Integrity Checking Mechanism with Effective and Realistic Approach in Distributed Database

Pallavi Saxena, Nitin Dubey, Sonali Pathak

Lovely Professional University, Department Computer Science Phagwara, Punjab, India

Abstract- In Constructing of a successful distributed database we require to introduce the strong integrity checking mechanism as less adhering may possibly breach security aspects. The mechanism should represent the semantic properties of stored data. Verifying the integrity of data from various outsourced safety is a challenge for database researchers, this paper proposes a specific approach to bring pureness of data by employ checking method to substantiating send back query result in distributed database. To accomplish our goals, a unique property is dynamically inserted into each tuples called faux attribute as well as this method also detects the unauthorized alteration on data.

Index Terms- faux attribute; verification; validation; trigger manager

I. Introduction

The race between confidentiality, integrity and availability has put terrible pressure for the need of out-of-box thinking. The increase in cruciality of information to the business has inflated the challenges in protecting and managing the data. The bulk of data that business must manage has driven planning to make data available on 59's rule that is 99.999%.The goal of business continuity is to assure information security. The main aspects are to increase the accessibility of information without compromising of security measures. Distributing data is one of the ways to provide business continuity. Distributed database is a aggregation of different databases that are logically distributed and stored in a variety of outsourced database system that aspects client as a centralized application where each database application may involve different database technology and different computer architecture that contribute the execution of global query. With the more promotion of internet technologies, the major problem is to preserve the integrity of data. Integrity is the unity of stored data. In decentralized environment, the more advanced view of Integrity as shown in Fig.1 is degree of correctness, completeness, freshness, timeliness, efficient, accuracy, entireness and identity of data set. Infects, there are more research on data confidentiality and integrity on centralized system but very little have been focused on integrity checking in distributed system keeping in mind the cloud environment that have brood of storage array's contain multiple fragment of data .Based on researches on traditional integrity maintenance techniques and methods, this paper design Real time integrity checking through Verifying and Validate Query result by accommodate the trusted third party in distributed system This is set up by dynamically inserting a unique attribute on every tuples of each fragment by using trigger called faux attribute.

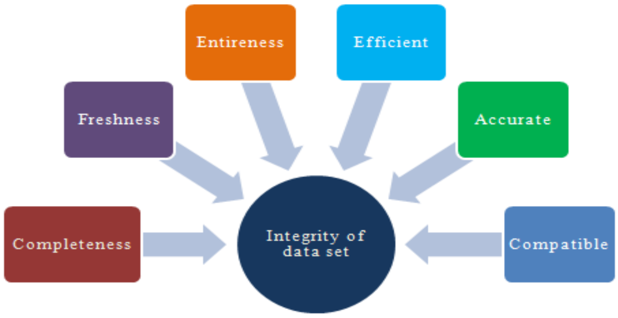


Figure1: Advanced view of integrity

.

Structure of the paper are as follow. In section 2, we briefly discuss the previous work on integrity and next section contain method and performance countermeasure and atlast we summarize our solution and outline the future work.

II. RELATED WORK

A Glimpse of pre-implemented techniques for integrity in distributed database

As with the increase in requirement of distributed data, different threats of database like Excessive Privilege Abuse, Legitimate Privilege Abuse, Privilege Elevation, Database Platform abuse, SQL Injection, Weak Audit Trail, Weak authentication, Backup Data Exposure, weak audit trail and Denial of service sneak their way inside which greatly breach the integrity of decentralized system. From the analysis as illustrated in fig. 2, it may be shown that the excessive privilege abuse and sql injection greatly affects the integrity of the global schema. By keeping in mind of these kinds of threats, the integrity can be categorized into different group.

