Extended RBAC –Based Design and Implementation for a Secure Data Warehouse

Dr. Bhavani Thuraisingham
The University of Texas at Dallas
bhavani.thuraisingham@utdallas.edu

Srinivasan Iyer
The University of Texas at Dallas
Sxv051100@utdallas.edu

Abstract

This paper first discusses security issues for data warehousing. In particular, issues on building a secure data warehouse, secure data warehousing technologies as well as design issues are discussed. Our design of a secure data warehouse that enforces an Extended RBAC Policy is described next. Finally directions for secure data warehouses are discussed.

1. Overview

Data warehousing is one of the key data management technologies to support data mining and other decision support functions. Several organizations are building their own warehouses. Commercial database system vendors are marketing warehousing products. As stated in [INMO93], the idea behind a data warehouse is that it is often cumbersome to access data from the heterogeneous databases. Several processing modules need to cooperate with each other to process a query in a heterogeneous environment. Therefore, a data warehouse will bring together the essential data from the heterogeneous databases. This way the users need to query only the warehouse. Essentially data warehouses provide support for decision support of an enterprise. For example, while the data sources may have the raw data, the data warehouse may have correlated data, summary reports, and aggregate functions applied to the raw data.

Now, in order for the data warehouse to be useful in many applications such as medical, financial, defense and intelligence, it must be secure. In other words the data warehouse must enforce the security policies enforced by the back-end data sources in addition to possibly enforcing additional security properties. Figure 1 illustrates a high level view of a secure data warehouse. The data sources are managed by secure database systems A, B, and C. The information in these secure databases are merged and put into a secure warehouse.

In this paper we discuss security for data warehousing. The organization of this paper is as follows. Some issues on building a secure warehouse is discussed in Section 2 which also includes the design issues for a secure data warehouse. Next our design and implementation of a secure warehouse based on an extended RBAC model will be discussed in section 4. Directions are discussed in section 5.

2. Security for Data Warehouses

2.1. Security Issues

There are various ways to building a secure data warehouse. One is to simply replicate the secure databases and enforce an integrated security policy. This does not have any significant advantage over accessing the secure heterogeneous databases. The second approach is to replicate the information, but to remove any inconsistencies and redundancies. This has some advantage, as it is important to provide a consistent picture of the databases. The third approach is to select a subset of the information from the databases and place it in the warehouse and at the same time ensuring that security is maintained by the warehouse. There are several issues here. How are the subsets selected? Are they selected at random or is some method used to select the data? For example, one could take every other row in a relation (assuming it is a relational database) and store these rows in the warehouse. The fourth approach, which is a slight variation of the third approach, is to determine the types of queries that users would pose, and then analyze the data, examine security policies to be enforced and store only the data that is required by the user. We will call this secure on-line analytical processing (SOLAP) as opposed to secure on-line transaction processing (SOLTP) where the back-end secure database systems are queried.

With a data warehouse, data may often be viewed differently by different applications. That is, the data is multidimensional. For example, the payroll department may want data to be in a certain format while the project department may want data to be in a different format. The warehouse must provide support for such multidimensional data. Furthermore different security policies may be enforced at different levels. For example, only managers

Second International Conference on Availability, Reliability and Security (ARES'07)
0-7695-2775-2/07 $20.00 © 2007 IEEE

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on April 29,2010 at 17:03:30 UTC from IEEE Xplore. Restrictions apply.

can see the individual salaries while the project leaders see average salaries.

In integrating the data sources to form the warehouse, a challenge is to analyze the application and select appropriate data to be placed in the warehouse. At times, some computations may have to be performed so that only summaries and averages are stored in the data warehouse. Note that it is not always the case that the warehouse has all the information for a query. In this case, the warehouse may have to get the data from the heterogeneous data sources to complete the execution of the query. Another challenge is what happens to the warehouse when the individual databases are updated? How are the updates propagated to the warehouse? How can security be maintained when propagating the updates? These are some of the issues that are being investigated. Security cuts across all layers and operations of the warehouse. In [THUR97] we discussed security for data warehousing. Since then there have been some efforts on secure data warehouses (see for example [BEST02]).

