MAINTAINING INTEGRITY WITHIN MOBILE SELF PROTECTING OBJECTS

Wesley Brandi

Department of Computer Science, Rand Afrikaans University

PO Box 524, Auckland Park, Johannesburg, 2006 South Africa

wb@eclab.rau.ac.za

Martin S Olivier

Department of Computer Science, Rand Afrikaans University

PO Box 524, Auckland Park, Johannesburg, 2006 South Africa

molivier@rkw.rau.ac.za

Abstract This paper examines the integrity issues involved when a Self Protecting Object (SPO) is moved to a site in a federated database which will eventually disconnect and become unreachable for some time. The SPO model guarantees that the custom security policy of a site participating in a federated database will be implemented and respected when the object it shares is accessed by others in the federated database, regardless of the objects location.

We introduce the Mobile Self Protecting Object (MSPO) and propose an architecture within which it will operate. Having looked at integrity issues which may arise we propose a way in which to maintain integrity within Mobile Self Protecting Objects. In particular, we propose a way in which to deal with mobile transactions which require authorisation. We discuss how MPSOs can be updated whilst in a mobile environment as well as how to ensure an MSPO has maintained its integrity upon reentering the federated database from a mobile site.

Keywords: Mobile self protecting object, federated database, integrity, security

1. INTRODUCTION

The demand for access to information in a database from a mobile environment is steadily increasing [TIP00]. This trend is applicable to clients in a federated database requesting access to a Self Protecting Object (SPO) from some form of mobile media. The SPO model pro-

posed by [OLI96] allows objects in a federated database to move from their originating site to a trusted remote site in the federated database whilst guaranteeing that the originating site's security policy will be implemented.

It may not suffice to merely have access to an SPO through the federated database from a mobile environment since the need may arise to access an SPO when access to the federated database is not feasible. This demand gives rise to a new form of SPO, the Mobile Self Protecting Object (MSPO).

An MSPO can be defined as an SPO that has been relocated from a site in the federated database to a mobile site for the exclusive use of the mobile user of that site upon disconnection. [BAD95] refers to such clients (sites) as *hoard clients*.

We define integrity within SPOs as a means of ensuring the validity of an SPO within a federated database environment. When we now consider integrity within MSPOs, the definition of integrity remains, but the way in which the integrity of an MSPO is maintained differs. When an MSPO is once again returned to the federated database from the mobile media on which it was previously located, how can one be certain that the MSPO has maintained its integrity? The aim of this paper is to provide and define a framework for the MSPO concept as well as to examine how integrity within MSPOs can be maintained and implemented.

In section 2 we briefly discuss the SPO architecture and the framework within which it operates. Before we discuss how integrity within the MSPO architecture can be implemented we first propose a model for the MSPO architecture. This is covered in section 3. We then examine how integrity can be implemented and maintained in section 4. The paper is concluded in section 5.

2. BACKGROUND

We follow The SPO model proposed by [OLI96]. It is an object-oriented model implemented within a federated database architecture. A federated database is a distributed database [CP85] where participating sites (database systems) are fairly autonomous. See [SL90] for a thorough discussion of federated databases. A federated database defines a general security policy which is implemented by all participating sites. The SPO model takes this further in that each site, having implemented the federated security policy, extends it to include their own security policy. It is assumed that this security policy will be respected and implemented by other sites when accessing data on that site.

This model therefore allows objects originating at one site to be relocated to another, ensuring that the security policy of the originating site will be implemented. Each site in the federated database must implement a core layer of the SPO model, this core is referred to as the Trusted Common Core (TCC), this core is responsible for ensuring that each site's security policy is maintained. In addition to this, code can be appended to objects in the form of a Trusted Extension (TE). This module is a form of a portable security policy specific to the object it is embedded in. In order for the TE of an object to function correctly, a Trusted Local Extension (TLE) is included at each site in the federated database, it provides any support required by the TE.

3. MSPO ARCHITECTURE

We define an MSPO as an SPO that has been transferred into a mobile environment and subsequently removed from the federated database. Before an SPO can be made mobile, the site requesting the transfer must be authorised to do so by the TLE of the site where the SPO originated from. The TLE of the original site will receive a request to make the SPO mobile. The request is much like that of a normal transfer request, in that an SPO is being moved from site A to site C. It differs only in that site C belongs to its parent site B, and site C will eventually disconnect itself from its parent site and therefore from the federated database. Site C is therefore a mobile site within the member site B.

