A Review on Partial Security Violation in Distributed Database

Dr. Sona Malhotra, Richa Punia
Department of Computer Science and Engineering
University Institute of Engineering and Technology
Kurukshetra University, Haryana India

Abstract—

This paper will examine the features of the distributed database architecture and learning the task of distributed database management system will lead us to a successful design. The design will improve scalability, accessibility and flexibility while accessing various types of data. Develope a successful distributed database system requires addressing the importance of security issues that may arise and possibly compromise the access control and the integrity of the system. We propose some solution for some security aspects such as multilevel access control, confidentiality and reliability that pertain to a distributed database system. The aim of a distributed database management system (DDBMS) is to process and communicate data in an efficient and cost-effective manner. It has been recognized that such distributed systems are vital for the efficient processing required in military as well as commercial applications. For many of these applications it is especially important that the DDBMS should provide partial security. For example, the DDBMS should allow users who are cleared at different security levels access to the database consisting of data at a variety of sensitivity levels without compromising security. In this paper we discuss partial security issues for a DDBMS.

Keywords: ***Distributed database system security, distributed database, distributed database management system, distributed database retrieval problems, discretionary security distributed database, query processing, multilevel security.***

I. INTRODUCTION

In Today's business environment with an increasing need for distributed database and client/server applications as the need for consistent, scalable and accessible information is progressively growing [1]. Distributed database system provide improvement in communication and data processing due to data distribution throughout the different network sites. Not only is data access faster, but its a single-point of failure is less likely to occur, and it provides local control of data to users. However, there is some of complexity when attempting to manage and control distributed database systems

II. Distributed Database System

A distributed database is a collection or gathering of databases which are distributed and the stored on multiple computers within a network. A distributed database is a set of databases stored on multiple computers that are typically appears to applications as a single database. "Consequently, an application can simultaneously access and modify the data in several databases throughout the network ". A database, link connection allows local users to access data on a remote database. In a distributed database system, the databases are stored on several computers. The computers in a distributed system communicate with one another through various communication media, such as for example. high-speed networks or telephone lines. They do not share main memory or disks. Each database may include different database management. System and different architectures that are used to provide distribution of the execution of transactions. The main objective of a distributed database management system (DDBMS) is used to control the management of a distributed database (DDB) in such a way that it appears to the user like a centralized database. For example, the DDBMS should allow users who are cleared or satisfied at different security levels access to the database consisting of data at a variety of sensitivity levels without compromising security.

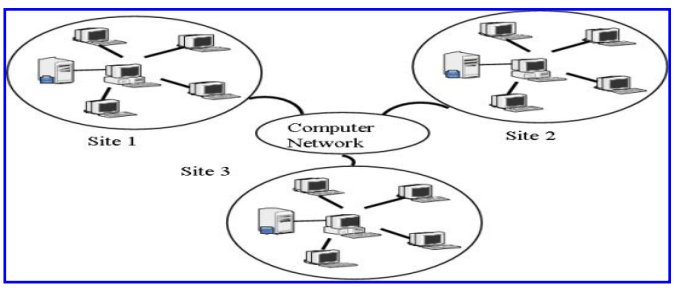

FIG 1.1(Architecture of Distributed Database System[1]

III. Characteristics of distributed database

- Data is used at one location only (other than centralized).
- Data accuracy, confidentiality, and security is a local responsibility.
- Files are simple and used by only a few applications. In this case, there is no benefit to maintaining complex centralized software. Cost of updates is too high for a centralized storage system.

Data is used locally for decision-support. Queries are against the database result in inverted lists or secondary key accesses. Such queries would decrease the performance of a centralized system. Fourth-generation languages used locally may require different data structures than the centralized systems [2].
Each database may include different database management systems and different architectures that distribute the execution of transactions. Providing the appearance of a centralized database system is one of the main aims but it is also have many objectives of a distributed database system. Such an image is accomplished by using the following transparencies: Location Transparency, Performance Transparency, Copy Transparency, Transaction Transparency, Fault Transparency, Fragment Transparency, Schema Change Transparency, and Local DBMS Transparency. These eight transparencies are believed to incorporate the desired functions of a distributed database system. Other goals of a successful distributed database include free object naming [3]. " Free object naming means that it is allow different users the ability to access the same object with different names, or different objects with the same internal name. Concurrency control is another issue among database systems. "Concurrency control is the main activity of coordinating concurrent accesses to a database in a multi-user database management system (DBMS)"[4]. There are a number of methods that provide concurrency control such as: Two phase locking, Time stamping, Multiversion timestamp, and optimistic non-locking mechanisms. Some methods are used to provide better concurrency control than others depending on the system