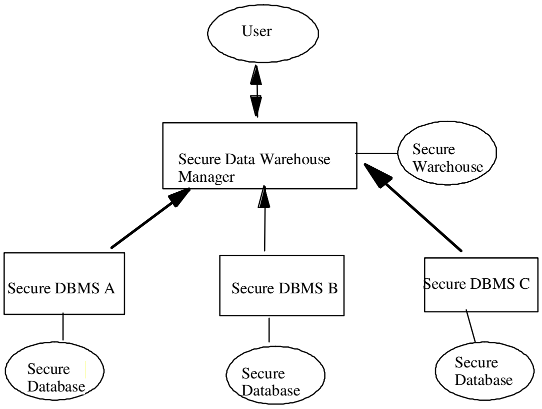

Figure1. Secure Data Warehouse Example

One of the major security and privacy challenges for data warehousing is the inference and privacy problem. For example, the data warehouse aggregates the data. Therefore the aggregated data could be highly sensitive or private while the individual data values may be unclassified or public. Various privacy preserving data mining and data warehousing approaches are being investigated [THUR05], [THUR03].

2.2. Technologies for Secure Warehousing

Note that several secure information technologies have to be integrated to develop a secure data warehouse. These include secure heterogeneous database integration, statistical databases, secure data modeling, secure metadata management, secure access methods and indexing, secure query processing, secure database administration, general database security, and secure high performance database management.

Secure heterogeneous database integration is an essential component to data warehousing. This is because data from multiple secure heterogeneous data stores may have to be integrated to build the warehouse. In the case of secure heterogeneous database integration discussed in [THUR94], there is usually no single repository to store the data. However, in a secure warehouse there is usually a single repository for the warehouse data and this repository has to be managed and security policies enforced.

Statistical databases keep information such as sums, averages, and other aggregates. There are various issues for statistical databases. For example, how can summary data maintained when the database gets updated? How can the individual data items be protected? For example, the average salary may be Unclassified while the individual salaries are Secret. Since warehouses keep summary information, techniques used to manage statistical databases need to be examined for warehouses.

Secure data modeling is an essential task for building a data warehouse. Is the secure data model influenced by the data models used by the back-end secure heterogeneous data sources? Should a data model be developed from scratch? Inmon has outlined several steps to developing a data model [INMO93]. He says that at the higher level there are three stages: developing a corporate model, an enterprise model, and a warehouse model. At the middle level there may be a model possibly for each subject, and at the physical level it includes features such as keys. Some argue this is too lengthy a process and that one should get to the warehouse model directly. As more experiences are reported on developing data warehouses, this issue may be resolved. New types of data models such as multidimensional data models and schemas such as star-schemas have been proposed for data warehousing. We need to integrate these models with secure data models that we have discussed in [THUR93a]. For example, in a project database, there is a central table that has key information on projects such as project number, project leader, estimated time duration, cost and other pertinent data. Each of the entries in this table could be elaborated in other tables. For example, estimated time duration could be in days, months, and years. Cost could be dollars, pounds, yens and other currency. Depending on who is using the data, different views of the data could be provided to the user.

Appropriate access methods and index strategies have to be developed for the warehouse. For example, the warehouse is structured in such a way so as to facilitate query processing. An example query may be: how many red cars costing more than 50K were bought in 1995 by physicians? Many relations have to be joined to process this query. Instead of joining the actual data, one could get the result by combining the bit maps for the associated data. The warehouse may utilize an index strategy called a bit map index where essentially there is a 1 in the bit map if the answer is positive in the database. So, if the color of the car is red, then in the associated bit map, there will be a 1. This is a simple example. Current research is focusing

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on April 29,2010 at 17:03:30 UTC from IEEE Xplore. Restrictions apply.

on developing more complex access methods and index strategies. We need to examine the security impact on query processing strategies for the warehouse. For example, does query modification apply for secure warehousing? Suppose the user is not able to see the sales figures for those living in region X. Then the query has to be modified as follows: How many red cars costing more than 50K did physicians who do not live in region X buy in Detroit in 1995?

