International Journal of Engineering Inventions
ISSN: 2278-7461, www.ijeijournal.com
Volume 1, Issue 4 (September 2012) PP: 10-14

Security Management for Distributed Environment

Ms. Smita Chaudhari[1], Mrs. Seema Kolkur[2]
[1]*Assi. Prof. Of S. S. Jondhale College Of Engineering,Dombivli,Mumbai University, INDIA*
[2]*Asso. Prof. Of Thadomal Shahani College Of Engineering, Mumbai University, INDIA*

Abstract—A mobile database is a database that can be connected to by a mobile computing device over a mobile network. Mobile processed information in database systems is distributed, heterogeneous, and replicated. They are endangered by various threats based on user's mobility and restricted mobile resources of portable devices and wireless links. Since mobile circumstances can be very dynamic, standard protection mechanisms do not work very well in such environments. So our proposed model enhances the security in mobile database system. In this paper we develop a security model for transaction management framework for peer-to-peer environments. If any attack still occurs on a database system, evaluation of damage must be performed as soon the attack is identified. The attack recovery problem has two aspects: damage assessment and damage repair. The complexity of attack recovery is mainly caused by a phenomenon called damage spreading. This paper focuses on damage assessment and recovery procedure for distributed database systems.

Keywords—Mobile Database, Transaction Management, Security

I. INTRODUCTION

In mobile environment, several mobile computers collectively form the entire distributed system of interest. These mobile computers may communicate together in an ad hoc manner by communicating through networks that are formed on demand. Such communication may occur through wired (fixed) or wireless (ad hoc) networks. Distributed database systems are made up of mobile nodes and peer-to-peer connection. These nodes are peers and may be replicated both for fault-tolerance, dependability, and to compensate for nodes which are currently disconnected. Several sites from this system must participate in the synchronization of transaction. There are different transaction models [5] available for mobile computing environment, but data transmission between the base station (BS) and the mobile station (MH (S)) is not secure which leads to data inconsistency as well as large number of rejected transactions. Typical operating system security features such as memory and file protection, resource access control and user authentication are not useful for distributed environment. A key requirement in such an environment is to support and secure the communication of mobile database. This paper focuses on security management processing for MCTO (Multi-Check-out Timestamp Order) [2] model by using symmetric encryption and decryption [1] between the Base station BS and the mobile host MH with the aim at achieving secure data management at the mobile host.

If any attack occurs on a database system, evaluation of damage must be performed as soon the attack is identified. If the evaluation of damage not performed soon after attack, the initial damage will spread to other parts of the database via valid transactions, consequently resulting in denial-of-service. As more and more data items become affected, the spread of damage becomes even faster. Damage assessment is a complicated task due to intricate transaction relationships among distributed sites. For the assessment, the logs need to be checked thoroughly for the effect of the attack. Damage recovery [6] can be "Coldstart" or "Warmstart". This paper focuses system that uses the "Coldstart" method for damage assessment and recovery. The proposed system uses DAA (Damage Assessment Algorithm) [3] to detect the spread of malicious transaction in distributed replicated database system. After detection of affected transactions, these are recovered using the recovery procedure.

II. THE PROPOSED MODEL

The architecture of the proposed system is as shown in fig.1. The mobile host in mobile network first gives the encrypted request to fixed proxy server. The fixed proxy server updates the data and the result is given back to the mobile network.

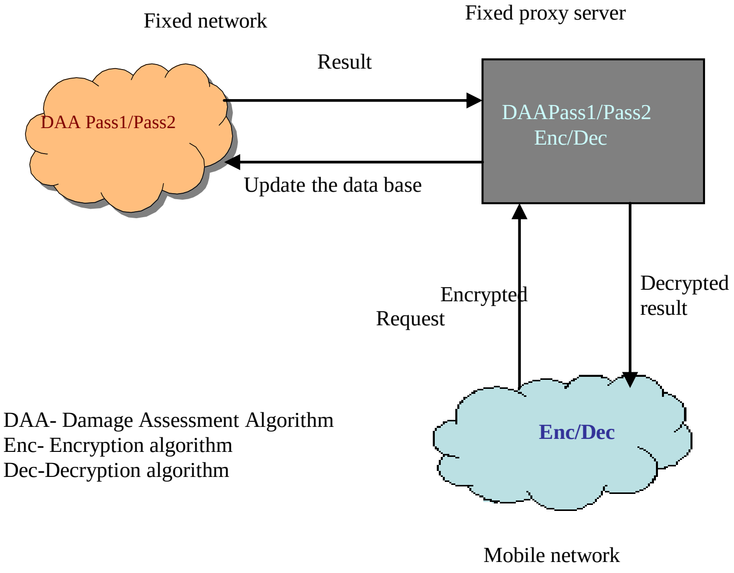

Fig.1 Architecture of the proposed system

The proposed model consists of both encryption and decryption algorithms located at the BS and the MH(s) as shown in Fig.1.The encryption algorithm is started when the data transferred. The decryption algorithm is started when encrypted data is received. The DAA (Damage assessment Algorithm) on the fixed network uses local logs.