IV. Security Issues in Distributed Database

Database security is the system, processes, and procedures that are used to protect a database from unintended activity. Unintended activity can be categorized into authenticated misuse, malicious attacks or inadvertent mistakes made by authorized individuals or processes. Database security is also a specialty within the broader discipline of computer security [5]. Traditionally database have been protected from external connections by firewalls or routers on the network perimeter with the database environment existing on the internal network opposed to being located within a demilitarized zone. Additional network security devices that detect and alert on malicious database protocol traffic include network intrusion detection systems along with host-based intrusion detection systems. [6] Databases provide many layers and types of information security and, typically specified in the data dictionary, including:

- **Access control:** Access control is a system which enables an authority to control access to areas and resources in a given physical facility or computer-based information system.
- **Auditing:** Audit has two components: the collection and organization of audit data and an analysis of the data to discover or diagnose security violations. Audit data needs protection from modification by an intruder .
- **Authentication:** Authentication is the act of establishing or confirming something (or someone) as authentic that is the claims made by or about the subject are true
- **Encryption:** In cryptography, encryption is the process of transforming information (referred to as plaintext) using an algorithm (called cipher) to make it unreadable to anyone except those possessing special knowledge, usually referred to as a key Integrity.

V. Partial security concept

In distributed database, the partial security is used to improve the performance of system. Systems that are partially secure allow potential security violations such as covert channel use at certain situations [7]. In many distributed applications, security is another important constraint, since the system maintains perceptive information to be shared by multiple users with different levels of security clearance. As more and more of such systems are in use, one cannot avoid the need for integrating them. It is important to define the exact meaning of partial security, for security violations of sensitive data must be strictly controlled. A security violation here indicates a potential covert channel, i.e., a transaction may be affected by a transaction at a higher security level. One approach is to define security in terms of a percentage of security violations allowed. However, the value of this definition is questionable. Even though a system may allow a very low percentage of security violations, this fact alone reveals nothing about the security of individual data. For example, a system might have a 99% security level, but the 1% of insecurity might allow the most sensitive piece of data to leak out. A more precise metric would be necessary for the applications where security is a serious concern.

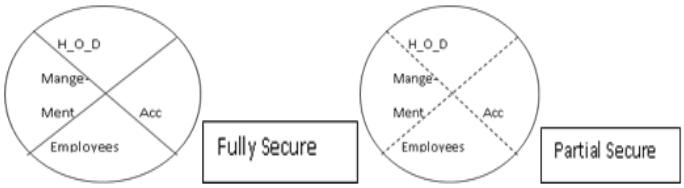

Fig. 2 (Full and Partial Security in distributed database[7])

Thus in partial security the violation is allowed in certain security levels shown in above figure. The solid line indicates that the system is fully secure whereas doted lines define that the system can violate security under certain circumstances. In distributed database, partial security is used to improve the performance of system. Systems that are partially secure allow potential security violations such as covert channel use at certain situations [3]. The basic purpose of requirement specification that allows the system designer to specify important properties of the database at a suitable level has been suggested. In many distributed applications, security is another important constraint since the system maintains perceptive information to be shared by multiple users with different levels of security clearance. As more and more of such systems are in use, one cannot avoid the need for integrating them.

