryption to ensure information oversaw by untrusted databases. In such a case, a fundamental issue to address is that cryptographic systems can't be naı¨vely connected to standard DBaaS since DBMS can just execute SQL operations over plaintext information. A few DBMS motors offer the likelihood of encoding information at the filesystem level through the purported Transparent Information Encryption characteristic [16], [17]. This gimmick makes it conceivable to construct a trusted DBMS over untrusted stockpiling. Then again, the DBMS is trusted and decodes information some time recently their utilization. Henceforth, this methodology is not pertinent to the DBaaS connection considered by SecureDBaas, on the grounds that we accept that the cloud supplier is untruste

A cloud is basically a huge scale circulated framework where each one bit of information is reproduced on various geographically distributed servers to attain high accessibility and elite. In this manner, we first survey the consistency models in appropriated frameworks. Ref. [10], as an issue course reading, proposed two classes of consistency models: information driven consistency also customer driven consistency. Information driven consistency model considers the inner condition of a stockpiling framework, i.e., how redesigns course through the framework and what ensures the framework can give admiration to redesigns. Then again, to a client, it truly does not make a difference whether a stockpiling framework inside contains any stale duplicates. The length of no stale information is seen from the customer's perspective, the client is fulfilled. Along these lines, customer driven consistency model focuses on what particular clients need, i.e., how the clients watch information overhauls. Their work likewise depicts diverse levels of consistency in circulated frameworks, from strict consistency to frail consistency. High consistency intimates high cost and lessened accessibility. Ref. [11] states that strict consistency is never required in practice, and is indeed considered destructive. Actually, commanded by the CAP convention [3], [4], numerous conveyed frameworks present strict consistency for high accessibility.

At that point, we survey the work on attaining diverse levels of consistency in a cloud. Ref. [12] explored the consistency properties gave by business mists and made a few helpful perceptions. Existing business mists normally confine solid consistency certifications to little datasets (Google's Megastore and Microsoft's SQL Data Services), or give just consequent consistency (Amazon's simpledb and Google's Bigtable). Ref. [13] portrayed a few answers for accomplish distinctive levels of consistency while sending database applications on Amanzon S3. In Ref. [14], the consistency necessities fluctuate about whether relying upon genuine accessibility of the information, and the creators give strategies that make the framework powerfully adjust to the consistency level by checking the condition of the information. Ref. [15] proposed a novel consistency show that permits it to naturally conform the consistency levels for distinctive semantic information. At last, we survey the work on checking the levels of consistency gave by the Csps from the clients' purpose of view. Existing arrangements can be arranged into follow based checks [7], [9] and benchmark-based confirmations [16]– [19]. Follow built checks concentrate in light of three consistency semantics: security, consistency, and atomicity, which are proposed by Lamport [20], and reached out by Aiyer et al. [21]. A register is sheltered if a read that is not simultaneous with any compose returns the estimation of the latest compose, and a read that is simultaneous with a compose can give back any quality. A register is standard if a read that is not simultaneous with any compose furnishes a proportional payback of the latest compose, and a read that is simultaneous with a compose returns either the estimation of the latest compose, or the estimation of the simultaneous compose. A register is nuclear if each perused gives back where its due of the latest compose. Misra [22] is the first and foremost to present a calculation for checking whether the follow on a read/compose register is nuclear. Taking after his work, Ref. [7] proposed logged off calculations for checking whether a key-esteem capacity framework has security, normality, and atomicity properties by developing a steered diagram. Ref. [9] proposed an online check calculation by utilizing the GK calculation [23], and utilized diverse measurements to evaluate the seriousness of infringement.

The fundamental shortcoming of the current follow based confirmations is that a worldwide clock is needed among all clients. Our answer has a place with follow based confirmations. Nonetheless, we concentrate on diverse consistency semantics in business cloud frameworks, where an inexactly synchronized clock is suitable for our answer. Benchmark-built confirmations concentrate in light of benchmarking staleness in a stockpiling framework. Both [16] and [17] assessed consistency in Amazon's S3, however indicated diverse results. Ref. [16]

utilized one and only client to peruse information in the trials, and demonstrated that few inconsistencies exist in S3. Ref. [17] utilized different topographically dispersed clients to peruse information, and discovered that S3 often abuses monotonic-read consistency. The consequences of [17] legitimize our two-level reviewing structure. Ref. [18] presents a customer driven benchmarking procedure for comprehension inevitable consistency in dispersed keyvalue capacity frameworks. Ref. [19] surveyed

3ARCHITECTURE DESIGN

SecureDBaaS is intended to permit various and autonomous customers to interface specifically to the untrusted cloud DBaaS with no halfway server. Fig. 1 depicts the general construction modeling. We accept that an occupant association gains a cloud database administration from an untrusted DBaaS supplier. The inhabitant then sends one or more machines (Customer 1 through N) and introduces a SecureDBaaS customer on each of them. This customer permits a client to interface with the cloud DBaaS to control it, to peruse and compose information, and indeed to make and alter the database tables after creation.