3.1. MOBILE SITES

[OLI97] proposes a model for mobiles nodes (sites) in a federated database. This model has each site hosting a mobile site implement a modified TCC, referred to as an HTCC (host TCC). Each mobile site then implements a Mobile Trusted Core, which essentially mirrors the HTCC. SPOs moved from a federated site to a mobile site are modified prior to their relocation. In being modified, the object will not include any methods or variables that the mobile site will not be authorised to use. This implies that once an SPO has been relocated to the mobile site, no authorisation will be necessary to use it.

We propose an alternative approach to the mobile architecture; we assume that an SPO will not be used at all if it is not accessed through the TCC. One can argue that modifying an SPO does not necessarily address the problem of security in our model, since although an SPO will be secure once modified and copied to a mobile site. This is not the

case when relocating an SPO to a normal site in the federated database, since SPOs relocated to these sites will not be modified.

A mobile site in our proposed model is much like that of a site which is a member of a federated database. The mobile site is slightly different in that it is a child site to a parent site which in turn is an official member site of the federated database. Some degree of authentication must exist between the mobile child site and its parent site, thus minimising the risk of an intruder masquerading as the mobile child site.

A mobile site must implement and trust the TCC, therefore as is the case with member sites of the federated database, a mobile site can only use an SPO through the TCC. This level of trust is essential. An SPO can not be moved to a site where the TCC is not trusted or is not implemented completely. In our proposed model, a mobile site can only have one parent site and can only connect to the federated database via this parent site. Figure 1 illustrates the mobile site architecture.

Figure 1 The mobile site model

3.2. RELOCATION

Relocating an SPO to a mobile site is similar to the transfer process of an SPO. Since a mobile site is not a complete member of the federated database, it will have its parent site make the request it's behalf. As is the case with transfer requests of an SPO in a federated database, the request to move an SPO to a mobile site must be authorised. When moving an SPO to a mobile site, authorisation must be gained from the SPO's originating site.

A request to move an SPO from site A to site C, assuming site C is a mobile site belonging to site B, would have the TCC include this

information in its request to the TLE of the SPOs home site. Upon authorisation, the TLE will then perform several tasks which are essential to maintaining the integrity of the soon to be MSPO:

- Keep record that the SPO has been moved to the mobile site C, a child site of the member site B.
- Create a duplicate of the SPO and save it locally.
- Create and store hashes of all the values in the SPO that require authorisation to be modified.
- Create and store a hash of the entire SPO.

If the mobile site that an SPO has been relocated to is still active, in that it has not yet been disconnected from the federated database, the SPO can still be used normally, since the TCC of the mobile site is similar to a TCC of a participating site in the federated database.

3.3. REMOVAL FROM THE FEDERATED DATABASE

The fact that a site is mobile means that it will eventually become unavailable, in that it will disconnect itself from the federated database. Before a mobile site can disconnect from the federated database (its parent site), it must request authorisation from its parent site. If an MSPO that the mobile site is hosting is currently in use, the request to disconnect the site will be denied. Once the request has been authorised, the mobile site is then free to disconnect. The parent site will then inform the originating sites of the MSPOs hosted by the mobile site that the MSPOs have been disconnected from the federated database.

Having been disconnected from the database, an SPO is no longer available for modification in the federated database until it has been reintroduced into the system. However, if necessary, it is possible for sites to retrieve data from the duplicated SPOs as long as the sites retrieving the data understand that the SPOs are currently active elsewhere, hence, the data may not be valid. We do not analyse the concurrency issues involved when an SPO becomes mobile and is disconnected from the federated database; this is not within the scope of this paper and will be covered in future research.

Upon disconnection of a mobile site from the federated database, a timer on the parent site will be associated with the length of time of that the mobile site has been disconnected. If the timer exceeds a predefined threshold, the mobile site and the MSPOs it was hosting will be consid-

ered lost and the duplicate SPOs will be introduced into the federated database from the SPOs respective original sites.

This rollback policy will ensure that any MSPO that has been lost while disconnected from the federated database will be reintroduced into the database in its most recent consistent state.

Since a mobile site may disconnect without prior warning, and therefore without authorisation, a site hosting the mobile site must be able to handle such situations effectively. The primary reason for a mobile site to obtain authorisation before disconnecting is to accommodate any transactions that may be running on any of the MSPOs it is currently hosting. The worst case scenario in such a situation may be that a transaction does not run to completion. This transaction would have to be aborted, the parent site would then begin the process of handling the mobile site disconnection, as described earlier.

4. MAINTAINING INTEGRITY OF AN MSPO

The proposed MSPO architecture of the previous section gives us a framework in which to now examine and discuss methods as to how the integrity of an MSPO can be maintained. We believe that there are three major issues, that when addressed and implemented in a proper manner, will ensure integrity within the MSPO architecture. The issues we have identified are the following:

- The proper management of transactions in a mobile environment that require aut horisation.
- An orderly and secure manner in which to update MSPOs.
- A process of re-entry for an MSPO into the federated database.