VI. RELATED WORK

As distributed networks become more popular, the need for improvement in distributed database management systems becomes even more important. For the past several years the most prevalent database model has been relational. While the relational model has been particularly useful, its utility is reduced if the data does not fit into a relational table. Many organizations have data requirements that are more complex than can be handled with these data types. Multimedia data, graphics, and photographs are examples of these complex data types. Many scholars do various different researchers in the field of distributed databases system (DDBMS) and related to security of the same. Some does work to propose new protocols to impose security restrictions, some veterans do research for the partial security constraints to save at least some part of the DDBMS systems. In Sang H. Son [1] presented a Conflicts in database systems with both real-time and security requirements can sometimes be irresolvable. They attack this problem by allowing a database to have partial security in order to improve on real-time performance when necessary. By their definition, systems that are partially secure allow security violations between only certain levels. In [2] Moses Aruba et al presentsted a multilevel secure database management system (MLS/DBMS) products no longer enjoy direct commercial-off-the-shelf (COTS) support. The prototype we implemented was used to instrument a series of experiments to determine the relative performance of the tuple , attribute, and element level fragmentation schemes. Their experiments measured the impact on the front-end and the network when various properties of each scheme, such as the number of tuples , attributes, security levels, and the page size, were varied for a Selection and Join query. They were particularly interested in the relationship between performance degradation and changes in the quantity of these properties. The performance of each scheme was measured in terms of its response time. The response times for the element level fragmentation scheme increased as the numbers of tuples , attributes, security levels, and the page size were increased, more significantly so than when the number of tuples and attributes were increased. The response times for the attribute level fragmentation scheme were the fastest, suggesting that the performance of the attribute level scheme is superior to the tuple and element level fragmentation schemes. fragmentation scheme exhibited the worst performance degradation compared to the and attribute level schemes. In [3]Steven P. Coyet al Security concerns must be addressed when developing a distributed database. When choosing between the object oriented model and the relational model, many factors should be considered. The most important of these factors are single level and multilevel access controls, protection against inference, and maintenance of integrity. When determining which distributed database model will be more secure for a particular application, the decision should not be made purely on the basis of available security features In [5]Bhavani Thuraisingham et al presents the security level of the schema of a relation is the security level of the user who creates the schema.

VII. CONCLUSION

At last we can say that with this proposed solution allows the security violation with security aspect of system performance. This means if particular job or transaction is blocked due to causes of network errors, in that situation it allows to access the data at its higher level with all security concerns defined at their corresponding levels. This will increase system performance by improving the response time and transaction access time.

REFRENCES

[1] Sang H. Son and Craig Chaney "Supporting the requirements for multilevel secure and real-time database in distribute environments", pp.136—147(1997).
[2] Moses Garuba-"Performance study of a COTS Distributed DBMS adapted for multilevel security" Consultant scientist, U.S. Government, Washington DC, (2004).
[3] Steven P. Coy-"Security Implications of the Choice of Distributed Database Management System Model: Relational vs. Object-Oriented". heru24.blogspot.com/.../tugas-sistem-basisdata.html(2010) .
[4] Ghazi Alkhatib-"Transaction Management in Distributed Database Systems: the Case of Oracle's Two-Phase Commit" Journal of Information Systems Education, Summer (2002).
[5] Bhavani Thuraisingham -"Multilevel Security Issues in Distributed Database Management Systems II"-Computers & Security, 10 (1991) 727-747(2007).
[6] Charles P. Pfleeger, Shari Lawrence Pfleeger **"**Security in Computing", www.studytemple.com/.../2830-security-computing-charles-p-pfleeger,4th Edition (2008).
[7] Millen /Lunt, A.Tamaru, F.Gilham, R.Jagannathan, C.Jalali, P.Neumann and H.Javitz, "Security for Object-Oriented Database "**,**IT-security and privacy-design and use of privacy"-enhancing (1992)

[8] Davidson, M.A. "*Security in an Oracle data base environment*". Information Systems Security (2007).
[9] Dutta, A. and McCrohan, K. (2002). Management's Role in Information Security in a Cyber Economy. California Management Review (2007) .
[10] Luftman, J., Managing the Information Technology Resource: Leadership in the Information Age. Upper Saddle River, NJ. Pearson Education, Inc. (2004).
[11] Newman, "*A. Database Security Best Practices Security*". Retrieved April 1, 2007 from Business Source Premier database. Palmquist, M., Busch, C., De Maret, P., Flynn, T., Kellum, R., Le, S., Meyers, B., (2005).
[12] Thuraisingham, B. "Database and Applications Security: Integrating Information Security and Data Management". Boca Raton, FL: Auerbach Publications(2005).
[13] D. E. Bell and L. J. LaPadula. "*Secure Computer Systems: Unified Exposition and Multics Interpretation,*" The Mitre Corp., March 1976.
[14] P. C. Clements, C. L. Heitmeyer, B. G. Labaw, and A. T. Rose. "*MT: A Toolset for Specifying and Analyzing Real-Time Systems,*" Real-Time System Symposium, Lake Buena Vista, FL, December 1990.
[15] Andre N. Fredette and Rance Cleveland. "*RTSL: A Language for Real-Time Schedulability Analysis*," Real-Time System Symposium, Raleigh-Durham, NC, December 1993.
[16] T. F. Keefe, W. T. Tsai, and J. Srivastava. "*Multilevel Secure Database Concurrency Control*," In Proceedings of the Sixth International Conference on Data Engineering, pp 337-344, Los Angeles, CA, February 1990.