Developing an appropriate query language for the warehouse is an issue. This would depend on the data model utilized. If the model is relational, then an SQL-based language may be appropriate. We then need to examine extending SQL to specify security constraints such as User group A cannot see any information about the purchase of red cars by physicians from region X. One may also need to provide visual interfaces for the warehouse.

Secure database administration techniques may be utilized for administering the warehouse. Is there a warehouse administrator? What is the relationship between the warehouse administrator and the administrator of the data sources? How often should the warehouse be audited? Another administration issue is propagating updates to the database. In many cases, the administrators of the data sources may not want to enforce triggers on their data. If this is the case, it may be difficult to automatically propagate the updates. What is the security impact on update propagation? What are the functions of the Systems Security Officer (SSO) for the warehouse? Should there be a Warehouse Security Offer (WSO)?

Security solutions for integrating heterogeneous and federated database systems discussed in [THUR94] may be applied to secure data warehouses. For example, we need to examine the challenges for secure federated database management to integrate the security policies for data warehousing. Figure 2 illustrates security policy integration for data warehousing. We need to develop secure transformations as we move from one layer to the next in building a warehouse.

As stated earlier, statistical database security is one of the technologies for securing the data warehouse. Since the warehouse gives out sums and averages, how can one protect the sensitive values from which the sums and averages are computed? Security controls also have to be enforced in maintaining the warehouse as well. This will have an impact on querying, managing the metadata, and updating the warehouse. In addition, if multilevel security is needed, then there are additional considerations. For example, what are the trusted components of the warehouse?

High performance computing including parallel database management plays a major role in data warehousing. The goal is for users to get answers to complex queries rapidly. Therefore, parallel query processing strategies are becoming popular for warehouses. Appropriate hardware and software are needed for efficient query processing. We need to integrate security into parallel database systems. Some preliminary work was reported in [THUR93b]. We need to carry out further investigations.

Secure metadata management is another critical technology for data warehousing. The problem is defining the metadata. Metadata could come from the data sources. Metadata will include the mappings between the data sources and the warehouse. There is also metadata specific to the warehouse. We need to examine the security impact on metadata management. e. There are three types of metadata. One is metadata for the individual data sources. The second is the metadata needed for mappings and transformations to build the warehouse, and the third is the metadata to maintain and operate the warehouse.

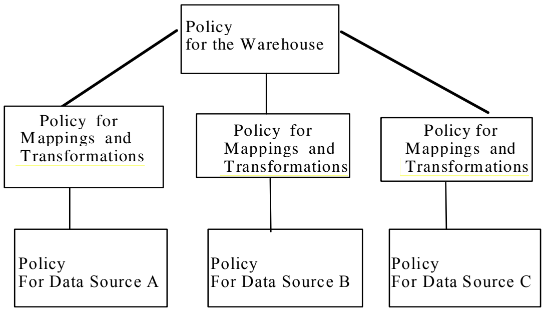

Figure2. Security Policy Integration

Secure distributed database technology discussed in [THUR91] plays a role in data warehousing. Should the warehouse be centralized or distributed? If it is distributed, then much of the technology for secure distributed database management discussed [THUR91] is applicable for data warehousing. In the non distributed case, there is a central warehouse for the multiple branches, say in a bank. In the distributed warehouse case, one may assume that each bank has its local warehouse and the warehouses communicate with each other.

2.3. Design Steps

Designing and developing the secure data warehouse is a complex process and in many ways depends on the application. A good reference to data warehousing is the book by Inmon [INMO93]. It describes the details of the issues involved in building a data warehouse. In this section we outline some of the steps to designing the secure warehouse. Figure 3 illustrates some of these steps. There are three phases to developing a secure warehouse. One phase focuses on structuring the secure warehouse so that secure query processing is facilitated. In other words, this phase focuses on getting the data out of the warehouse. Another phase focuses on bringing the data into the warehouse. For example, how can the secure heterogeneous data sources be integrated so that the data

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on April 29,2010 at 17:03:30 UTC from IEEE Xplore. Restrictions apply.

can be brought into the warehouse and yet security be maintained? The third phase maintains the warehouse once it is developed. This means the process does not end when the secure warehouse is developed. It has to be continually maintained. We first outline the steps in each of the phases.