We begin this discussion by looking at general functionality of an MSPO and move onto an analysis of how transactions which require authorisation are dealt with whilst the mobile site is disconnected. We will then discuss how an MSPO can be updated and examine the process involved when an MSPO is to re-enter the federated database.

4.1. MSPO FUNCTIONALITY OUTSIDE OF THE FEDERATED DATABASE

Since the site an MSPO is hosted on is mobile, upon disconnection the TCC associated with that site may have no means of making contact with any other sites of the federated database. This has an impact particularly on transactions which require some kind of interaction with

the TLE of an MSPO's originating site. Since the TCC may have no way of establishing contact with any member sites of the federated database, transactions which require authorisation or some kind of interaction with the MSPO's originating site may be denied execution by the TCC.

In light of this problem, before a transaction will be allowed to execute within a mobile site, the TCC will first analyse the nature of the transaction. From this, the TCC will determine what (if any) authorisation requirements there are for the transaction and whether or not the authorisation may be attained.

It could be the case that a mobile site may be able to establish contact with its parent site via a cellular link or remote network, thereby allowing the TCC to communicate with other sites of the federated database and attain proper authorisation for transactions. On the other hand, it may be the case that the mobile site may not be able to establish contact with its parent site, any transactions requiring authorisation will therefore be denied permission to execute.

Figure 2 illustrates the flow of operations as a transaction which requires authorisation from a site on the federated database attempts to execute.

1. Request to execute transaction 'T1'.
2. The TCC analyses the nature of transaction T1 and determines authorisation from site A is required.
3. The TCC attempts to establish a connection to its parent, site B.
4. Since the TCC can not establish a connection to site B, transaction T1 is denied permission to execute.

Figure 2 An illustration of the steps taken by a mobile site to establish a connection for a transaction requiring authorisation.

Since an MSPO can only be used within the framework of a trusted TCC on a mobile site, and since transactions which require authorisation will only be executed provided that the TCC of the mobile site can establish contact with its parent site, we can be assured that the integrity of the MSPO will not be maliciously compromised whilst it is disconnected from the federated database.

The integrity of each MSPO is still subject to several forms of compromise the likes of errors, viruses or failures in the system [CFM95]. Section 4.4 discusses how such compromises are detected and what action is taken upon detection.

4.2. EXECUTING TRANSACTIONS WHICH REQUIRE AUTHORISATION

If the TCC of a mobile site is able to establish contact with its parent site, it can begin the process of obtaining authorisation for the transaction to be executed within the mobile site on an MSPO. Having obtained authorisation the transaction will execute and run to completion. Upon completion of the transaction, steps similar to those performed on an SPO will be taken in order to maintain the integrity of the MSPO: A duplicate of the MSPO will be made and sent to its originating site where a hash of the MSPO as well of each value that requires authorisation to be modified will be created and stored.

As an example of executing transactions on mobile sites which require authorisation, consider a car salesman who has moved an SPO of type Car onto his laptop. He is currently at a client's premises and the client has requested details on the lifetime of the gear box. The salesman queries the gear box lifetime property of the Car MSPO and is subsequently requested to connect to the federated database. The gear box lifetime property is a property which requires authorisation in order to be read. The TCC on the salesman's laptop observed the nature of the salesman's query and identified the need for authorisation, hence, the salesman has been prompted to connect to the federated database.

Should the salesman not connect to the database, the TCC will be unable to obtain authorisation and the query to view the gearbox lifetime property will be denied. If the salesman does connect the laptop to the federated database, he may be prompted for a username and password, upon successful authorisation the TCC will grant permission to run the query and hence view the gearbox lifetime property.

We have so far assumed that should a mobile site be unable to establish a connection to its parent site, any transactions that require authorisation will simply be denied permission to execute and subsequently

deleted. Further research may show that a more feasible approach may be to queue the transaction until the mobile site is able to connect to its parent site, obtain authorisation for the transaction and then allow the transaction to execute.

4.3. UPDATING AN MSPO

There are several problems in allowing an MSPO to be updated whilst disconnected from the federated database. We are concerned with the problem of integrity. Since an MSPO is not connected to the federated database, how can we be certain that an MSPO which has been updated whilst disconnected from the federated database has undergone a series of authorised modifications? One could argue that we can be sure the MSPO has maintained its integrity since it will only be used via the TCC.