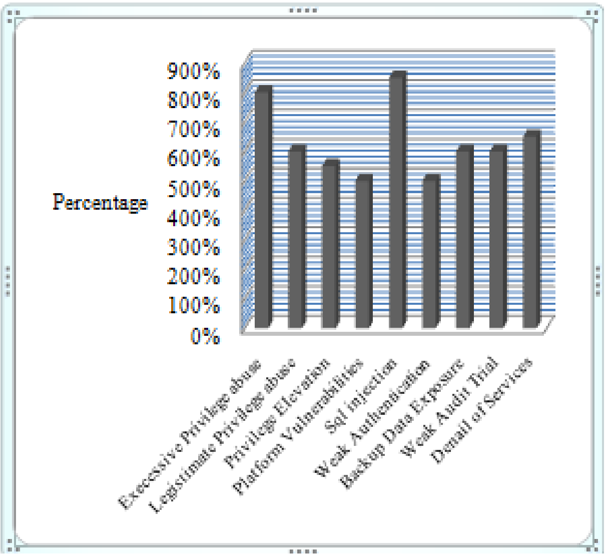


Figure 2: Effect of database treats on distributed system

From the last 5 year, there have been many research fruits on these vulnerabilities. Some discussion on the multilevel using security policy or tag to achieve integrity, as in [5], author upholds a integrity of data in term of completeness and freshness by using fake attribute, as in [2], but this fake attribute is generated manually by owner not dynamic, moreover in distributed environment because for every fragment to insert a fake attribute manually is very time consuming also privacy of fake attribute is compromised, when malicious user will able to break through the fake attribute. Some research focus on intrusion-tolerant mechanism to maintain integrity and confidentiality in distributed environment, as in [9], the approach like strategies of the control, test of the integrity, the test of efficiency and legality of data from client to assure the integrity of data with the technology of tighten, defaulting, ruling, in database, as in [6], and author use the concept of traditional hashing and encryption techniques by assembling a hash function through the application of grouping of encryption algorithm to simultaneously provides data confidentiality and integrity during storing and operating to protect illegally stolen and distorted in database application. This database assurance can be obtained from the encryption, as in [5], However from the encryption the main drawback is to big amount of key and more complex management of key. Adopting one-way hash function signature ensure the client as tuples receive by them are not tampered or destroy by malicious transaction In hashing some serious drawback is before digest the database tuples must be sorted and cannot support dynamic update properly, update will generate a new value that is time consuming work. Moreover Table I shown that there will be more number of collision attacks in different hashing algorithm.

Table 1: List of Attacks on Hash Algorithm

Algorithm	Designers	Digest size	Attack	year	Scouter
MD2	Ronald Rivest in 1989	128 bits	Finding a preimage attack with time complexity equivalent to 2^{104} applications of the compaction function.	2004	Muller
MD5	Ronald Rivest in 1992	128 bits	Vulnerable to a primate attack against MD5 This attack is only theoretical, with a time complexity of $2^{123.4}$.	2009	Toe Xie and denggu o feeng
SHA-0	National Security Agency	160 bits	Determine the collision attack with time complexity 2^{51}	2004	Joux, Carribault, Lemuet, and Jalby
SHA-1	National Security Agency	160 bits	The attacks can find collisions in the full version of SHA-1, requiring time complexity smaller than 2^{69}operations.	2005	Professor Wang

Concepts

In this paper ,the main concepts behind this is the owner store the encrypted database that contain faux attribute on the service provider and meanwhile for the verification and validation of data transmit the encrypted total sum value of faux attribute to the trusted third party. The encrypted value in trusted third party should be able to protect and secure reasonable data and should be in accordance with the data in service provider.