One of the key steps in getting the data out of the warehouse is application analysis. For example, what types of queries will the users pose? How often are the queries posed? Will the responses be straightforward? Will the users need information like summary reports? A list consisting of such questions needs to be formulated. Furthermore, we need to examine the security constraints enforced by the warehouse and determine how these constraints may be enforced.

Another step is to determine what the user would expect from the warehouse. Would he want to deal with a multilevel relational model or a multilevel object-oriented model or both? Are multiple views needed? How can access be controlled to the views? Once this is determined, how do you go about developing a secure data model? Are there intermediate models?

A third step is to determine the metadata, index strategies, and access methods. Once the query patterns and data models have been determined, one needs to determine what kinds of metadata have to be maintained. What are the index strategies and access methods enforced? What are the security controls on the index strategies and access methods?

A closely related task is developing the various schemas and policies for the warehouse. Note that the individual databases will have their own schema and security policies. The complexity here is in integrating these schemas to develop a global schema for the warehouse. While schema integration techniques for distributed and heterogeneous databases may be used, the warehouse is developed mainly to answer specific queries for various applications. Therefore, special types of schemas such as star schemas and constellation schemas have been proposed in the literature. Products based on these schemas have also been developed. However we need to take a closer look at the schemas and examine the security impact. Furthermore we need to explore techniques to integrate the security policies.

There are several technical issues in bringing the data into the warehouse from the different data sources. What information should be deleted from the individual databases when the data is migrated to the warehouse? How should integrity be maintained? What is the security policy? How can inconsistencies be resolved? For example, we need to ensure that by querying the warehouse the sensitive information in the back-end databases is not revealed to the user who does not have proper access control to this information. This requires a lot of work. Various algorithms for integrating heterogeneous databases have to be examined. At the end of this stage, one would have some form of a secure warehouse. Multi-tier architecture is becoming popular for data warehousing. Essentially, data passes through multiple tiers before reaching the warehouse. We need to examine the security impact on multi-tier architecture. At the bottom tier are the data sources. At the top tier is the data warehouse. Between the top and bottom there may be multiple tiers. Each tier has its own schemas, metadata, and various administration details as well as policies, and each tier takes advantage of the work done at lower tiers.

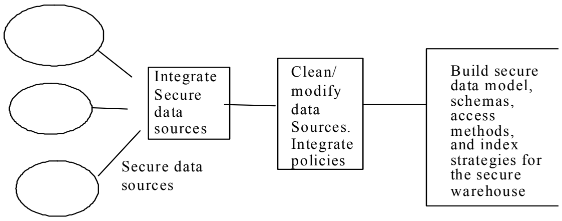
Integrate Secure data sources
Clean/ modify data Sources. Integrate policies
Build secure data model, schemas, access methods, and index strategies for the secure warehouse
Secure data sources

Figure 3. Developing a secure data Warehouse

Once the secure warehouse is designed and developed, there are also some additional considerations for maintaining the warehouse. How is the security of the warehouse maintained? Should the warehouse be audited? How often is the warehouse updated? How are the changes to the local databases to be propagated to the warehouse? What happens if a user's query cannot be answered by the warehouse? Should the warehouse go to the individual databases to get the data if needed? How can data quality and integrity be maintained?

We have outlined a number of phases and steps to developing a secure data warehouse. The question is, should these phases and steps be carried out one after the other or should they be done in parallel? As in most software systems, there is a planning phase, a development phase, and a maintenance phase. However, there are some additional complexities. The databases themselves may be migrating to new architectures or data models. This would have some impact on the warehouse. New databases may be added to the heterogeneous environment. The additional information should be migrated to the warehouse without causing inconsistencies. These are difficult problems and there are investigations on how to resolve them. Although there is much promise, there is a long way to go before viable commercial secure data warehouse products are developed.