Within the framework we have proposed so far, this is indeed the case, but usage of an MSPO limited. There will be properties of an MSPO which require authorisation in order to be modified. Since the MSPO is on a mobile site, connecting to the federated database to obtain authorisation may not always be feasible. Does this mean that one is denied the ability to update an MSPO property which requires authorisation, unless the mobile site can establish contact with the federated database? This is indeed the case if we are to be certain that integrity will be maintained.

An alternative to this approach may be to make use of a slightly modified TCC on the mobile site, modifications can be made to the way a TCC handles transactions in general. The TCC could allow transactions which require authorisation to simply be executed, whilst maintaining a log of all the activities of each transaction. Upon re-entry into the federated database the transaction log for the MSPO would be analysed and executed on the duplicate copy made of the MSPO before becoming mobile. If unauthorised modifications are identified, these transactions can simply be ignored. The duplicate copy of the MSPO, having undergone all the authorised transactions of the transaction log would then be permitted to re-enter the federated database. Figure 3 illustrates the proposed alternative to handling unauthorised transactions.

1. Request to execute transaction T1'.
2. Authorisation to execute the transaction is automatically granted.
3. The TCC logs the transaction.
4. Upon reconnection, the logged transactions are executed on the duplicate SPO. All unauthorised transactions are ignored.
5. If this task is completed successfully, the SPO is granted access into the federated database.

Figure 3 An alternative to handling unauthorised transactions.

4.4. RE-ENTRY INTO THE FEDERATED DATABASE

Before an MSPO can be allowed back into the federated database, one has to be certain it is an MSPO which has undergone a series of authorised transactions and has, essentially, maintained its integrity.

Assuming the time that the mobile site has been disconnected has not exceeded the maximum time that it is allowed to be disconnected, each MSPO that is hosted by the mobile site will in turn undergo a re-entry process which will determine whether or not the MSPO will be allowed to re-enter the federated database.

The primary goal of the re-entry process is to ensure that the MSPO being reintroduced into the database has not had its integrity compromised and at the very least, is the MSPO that it claims to be. As the mobile site reconnects to its parent site, the MSPO will be moved from the mobile site to its originating site. The Unique Identifier (UI) of the SPO will then be extracted and a hash made of the object as well as hashes made of each of the values that require authorisation in order to be modified. The UI and the hashes will then be compared to the duplicated UI and hashes stored at the SPO's originating site before the SPO became mobile or after any authorised modifications made whilst the MSPO was mobile. If they are alike, the SPO will then undergo the second and final phase of the re-entry process.

Should the SPO fail the first phase of the re-entry process, the SPO will be denied re-entry into the database. Having failed the first phase, one can with certainty deduce that the integrity of the SPO has indeed been compromised. The SPO will be deleted and the duplicate copy made at its originating site will be used to reintroduce the SPO into the federated database, in its last uncompromised state.

The second phase of re-entry has the SPO undergo an order of operations which will ensure that any valid changes made to the SPO whilst mobile will now be saved and the SPO will be declared as being in a valid, uncompromised state. The new duplicate copy and hashes will simply replace the previous duplicate copy and hashes stored on the SPO's originating site.

5. CONCLUSION

In this paper we have introduced the concept of Mobile Self Protecting Objects and have proposed an architecture within which it can be implemented. We have discussed how MSPOs can be used on mobile sites and how transactions requiring authorisation can be handled with the ultimate goal of ensuring that the integrity of the MSPO will always be maintained.

We have assumed throughout this paper that mobile sites will implement and trust the TCC. In doing so, we can be certain that the integrity of an MSPO will be maintained regardless of malicious intent or error.

References

- [BAD95] B. R. Badrinath, Shirish Hemant Phatak, An Architecture for Mobile Databases, Rutgers University, Department of Computer Science, Technical report #DCS-TR-351 1995
- [CFM95] Silvana Castano, Mariagrazia Fugini, Giancarlo Martella, Pierangela Samarati, Database Security, Addison-Wesley & ACM Press 1995
- [CP85] Stefano Ceri and Guiseppe Pelagatti, Distributed databases - principles and systems, McGraw-Hill 1985
- [OLI96] Martin S Olivier, Self-Protecting Objects in a Secure Federated Database, In Spooner, SA Demurjian and JE Dobson (eds), Database Security IX: Status and Prospects, 27-42, Chapman & Hall, 1996
- [OLI97] Martin S Olivier, Secure Mobile Nodes in Federated Databases-base, South African Computer Journal, 20, 11-17, 1997
- [SL90] Amit P. Sheth, James A. Larson ACM Computing Surveys, Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases, Vol. 22, No. 3, September 1990

[TIP00] Kimberly Tipton, Fortune 1000 Executives Forecast Dramatic Increase in Need for Mobile Communications, AVT Corporation News Release, http://www.mobiledelivery.com/pr_report.htm 2000