III. METHODOLOGY

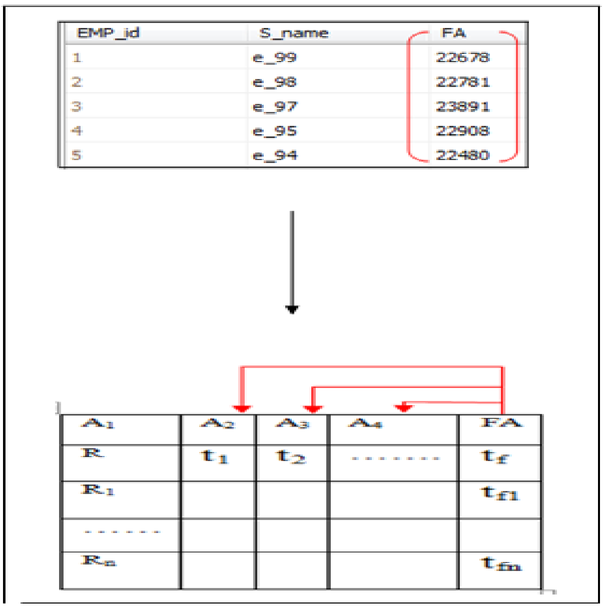
A. Preliminaries

The sensitivity of metadata for distributed storage. At present most enterprise network**,** use distributed system to store files in different nodes, we need to store Data definition call catalog files in specialized server that facilitates the management, access and retrieval of massive fragment of schema. catalog files are used for the translation, optimization and execution of transaction and their statistics is also updated when data definition changed with respect to local autonomy of each site. The catalog consists of a global schema module includes the global relation's attributes, Fragments distribution modules includes the attributes belong to each fragment, qualification of fragment ,fragment mapping or allocation module consists of binding the name of physical images to the local data stored at each site in related to access matrix module includes accessibility of fragments with respect to remote and site autonomy, statistics module consists of profile of each of the fragments, and state management module comprise lock and recovery information. Catalog management system located in corresponding with different fragment to ensure the files access efficiency for users. We don't need to access all sites for finding location of data just focus on catalog for fast retrieval of data which enhances the performance and scalability of Distributed System.

Faux attribute. The idea of the approach is to consider faux attribute which is the basic protection unit in distributed system. We use trigger to generate faux attribute, as trigger is an stored procedure which is automatically fired when insertion, deletion and updating happen in database. When insertion of data goes on, trigger manager automatically inserts unique faux attribute on every tuples into the schema.

Definition. For the purpose of convenience, **"**FA**"** in this paper means faux attribute, and refers to the faux tuple of Row N and column M **"**RPN**"** stands for random parity number. S(FA) means the process of adding parity number to faux attribute. TS(FA) stands for the process of adding faux attribute and generation total sum of it. E(TS((FA)) means the process of encrypting total sum by key **"**K**"**.

Techniques of generating Faux attribute. Let us suppose a tuple $R<t_1,t_2,t_3..t_n>$ where t_1 is an attribute of the R_0 and t_1 is of any data type to fulfill our mechanism we add a faux attribute call t_f ,now the tuple become $R_f<t_1,t_2,t_3…t_n.,t_f>$. The generated faux attribute will correlate with every tuples, Fig.3 shows the structure. Irrespective of this random parity number is added to the faux attribute it is same as in case of CRC in networking to check whether the value of faux attribute receive by user is same as send by providers, this improve the privacy of faux attribute then calculate the total sum of all faux attribute after that encrypt sum from symmetric key with algorithm triple DES.



EMP_id	S_name	FA
1	e_99	22678
2	e_98	22781
3	e_97	23891
4	e_95	22908
5	e_94	22480