In summary, a secure data warehouse enables different applications to view the data differently and at the same time enforce the security policies. That is, it supports multidimensional and multilevel data. Data warehouse technology is an integration of multiple technologies including heterogeneous database integration, statistical databases, and parallel processing. The challenges in data warehousing include developing appropriate data models, architectures (e.g., centralized or distributed), query languages, and access methods/index strategies, as well as developing techniques for query processing, metadata management, maintaining integrity and security, and

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on April 29,2010 at 17:03:30 UTC from IEEE Xplore. Restrictions apply.

integrating heterogeneous data sources. Integrating structured and unstructured databases, such as relational and multimedia databases, is also a challenge. Security cuts across all layers and all stages of the development.

While the notion of data warehousing has been around for a while, it is only recently that we are seeing the emergence of commercial products. This is because many of the related technologies such as parallel processing, heterogeneous database integration, statistical databases, and data modeling have evolved a great deal and some of them are fairly mature technologies. Furthermore secure database technology is also fairly mature. There are now viable technologies to build a secure data warehouse. We expect the demand for secure data warehousing to grow rapidly over the next few years.

It should be noted that many of the developments in data warehousing focus on integrating the data stored in structured databases such as relational databases. In the future we can expect to see secure multimedia data sources being integrated to form a warehouse.

3. Design of Extended RBAC for Secure Data Warehousing

3.1. Overview

Data warehousing needs end to end security because, the entire data warehousing environment is not just the database, it as an enterprise wide system where there is an operation system from which data is extracted, a transformation system which transfers the data to the data warehouse and possible distribution to various other data marts or end users. Data management and data mining is an integral part of any corporation. The efficient management of the data allows increasing the performance of the entire corporation. On the whole data warehousing is a very complex part, it has multiple data sources, numerous applications and various end-users. It has to be prevented from being hacked by illegitimate users, preventing the access of data by unintended users and protect the data from damages and modification by them.

There can be various organization involved in corporation, each organization should have a control over their data and should be able to release whatever they like to share it in the corporation. In the previous section we discussed in detail about the existing security components for data warehouses. Now in this section we will look into RBAC in Data warehousing, the issues in the existing model, new system design, advantages over the existing model, Implementation issues.

In this section we discuss the design and implementation of an extended RBAC model for secure data warehouse. First in section 3.2 we provide an overview of RBAC and describe its limitations, then in section 3.3 we discuss the Extended RBAC model as well as the design and implementation of a secure data warehouse, experiments and challenges.

3.2. Role Based Access Control for Enterprise Wide Data Warehousing

3.2.1. Overview

The corporations have strict laws to maintain privacy and confidentiality. Every organization is willing to spend enormous amount of money in identity management which is also an integral part of data warehousing. There are can be scenarios where the private data can be in the database that is being mined by different organizations in the corporation. The security model should be easy to implement, low in maintenance, should administrate access controls ensure network and data security. While RBAC can be challenging to design and implement, it can be tailored to a company's business model and security risk tolerance. Once implemented, it scales for growth and requires minimal maintenance.

Once all of the employee roles are populated into the database, role-based rules are formulated and workflow engine modules are implemented. Through these elements, role-based privileges can be entered and updated quickly across multiple systems, platforms, applications and geographic locations RBAC provides company wide control process for managing data and resources. RBAC systems also can be designed to maximize operational performance, maintain data consistency and integrity. They can streamline and automate many transactions and business processes and provide users with the resources to perform their jobs better, faster and with greater personal responsibility. With an RBAC system in place, organizations are better positioned to meet their own statutory and regulatory requirements for privacy and confidentiality, which is crucial for data management, as well as requirements imposed by external business partners and government agencies. Directors, managers and IT staffers are better able to monitor how data is being used and accessed, for the purpose of preparing more accurate planning and budget models based on real needs.

3.2.2. Issues with the Existing Model

In an Enterprise wide data warehousing system the security component that has been widely in use is the role based access control. It is a traditional access control model, with strong administrative security. Role based access control is capable of handling an entire system, but there are certain issues that have been identified in RBAC. The definition of users and groups are not clear. Duties are not defined along with the roles. The RBAC model mainly arises based on the roles that naturally exist in the system. It has certain limitation with respect to maintaining temporal dependencies. Incase if there is need for an order

Second International Conference on Availability, Reliability and Security (ARES'07)
0-7695-2775-2/07 $20.00 © 2007 IEEE

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on April 29,2010 at 17:03:30 UTC from IEEE Xplore. Restrictions apply.

