Volume 2, Issue 12, December 2012 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Partial Security Violation Model in Non-Blocking Distributed Databases

Ranjana Kumari
Dept.of Computer Science
Rajasthan Technical University

Arun Kumar Singh
Dept. of Computer Science
Rajasthan Technical University

Abstract-- This paper will examine the underlying features of the distributed database architecture .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. Developing a successful distributed database system requires to address 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 security, distributed database, distributed database management system, distributed database retrieval problems, discretionary security distributed database, query processing.

1. INTRODUCTION

Today's business environment has 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 provides an improvement on communication and data processing due to its data distribution throughout different network sites. Not only is data access faster, but a single-point of failure is less likely to occur, and it provides local control of data for users. However, there is some complexity when attempting to manage and control distributed database systems.

A. Distributed Database System

A distributed database is a collection of databases which are distributed and the stored on multiple computers within a network". A distributed database is also a set of databases stored on multiple computers that typically appears to applications as a single database. "Consequently, an application can simultaneously access and modify the data in several databases in a network ".

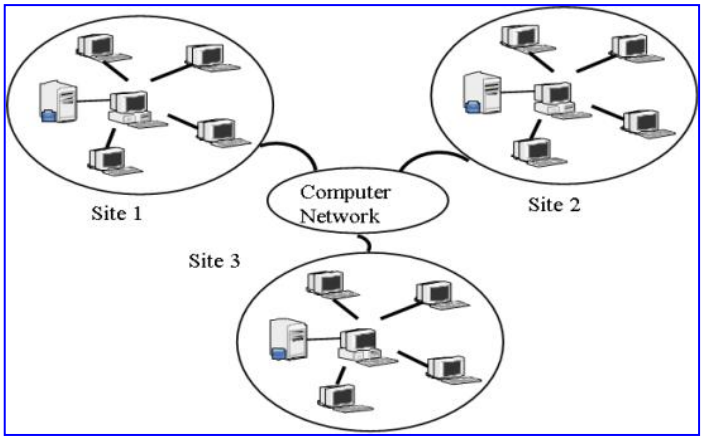

Fig1. Architecture of Distributed Database[1]

A database, link connection allows local users to access data on a remote database. In a distributed database system, the database is stored on several computers. The computers in a distributed system communicate with one another through various communication media, such as high-speed networks or telephone lines. They do not share main memory or disks. Each database may involve different database management system and different architectures that distribute the execution of transactions. The objective of a distributed database management system (DDBMS) is to control the management of a distributed database (DDB) in such a way that it appears to the user as a centralized database. 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.

B. 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. [2] Queries against the database result in inverted lists or secondary key accesses. Such queries would degrade the performance of a centralized system. Fourth-generation languages used locally may require different data structures than the centralized systems.

Each database may involve 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 many objectives of a distributed database system. Such an image is accomplished by using the following transparencies: [3] 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. "[4] Free object naming means that it allows 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 activity of coordinating concurrent accesses to a database in a multi-user database management system (DBMS)." 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 provide better concurrency control than others depending on the system.

C. Security Issues in Distributed Database

Database security is the system, processes, and procedures that protect a database from unintended activity. Unintended activity can be categorized as authenticated misuse, malicious attacks or inadvertent mistakes made by authorized individuals or processes. [5] Database security is also a specialty within the broader discipline of computer security. Traditionally databases 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, 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.

II. PARTIAL SECURITY CONCEPT

In distributed database, partial security is used to improve the performance of system. [7] Systems that are partially secure allow potential security violations such as covert channel use at certain situations. 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.

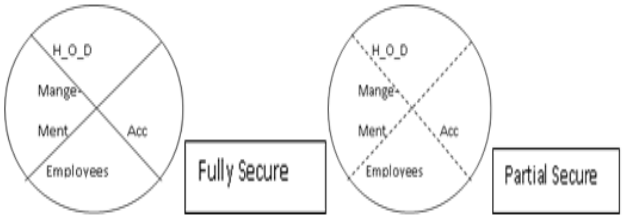

Fig. 2 Full and Partial Security in distributed database[7]

Thus in partial security the violation is allowed in certain security level as 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. [3] Systems that are partially secure allow potential security violations such as covert channel use at certain situations. 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.

A. Framework of Proposed Solution

User is authorized to access the data depending upon permitted user access level. Three levels of users have been proposed depending upon the access permissions given to each user.

A. Level-1:*At Admin/super users level*

This is fully secure level. No information is accessible to the job recruiter or job seeker until security violation is allowed by the administrator himself under certain condition or circumstances.

B. Level-2:*At job provider's (recruiters) level*

At this level, a job provider is allowed to access to list of shortlisted candidates maintained by the other job providers, hence, allowing the security violation to a certain specified extent. He is also allowed to upload the questionnaire from the other recruiter's sites if he requires.

C. Level -3:*At job seekers level*

If the user is unable to get the online test result, he can get it at super user level by violating the security. Even the user can view all available jobs, if he is unable to view the list of jobs at his interface layer. The user is restricted from viewing different fields/data, if he/she does not fulfill the permissible basic security violation criteria.