The proposed system used MCTO model [2]. The model has two types of networks, i.e., the fixed network and the mobile network. For the fixed network, all sites are logically organized in the form of two-dimensional grid structure. For example, if the network consists of twenty-five sites, it will logically organize in the form of 5 x 5 grids. Each site has a master data file.

A. Diagonal replication on Grid (DRG) Technique

For replication, the proposed system used a DRG [4] technique. In the fixed network, the data file will replicate to *diagonal sites*. While in the ad hoc network, the data file will replicate asynchronously at only one site based on the most frequently visited site. As an example, Assume that in5x5grid, the same file will be replicated to s (1,1),s (2,2), s (3,3),s(4,4) and s(5,5). The 'commonly visited site' is defined as the most frequent site that requests the same data at the fixed network (the commonly visited sites can be given either by a user or selected automatically from a log file/ database at each centre).

B. Damage evaluation protocol

The purpose of this model [3] is to provide an efficient method to assess the effects of a malicious transaction in a fully distributed replicated database system. The model is based on the following assumptions.

- The local schedules are logged in each site and the attacker cannot destroy the log. The extended log can be considered to include all the read operations in addition to the write operations in the log.
- The attacking transactions are identified.
- Blind writes are not permitted. That is if a transaction writes some data item it is assumed to read the value of that item first.

The following transaction classifications are used in the model: Malicious transaction, authentic transaction, affected transaction, bad transactions and unaffected transaction. Consider a distributed database system consisting of two logs as shown in fig.3 that are replicated at different sites. Since this is a replicated distributed database system, any change in one log is appeared to every site where that log is replicated.

R_1 (a) W_1 (x)	$R_{2.2}$(c)
$R_{2.1}$ (b) $W_{2.1}$ (b)	$R_{4.2}$(c) $W_{4.2}$(c)
$R_{4.1}$ (d)	R_5 (g)
$R_{6.1}$ (d) $W_{6.1}$ (d)	$R_{6.2}$ (h) $W_{6.2}$ (h)
R_7 (d)	R_8 (g) W_8 (g)
$R_{12.1}$ (e) W12.1 (e)	$R_{9.2}$ (g) $W_{9.2}$ (g)
$R_{14.1}$ (a)	$R_{12.2}$ (k)
R_{25} (d) W_{25} (y)	$R_{21.2}$ (g) $W_{21.2}$ (g)
Site 1	Site

Fig.3 Effect of Transaction distribution

The underlined transactions are malicious transactions. Let us first consider the log at site 1 in which T6 is marked as the first malicious transaction. When DAA procedure is executed at site 1 the affected transactions will be detected and added to the undo list (the list of transactions whose effect must be removed from the database). For example, transaction T7 and T25 are affected transactions since they both read the damaged data item *d*, which has been updated by T6. At site 2, T8 is marked as malicious hence T9 and T21 will be detected as affected. After assessing the affected transactions, all affected transactions are repaired using damage recovery procedure.

i)Proposed Algorithm for Damage Assessment

Input: The update log, read log, the set of malicious transactions M

Output: The set of bad transactions **TB**, the set of dirty items **D** and the set of global bad Transactions **GB.**

```
1. Initializations: TB ={}, tmp_bad_list ={}, D = {}, tmp_dirty_list    = {}.
2. Find the first malicious transaction committed in the log.
3. For each transaction Ti read its entry $P_{ij}$ (x) in the log
    3.1 If Ti is in M then
         If $P_{ij}$ is a write operation then
             D = D U{x}
             \* Add the data item to the dirty list*\
    3.2 Else
         3.2.1 Case $P_{ij}$ is a read operation
                 If x is in D
                    tmp_bad_list = tmp_bad_list U {Ti}
                    /*Add Ti to tmp_bad_list*/
         3.2.2 Case $P_{ij}$ is a write operation
                    tmp_dirty_list = tmp_dirty_list U {x}
                    /* Add x to tmp_dirty_list*/
         3.2.3 Case $P_{ij}$ is an abort operation
                    Delete Ti from tmp_bad_list
                    Delete x from tmp_dirty_list
         3.2.4 Case $P_{ij}$ is a commit operation
                 If Ti is in the tmp_bad_list
                    Move Ti from tmp_bad_list to TB.
                    Move all the data items of Ti from the
                    tmp_dirty_list to the D.
```

The algorithm starts by adding data items updated by malicious transactions to the dirty list. After that it scans the log for all presumed-bad transactions and adds the data items that have been updated by them to the dirty list.

ii) Proposed Algorithm for Damage Recovery

Input: The update log, set of malicious transaction M, set of affected transactions A.