of causality in some process the RBAC model will have a difficulty to find out which process has to occur first and which should continue or which one should be revoked. The decision process is very weak in RBAC. It does not take into consideration the pre-approvals needed for any process, it also does not have a component to decide the on-going approvals. The decision for any process is decided before the start of the process, which limits the system performance. Incase some process needs more privilege during the process or some object has to change its attribute say one user trying to extract multiple data source simultaneously and needs authorization of a data which was not approved at the beginning of the session, then the process has to be terminated and restarted again. It does not have mutable attributes. There are certain objects which need to change its attribute definition to support the smooth running of the system; such mutable attributes are not available in RBAC. To summarize the issues in using RBAC in Data warehousing are User/Group definition, Temporal Dependencies, Mutable attributes and Decision process.

3.3. ERBAC System Architecture for Enterprise Wide Data Warehousing

3.3.1. Overview of ERBAC

The existing systems mostly use RBAC which has more limitation with respect to resource management. It has many issues regarding decision process, multiple roles, multiple session and many other temporal dependencies. So the warehouse needs are not fully met only having RBAC has a security model. UCON is one of the modern security models which covers most of the traditional access model functionalities and has more new functions. UCON cannot exist alone and manage data security. It is not one for all complete solution. UCON has to co exist with some other traditional component in order to provide a strong secured data warehouse because it's a specific component which is mainly strong only in decision factor. The delegations of role are derived from traditional access control lists. The administrative functions are not as robust as in RBAC. So RBAC and UCON combination will form a strong access and usage control security component. This newly proposed component is E-RBAC (Extended Role Based Access Control). For details of RBAC and UCON we refer to [PARK04], [SAND96].

3.3.2. System Architecture Design for ERBAC

The system architecture of the Extended RBAC is similar to that of the existing security model. It is going to be a combination of the RBAC security component and UCON security model. The Administrative Security, Role delegations are part of RBAC component and the rights of objects and decision process is a combination of UCON. It will take into account the obligation approvals for pre and on-going transactions. It will also check the environment conditions before it gives the approval for authorization and the predicates of the authorization the obligations. The UCON model also helps in tracking the temporal dependencies there by helping to know if the current system can grant or revoke the operation that is going to be performed or that is already being performed. This component is not available in the existing role based access control. Combining both the models we get a secured system which encompasses both a highly secured administration rules. It also helps in the protection of identity management. The rights manager helps in checking the privacy constraints of the corresponding data.

In the architecture shown in figure 4 we see the architecture of the newly proposed system RBAC with UCON. The Administrator component provides the extension of role based access and usage control. The architecture also shows the imaginary division between the RBAC and UCON components. There are five managing components as shown in Figure 4: User Manager, Role Manager, Decision Manager, Session Manager and Data Manager. The administrator is the configuration controller which manages all the security components.

User manager takes care of the list of users in the database using the system. It interacts with the role manager and gets the corresponding role of the user for the respective session. The session manager helps in maintaining multiple sessions and maps the history of the user in each session. Each time the user requests for some data the UCON comes into effect. The decision manager comes into the context and checks for the condition of the current system and check if it is going to be consistent even after the request being served. The Obligations are checked and the rights for the corresponding data are checked for pre approval and on-going approval. When the predicate approval is done depending on the role of the user the authorization is done. If all these decision process are checked then the user request is serviced. Thus the role

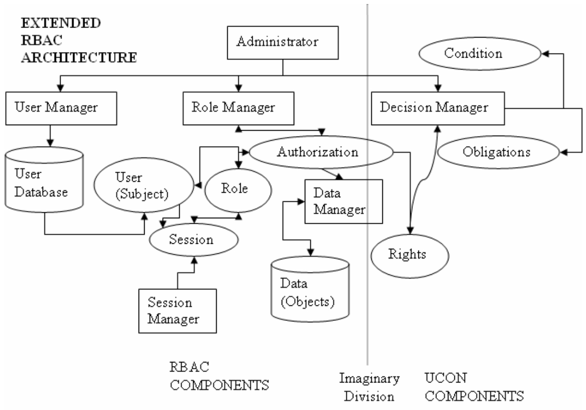

Figure4. ERBAC Architecture with UCON_{ABC} Extension

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on April 29,2010 at 17:03:30 UTC from IEEE Xplore. Restrictions apply.

based access and usage control is setup in a same system which has a strong administrative and decision process with temporal dependencies, mutability and identity management.