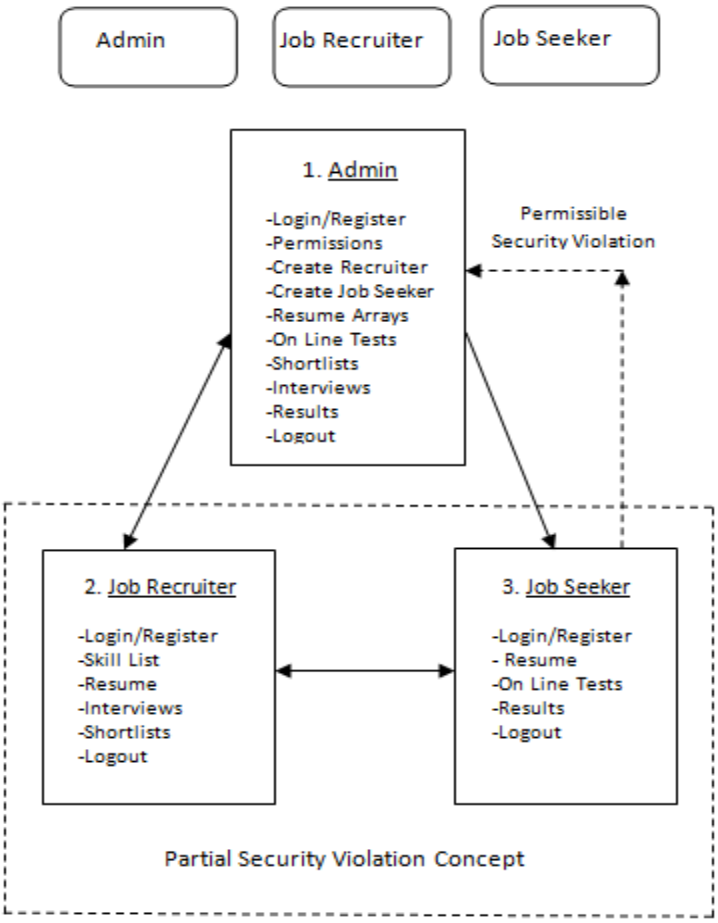

Fig 3. Partial security concept

III. METHODOLOGY & ASSUMPTIONS

IV. PROPOSED ARCHITECTURE

Depending upon the above assumptions we have proposed an architecture that will display the above mentioned methodology levels defined. The proposed architecture consists of three basic levels ADMIN, JOB RECRUITER and JOB SEEKER. Depending on the levels the security violation is defined in the following architecture. At the top level is ADMIN, next level is owned by JOB RECRUITER, last level is obtained by JOB SEEKER.

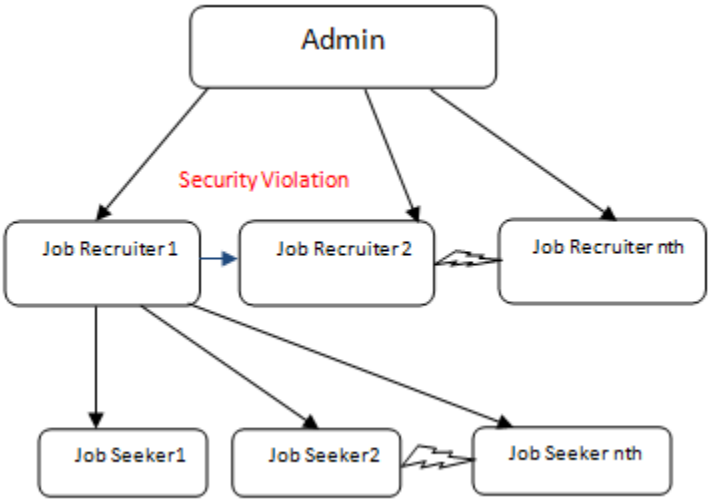

Fig 4. Security violations for Job Recruiter at Layer 2

The above shown figure depicts the partial security violation at the upper level i.e. at level 2 as levels mentioned in the methodology proposed. According to the figure above if jobrecruiter1 wants to share questionnaires from the other job recruiters, he should be able to do this by violating the security at same levels i.e. is at the job recruiters levels. But he is not allowed the access the data at higher level i.e. at admin level because at admin level there is fully security.