Output: Set of recovered transactions whose effect has been undone: UN

Intermediate input/output: The set of bad transactions TB, the set of dirty items D, difference, temp.

```
1. Move to the position in the log where the first malicious transaction appeared.
2. For each transaction Ti in M,
        temp ← old value of item X
3. For each Affected transaction Ti in A, read its entry Pij (x) in the log.
    3.1 If x is a numeric value,
          Read the old values and new values of item x from the log.
        Calculate difference ← new value of x – old value of x.
    3.2 Else
        Read the old values and new values of item x from the log.
4. For every transaction Ti in the update log, read its entry Pij (x) in the log
    4.1 If Ti is in A,
        4.2 If x is a numeric value
            Update Old value of x ← temp
            New value of x ← temp - difference.
        4.3 Else
            Copy temp to new value of x.
5. Update the original table’s state according to recovered transactions in the log.
```

The algorithm begins by scanning every affected transaction in the update log. If the item updated by affected transaction is a numeric value, it calculates the difference between the old value and new value of the item. The difference is updated to new value of item. If item is a character value, copy old value of item to new value. Update the changes in the tables to bring the database back in running state.

III. IMPLEMENTATION ISSUES

A. System Design

The system uses VB as front-end and Oracle9i as a backend. There are two parts in which the proposed security management system works. In the first part, the mobile user retrieves the desired information from the distributed database system. The mobile user connects with distributed database system using wireless communication. During communication, the request or result may get violated during transmission. To avoid it, the communication between mobile user and distributed database system is secured using MD5 algorithm. The database is replicated on distributed database system using Oracle’s MMR (Multi-Master Replication) in such a way that the consistent data is available at different sites.

In the second part, Damage Assessment and Repair module is developed. The main component of this module is log, which captures different operations on database. Two types of log are required: read log and update log. The prototype is implemented on top of an Oracle server. Since Oracle redo log structure is confidential and difficult to handle, read and write information is maintained manually. In particular, a Proxy is used to mediate every user transaction and some specific triggers to log write operations. A trigger is associated with each user table to log the write operations on that table. All the write operations are recorded in the log table.

Oracle triggers cannot capture every read operation. Oracle triggers can capture the read operations on a data item that is updated or deleted but cannot capture read-only operations. To capture every read operation, read set from SQL statement is extracted and stored in read log table.

B. Performance Issues

The performance of the first part i.e. data retrieval can be measured by the response time for user transactions. It is the time between mobile users enters his queries to the time he gets back the result. It depends on the number of records the user wants to retrieve.

The performance of the attack recovery subsystem can be measured by the average repair time, which indicates how efficient the subsystem is at repairing damage. The Average repair time depends on the number of Affected Transactions.

IV. CONCLUSIONS AND FUTURE WORK

In this paper we have developed a mobile transaction model, which captures data and movement nature of mobile transactions. This model is based on multi-check-out. The model describes a mobile transaction Management by Timestamp Order. The Encryption and Decryption algorithms are used for secured data transmission between the base station BS and the mobile host MH(s). After an attack, the DAA procedures are used to avoid and repair the effect of malicious transactions.

As a future work, an agent can be used on the fixed proxy server. When the mobile client is disconnected in MCTO model, the result of the transaction is not lost but will be stored with the mobile agent. When the transaction is completed, the agent returns and delivers the result to the user. If the user is disconnected, it waits until the user is reconnected. The Agent is also used in order to maintain serializability in multi check-out mode, timestamp ordering to serialize the mobile transaction at the fixed proxy server.

REFERENCES

1. Abdul-Mehdi, Z.T.; Mahmod, R., “Security Management Model for Mobile Databases Transaction Management” Information and Communication Technologies: From Theory to Applications, 2008.,3rd International Conference on
7-11 April 2008 Page(s):1 – 6.

2. Mehdi, Z.T. Mamat, A.B. Ibrahim, H. Dirs, Mustafa.M. 2006.“Multi-Check-Out Timestamp Order Technique (MCTO) for Planned Disconnections in Mobile Database”, The *2nd* IEEE International Conference on Information & Communication Technologies*: from Theory to Applications,* 24-28 April, Damascus, Syria, Vol.1, and p.p 491-498.

3. Rami Samara and Brajendra Panda “Investigating the Effect of an Attack on a Distributed Database” IEEE Workshop on Information Assurance United States Military Academy, West Point, NY 1-4244-0130-5/06/$20.00 ©2006 IEEE 312

4. M. D. Mustafa, B.Nathrah, M. H. Suzuri, M. T. Abu Osman “Improving Data Availability using hybrid Replication technique in Peer-To-Peer Environments” processing of the 18th international conference on Advanced Information Networking and application 0-7695-2051-0/04 $ 20.00@ 2004 IEEE

5. Weider D. Yu, Sunita Sharma “A Mobile Database Design Methodology for Mobile Software Solutions” 31st Annual International Computer Software and Applications Conference (COMPSAC 2007)

6. Paul Ammann, Sushil Jajodia, Peng Liu “Recovery from Malicious Transactions” IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL.14,NO.5,SEPTEMBER/OCTOBER 2002