3.3.3 Implementation of ERBAC in Data Warehousing

Data warehousing is a complex system, it needs a proper security component that can ensure the safety of the entire system. The ERBAC model used in the data warehouse system mainly has an administrator component which manages the entire system security. The administrator component is linked from the beginning of a process through the end of the process. It clearly mentions the session details and also maintains history of each multiple session, multiple users in the same session and single user in the multiple sessions simultaneously.

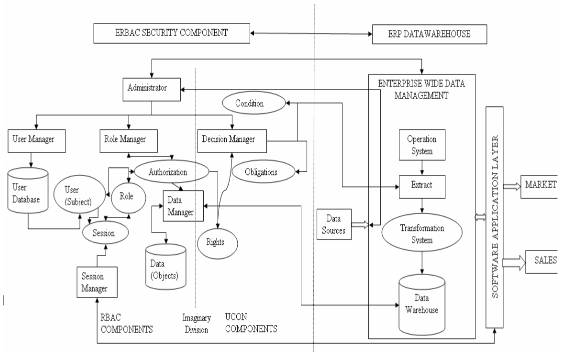

Figure5. ERBAC in Enterprise Wide Data Warehouse

In a typical data warehouse shown in figure 5 the data is supplied by various data sources. The ERBAC component helps for each of the data sources to define the set of users/groups, there access privileges. The release attributes rights, coalition access policy. Once the data sources are authenticated, they are available for the enterprise through a secure connection. The data sources provide their rights, policy, access privileges to the administrator components which in turn send it to the usage control component. The usage control component has a rights manager and manages policies for the individual data sources and the corporation.

Once a user wants to make use of the data in the enterprise, he has to login to have a secure session and to provide him with the history of the previous sessions that he has been involved. The process is managed by the session manager component of ERBAC. It also allows the user to participate in the existing session and single user in multiple sessions. Once the session is created the process of accessing the enterprise system is initiated, the operations system access the data source which in turn access the rights manager gets the pre- approvals needed for the corresponding process. If the pre-approvals are granted the process loads the required data and the object attributes required for the process based on the access level of the user and the usage limit on the data for the corresponding user.

Once the pre-approvals are authenticated the process begins and while the process is in progress there is a chance that some of the mutable attributes change the definition, in such cases the rights manager is accessed and the decision process also plays an important role, the rights manager checks if the corresponding attribute is allowed to change and if the change affects the system condition. If the system is consistent and the object is granted for the on-going approval, the process continues or if the grant is revoked then the process terminates.

Once the operations system gets the access it mines the data sources checks for the required pattern extracts and sends the data to the transformation system. The condition of the data and transformation are checked for system consistency, if it is consistent the system is sent to the warehouse. After the data is processed and stored in the warehouse it is distributed to the end-user through the application layer which acts as a wrapper enclosing the application, enterprise warehouse and the ERBAC security component.

To summarize the advantages of the ERBAC component, the roles of the RBAC gives a hierarchical use of the data in the system but does not provide usage control over the data. There are also problems in maintaining single user in multiple sessions. The UCON component provides the decision component which is stronger than the RBAC. It checks for the pre-approvals which are the pre-conditions that need to be satisfied for safe execution on some data. It checks for the system condition for every change that is made on the data and stops if the data is going to make the system state inconsistent. The on-going approvals for the mutable attributes and other object attributes whose usages limits are extended on some particular scenario are checked by UCON. This pre-approval, on-going approval and system condition check provides a strong authentication and clear decision process. So ERBAC provides a single unified framework for RBAC and UCON to exist together and provide a strong administrative component and a clear decision making system.