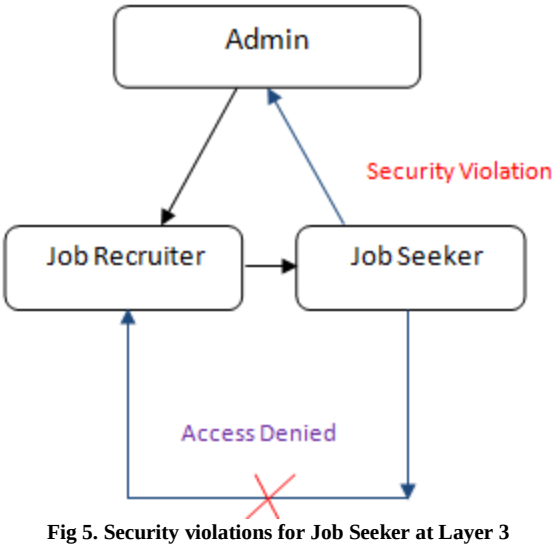

Fig 5. Security violations for Job Seeker at Layer 3

V. RESULTS & DISCUSSION

We work in the field of distributed database to support the better performance of the system by providing user the required information from the same level if possible, if not to provide the same from the upper levels of the layered architecture displayed in the proposed framework above. The factors considered in the proposed work are:-

A. Response Time:

The response time is the total time taken to complete the job requested by the user right from the initial stage till the completion of the job. For betterment of the system the response time is needed to be as minimum as possible.

B. Transaction Access Time:

The transaction access time is the time taken by the system to complete the desired transaction. It constitutes the total time from the transaction taken into queue to process to the final completion of the transaction. For betterment of the system the transaction time is needed to be as minimum as possible same as that as in the case of response time.

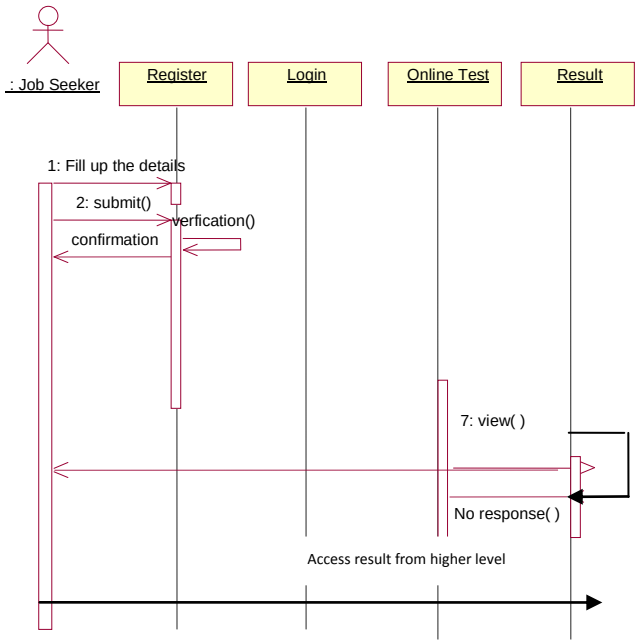

Fig6. Shows the security violation at Layer 3 i.e at Job Seeker Level

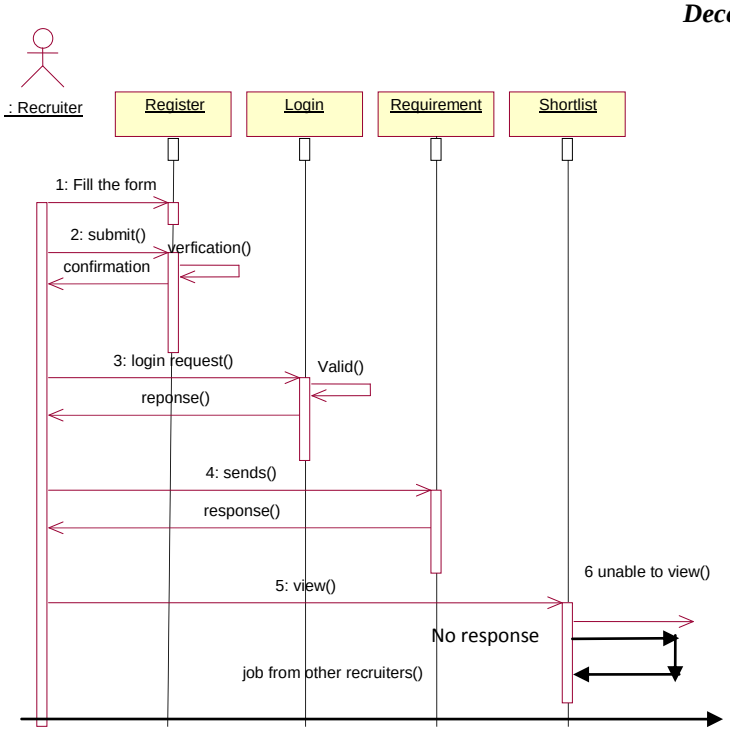

Fig7. Shows the security violation at Layer 2 i.e at Job Recruiter Level

As shown in the figure 6 shows the security violation at level 3, if the job seeker is unable to get result from its corresponding job recruiter that he can be able to access the same information from the top most upper level i.e. the admin level but that approach is permissible. Likewise in figure 7 shows the security violation at level 2, if the job recruiter wants to questionnaires or the job lists matching skill lists to some job seeker he can be able to access this information from the same level i.e. from the job recruiter level.

VI. 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.

REFERENCES

[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.