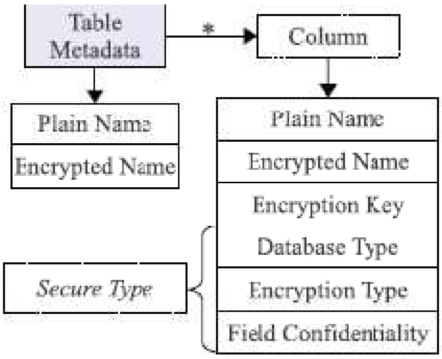


Fig. 2. Structure of table metadata.

We accept the same security demonstrate that is normally embraced by the writing in this field (e.g., [8], [9]), where inhabitant clients are believed, the system is untrusted, and the cloud supplier is fair however inquisitive, that is, cloud administration operations are executed effectively, yet occupant data classifiedness is at danger. Hence, occupant information, information structures, and metadata must be encoded before leaving from the customer. An intensive presentation of the security model received in this paper is in Appendix An, accessible in the online supplemental material.

The data oversaw by SecureDBaaS incorporates plaintext information, encoded information, metadata, and scrambled metadata. Plaintext information comprise of data that an occupant needs to store and process remotely in the cloud DBaaS. To keep an untrusted cloud supplier from abusing classifiedness of inhabitant information put away in plain structure, SecureDBaaS receives various cryptographic procedures to change plaintext information into encoded inhabitant information and scrambled occupant information structures in light of the fact that even the names of the tables and of their segments must be scrambled. SecureDBaaS customers deliver additionally a set of metadata comprising of data needed to encode and unscramble information and also other organization data. Indeed metadata are encoded furthermore put away in the cloud DBaaS.

SecureDBaaS moves far from existing architectures that store only occupant information in the cloud database, and recovery metadata in the customer machine [9] or part metadata between the cloud database and a trusted intermediary [8]. At the point when considering situations where various customers can get to the same database simultaneously, these past arrangements are very wasteful. For instance, sparing metadata on the customers would require cumbersome systems for metadata synchronization, and the useful incomprehensibility of permitting various customers to get to cloud database benefits freely.

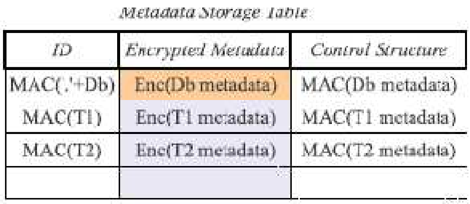
Arrangements focused around a trusted intermediary are more doable, yet they present a framework bottleneck that decreases accessibility, versatility, and adaptability of cloud database administrations. SecureDBaaS proposes an alternate methodology where all information and metadata are put away in the cloud database.

SecureDBaaS customers can recover the fundamental metadata from the untrusted database through SQL articulations, so that various occurrences of the SecureDBaaS customer can get to to the untrusted cloud database autonomously with the assurance of the same accessibility and adaptability properties of average cloud DBaaS. Encryption techniques for inhabitant information and creative answers for metadata administration also capacity are portrayed in the accompanying two segments.

SecureDBaaS stores metadata in the metadata stockpiling table that is spotted in the untrusted cloud as the database. This is an unique decision that increases adaptability, however opens two novel issues regarding productive information recovery and information secrecy. To permit SecureDBaaS customers to control metadata through SQL articulations, we spare database and table metadata in a plain structure. Indeed metadata classifiedness is ensured through encryption.

Database and table metadata are scrambled through the same encryption key before being spared. This encryption key is known as an expert key. Just trusted customers that as of now know the expert key can decode the metadata and gain data that is important to encode and unscramble inhabitant information. Every metadata can be recovered by customers through an related ID, which is the essential key of the metadata capacity table. This ID is processed by applying a Message Confirmation Code (MAC) capacity to the name of the object (database or table) depicted by the relating column. The utilization of a deterministic

MAC capacity permits customers to recover the metadata of a given table by knowing its plaintext name.



Metadata Storage Table

ID	Encrypted Metadata	Control Structure
MAC('.'+Db)	Enc(Db metadata)	MAC(Db metadata)
MAC(T1)	Enc(T1 metadata)	MAC(T1 metadata)
MAC(T2)	Enc(T2 metadata)	MAC(T2 metadata)

Fig. 3. Organization of database metadata and table metadata in the metadata storage table.