3.3.4 Experiment and Challenges

The security component designed above has been implemented in a simulated Data warehouse. The front end of the system is designed using Java, back end is designed using oracle 10G XE, The simulated system is an inventory system, and it helps in displaying the data management using the ERBAC component. The roles and groups are defined in the database by the administrator. The rights are also managed within the database using a rights attribute associated for each data. It clearly states the usage limit, owner for the data. If the limit needs to be changed, it

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on April 29,2010 at 17:03:30 UTC from IEEE Xplore. Restrictions apply.

requires approval from the owner of the data. In some cases the administrator can override the owner's rights if the data is inconsistent or might damage the consistency of other data. The rights of the data can be mentioned by the owner and has all the rights to grant or revoke the access whenever he wants. So even if the data is granted to other user and it is in use the data can be revoked from continues access. The administrator will ensure the stable condition of such on-going approval or on-going revocation.

Application simulated will generate scenarios where in it can show case the list of pre-approvals needed for the execution of a process and incase it needs an on-going approval it request for the approval to the administrator or the data owner. Apart from the rights that are mentioned in the database, there is a data policy manager encoded in xml format which acts as a data layer. It interacts between the application and the database and manages the rights of the data. The rights manager gives a list of pre-approvals needed for executing the process. The process continues until the on-going approvals are granted, there are some mutable attributes which can be loaded while the process is executed.

Here we discuss some of the challenges and issues faced during the implementation of such data warehouse in an Enterprise wide data Warehouse. The data rights and usage limits should be specified clearly. The Role and Rights should not conflict with each other. The management of mutable attributes increases the process time. The on-going approval increases the cost of the query. The process is slowed when there are some objects loaded during the course of the process. This can be solved to an extent if the course of the process and the attributes that are needed are known, a knowledge engine can be maintained which can ensure to load all the attributes at the pre-approval state.

4. Summary and Directions

This paper has discussed secure data warehousing. We started with a discussion of a definition for a secure data warehouse, the technologies for a secure data warehouse, functions of a secure data warehouse, and issues on developing a secure data warehouse. Key concepts in secure data warehousing include developing a secure data model, security architecture, and access methods and index strategies. We then described the design and implementation of a secure data warehousing system based on the Extended RBAC model.

While some progress has been made on secure data warehousing and we are seeing commercial products incorporate some security features, there is still a lot to do. We need to develop ways to integrate security policies in building a warehouse. We also need a thorough investigation of the security issues in both building a warehouse as well as extracting data from the warehouse. We need to examine the security impact on integrating data mining with data warehousing. Finally we need to examine the inference problem and privacy problem that arise due to data warehousing and data mining. Some discussions on data mining, security, the inference problem and the privacy problem are given in [THUR02b].

References

[BEST02] Bestougeff , H., et al, (Editors) Heterogeneous Information Exchange and Organizational Hubs, Kluwer, MA, 2002.

[INMO93] Inmon, W., "Building the Data Warehouse," John Wiley and Sons, NY, 1993.

[PARK04] Jaehong Park and Ravi Sandhu. "The UCON$_{ABC}$ Usage Control Model." *ACM Transactions on Information and System Security*, Volume 7, Number 1, February 2004.

[SAND96] Ravi Sandhu, Edward Coyne, Hal Feinstein and Charles Youman, "Role-Based Access Control Models." *IEEE Computer*, Volume 29, Number 2, February 1996.

[THUR91] Thuraisingham B., Secure Distributed Database Systems, Computers and Security, December 1991.

[THUR93a] Thuraisingham, B. Towards a Standard Multilevel Relational Data Model, Computer Standards and Interface Journal, 1993.

[THUR93b] Thuraisingham, B., Parallel Processing and Trusted Database Management Systems, Proceedings of the ACM Computer Science Conference, Indianapolis, IN, February 1993.

[THUR94] Thuraisingham, B., Security Issues for Federated Database Systems, Computers and Security, Volume 13, #6, 1994.

[THUR97] Thuraisingham, B., Data Warehousing, Data Mining and Security, Presented at the IFIP Database Security Conference, Como, Italy, 1996 (paper published in formal proceedings by Chapman and Hall, 1997).

[THUR05] B. Thuraisingham, Database and Applications Security, CRC Press, 2005

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on April 29,2010 at 17:03:30 UTC from IEEE Xplore. Restrictions apply.