Figure 3: Structure of faux attribute

B. Verification.

We use five participants in our approach, Fig. 4 show the procedure

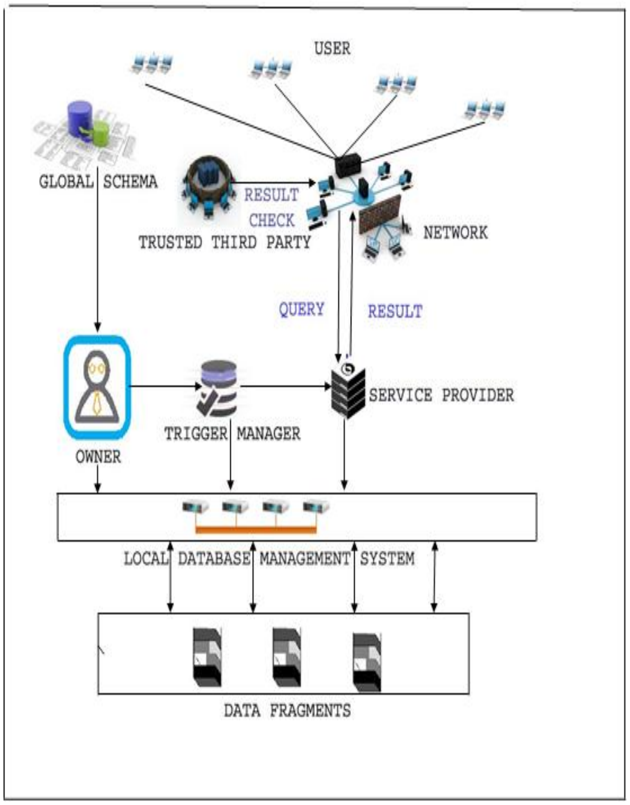


Figure 4: Structure of verification mechanism

Trigger manager. Here trigger manager has responsibility to manage special kind of catalog which stores the information related to different trigger for different fragments. Whenever new tuples insert into fragment, trigger manager automatically fires trigger for that fragments that will insert faux attribute in respect of that tuple.

Owner. Owner refers to the current authorization identifier which creates the database. Owner firstly make some operation, they breakup the whole global schema into logical units called fragments meanwhile with the help of trigger manager generate faux attribute by using (1) and insert it into the fragments of each database and then allocate those fragments into service provider.

$$S(FA) = ((t_{f1})+t(RPN))+(t_{f2}+t(RPN))+\ldots(t_{fn}+t(RPN)) \qquad (1)$$

$$T(S(FA)) = FA_1+FA_2+FA_3+\ldots FA_n \qquad (2)$$

$$T(S(FA)) = E(TS((FA)) \qquad (3)$$

Service Provider. After inserting faux attribute, owner resides the encrypted database on the premises of provider. It is an organization that provides access to database. It processes the user query and facilitates verification and validation of data.

TTP. Trusted Third Party is a constitution that assists the real time integrity checking of data by different users. The owner send the encrypted total sum value E(TS((FA)) of faux attribute to the TTP. When user get result from service provider at the same time will receive the E(T(FA)). TTP ease interaction between service provider and user who both have trust to TTP.

User. When one authenticate user desire to access the database, he will fetch query to the service provider, user receive the predicted result that hold the integrity checking module TS((FA)) generated by the service provider by Using(2), In the meantime user gets the E(T(FA))from the TTP, decrypt it. The same value ensures the verification of result and implies no alteration take place. Since faux attribute relates with every tuples as suppose tuples <t_1,t_2,t_3..t_n>will contain data {x,y,z} then faux attribute will be t_{f1} = {(xyz)+RPN},.If any unauthorized modification will happen and tuple changed to {r,y,z}, then faux attribute value clearly interpolate the inconsistency by matching it with each tuples and detect where violation of integrity occur. Furthermore, user recalculates the value of faux attribute by deducting it with RPN to assure that faux attribute is not manipulated during transmission by malicious transaction.

IV. PERFORMANCE COUNTERMEASURE

Compared with the traditional method for integrity checking as hash function like MD5 and CRC a more compact hash (< 128 bits) are convenience and do attention about performance but very prone to security attacks while SHA-256 or SHA-512 where security is predominant but for a network system with high traffic, these hash functions can take a substantial toll on the CPU. Moreover this algorithm is more vulnerable to attack and susceptible to collision attack. The addition of faux attribute that is numeric for Simplicity will have great impact on security and performance, illustrate in Fig.5. .