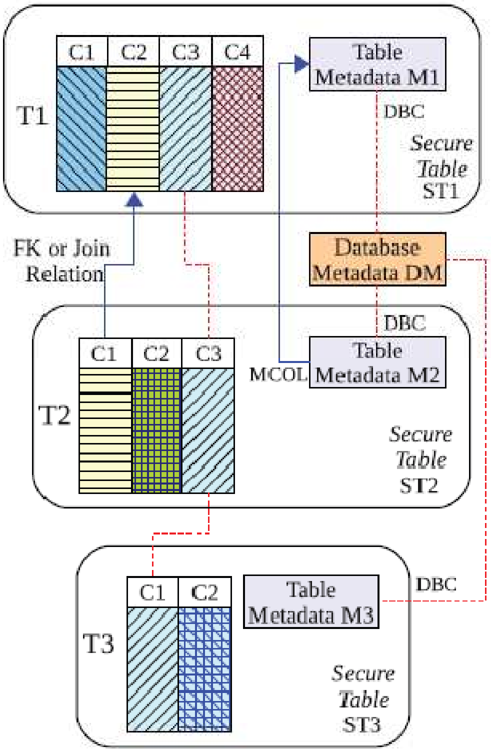
4 Operations

We portray how to introduce a SecureDBaaS structural planning from a cloud database administration gained by an inhabitant from a cloud supplier. We accept that the DBA makes the metadata stockpiling table that toward the starting contains just the database metadata, and not the table metadata. The DBA populates the database metadata through the SecureDBaaS customer by utilizing arbitrarily produced encryption keys for any blends of information sorts and encryption sorts, and stores them in the metadata stockpiling table after encryption through the expert key. At that point, the DBA conveys the expert key to the genuine clients. Client access control strategies are administrated by the DBA through some standard information control dialect as in any decoded database.

In the accompanying steps, the DBA makes the tables of the encoded database. It must consider the three field privacy traits (COL, MCOL, and DBC) presented toward the end of the Section 3. Given us a chance to portray this stage by alluding to a basic yet illustrative case indicated in Fig. 4, where we have three safe tables named ST1, ST2, and ST3. Each one table STi (i ¼ 1; 2; 3) incorporates an encoded table Ti that contains scrambled inhabitant information, and a table metadata Mi. (Albeit, as a general rule, the names of the segments of the protected tables are haphazardly created; for the purpose of straightforwardness, this figure alludes to them through C1-CN.

Sequential SQL Operations

We portray the SQL operations in SecureDBaaS by considering a beginning straightforward situation in which we accept that the cloud database is gotten to by one customer. Our objective here is to highlight the principle preparing steps; consequently, we don't consider execution advancements what's more concurrency, accessible in the online supplemental material. The principal association of the customer with the cloud DBaaS is for confirmation purposes. SecureDBaaS depends on standard confirmation and approval systems gave by the first DBMS server. After the confirmation, a client connects with the cloud database through the SecureDBaaS customer. SecureDBaaS dissects the first operation to recognize which tables are included and to recover their metadata from the cloud database. The metadata are unscrambled through the expert key and their data is utilized to interpret the first plain SQL into a question that works on the encoded database.



ig. 4. Management of the encryption keys according to the field onfidentiality parameter.

3. Concurrent SQL Operations:

The backing to simultaneous execution of SQL articulations issued by various free (and potentially geologically appropriated) customers is a standout amongst the most imperative advantages of SecureDBaaS concerning best in class arrangements. Our building design must surety consistency among encoded inhabitant information and encoded metadata in light of the fact that undermined or out-of-date metadata would counteract customers from deciphering encoded occupant information bringing about lasting information misfortunes. An intensive examination of the conceivable issues and arrangements identified with simultaneous SQL operations on scrambled occupant information and metadata is contained, accessible in the online supplemental material. Here, we comment the essentialness of recognizing two classes of proclamations that are backed by SecureDBaaS: SQL operations not bringing on changes to the database structure, for example, read, compose, and overhaul; operations including modifications of the database structure through creation, evacuation, and change of database tables (information definition layer operators).a

CONCLUSION

In this paper, We propose an imaginative structural planning that ensures privacy of information put away in broad daylight cloud databases. Dissimilar to best in class approaches, our answer does not depend on a transitional intermediary that we consider a solitary purpose of disappointment and a bottleneck constraining accessibility and versatility of regular cloud database administrations. An expansive part

of the examination incorporates answers for backing simultaneous SQL operations (counting explanations adjusting the database structure) on encoded information issued by heterogeneous furthermore conceivably geologically scattered customers. The proposed structural planning does not oblige alterations to the cloud database, and it is quickly relevant to existing cloud DBaaS

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.