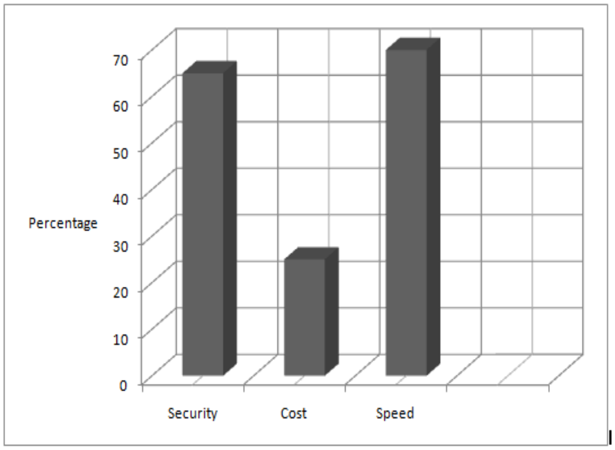


Figure 5: Performance Result

1) More secure as for every row there will be unique faux attribute it is difficult for attacker to guess which attribute is fake, Moreover addition of RPN which is only known to user and service provider increase the confidentiality of faux attribute as it ensure that authenticate users will get the result.
2) Save storage space as only one column is added to the original table this reduce the cost of transmission, therefore increase performance.

3) Sustain integrity between data as Faux attribute is related with every tuples, it will be easy to find where violation can take place by examine the faux attribute.
4) Although, no manually work is perform by owner for insertion, updating and deletion of faux attribute. Trigger manager will automatically insert and update faux attribute when require in each fragment of table, improve the speed of system.

V. CONCLUSION

The main motive of the paper is to check the unity of important and sensible data system by infix faux attribute in each fragment of distributed system. The faux attribute bring real time integrity check value which can be used to provide verification and validation of data. This method also provides trust to service provider by verifying result of send back query with TTP. Moreover, the solution requires less consuming power as compared with other solutions. Uniqueness of faux attribute will be bottleneck, but with effectual use of advanced trigger can impart to the problem solving approach

ACKNOWLEDGMENT

First of all we would like to thank IJSRP for providing us with the platform to present our work to people and for giving us opportunity to get our research completed. We thank Lovely professional University for its support throughout the research work. This project was supported by the University efficiently, so we thank everyone involved throughout the work. We would like to thank the university for providing us with tools required to complete the research work

REFERENCES

[1] Bertino, E, Sandhu, R, " Database Security –concepts and challenges," on IEEE Transaction on Dependable and Secure Computing, vol.2 (1), 2005, pp.2-19.

[2] Dai Jiazhu, Zhang Yurong, Li Xin, Luo Shuangyan, "A completeness and freshness guarantee scheme for outsourced database," on Second International Conference on Networking and Distributed Computing, 2011, pp.201-205.

[3] Min Xie, Haixun Wang, Jian Yin, and Xiaofeng Meng. Integrity auditing of outsourced data. In VLDB, 2007.

[4] Li-xin Xu, Dong Sun, and Dan Liu, " study on method for data confidentiality and data integrity in relational Database," in 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT),vol(1),2010,pp.292-295.

[5] S. Jajodia and R. Sandhu, "Polyinstantiation Integrity in Multilevel Relations," in Proc IEEE Symposium on Research in Security and Privacy, 1990, pp. 104-105.

[6] S. Jajodia and R. Sandhu, "Integrity Mechanisms in Database Management System," in Proceedings of the 13th NINS-NCSC National Computer Security Conference, 1990-10, pp. 526-540.

[7] V. Daza and J. Herranz, "A Distributed and Computationally Secure Key Distribution Scheme. ISC'02. LNCS 2433.," on http://eprint.iacr.org/2002/069 , 2002.

[8] Zubi, Z.S. " On distributed database security aspects," in International Conference on Multimedia Computing and Systems, pp231-235, April 2009

[9] Gu-Ping Zheng andLu-Feng Xu, " Distributed database system security model of power enterprise based on intusion tolerance technology, " on International Conference on Power System Technology, pp1-4, October2006

[10] http://en.wikipedia.org/wiki/MD2_(cryptography)

AUTHORS

First Author – Pallavi Saxena, M.Tech, Student, Lovely Professional University,Department.Computer Science Phagwara, Punjab, India, Email: pallavisaxena021@gmail.com
Second Author – Nitin Dubey, Assistant Professor, Lovely Professional University,Department.Computer Science Phagwara, Punjab, India, Email: nitmeerut@gmail.com
Third Author – Sonali Pathak, M.Tech, Student, Lovely Professional University,Department.Computer Science Phagwara, Punjab, India, Email: Pritipathak.pathak@gmail.com