A Security Concept for OLAP

Remzi Kirkgöze, Nevena Katic, Mladen Stolba, A Min Tjoa

Institute of Software Technology (IFS)
{remzi, katic, stolba, tjoa}@ifs.tuwien.ac.at
Vienna University of Technology
Resselgasse 3, A-1040 Vienna, Austria

Abstract

A data warehouse collects and integrates data from multiple, autonomous, heterogeneous sources with the purpose of efficiently implementing decision support or OLAP queries. Much work in data warehousing has been performed on view materialization and data integration; we focus on access and security management in OLAP and N-dimensional cube. Since data in data warehouse are valuable and an important cooperate resource, we define a security model for data warehouses which describes security constrains for roles in the data warehouse. Each user in the data warehouse has a role and each role has a security constrain list that builds the security profile of the role. According these role profile the user is authorized to query data from the data warehouse

1. Introduction

During the early stages of database security research the main focus was directed to the discretionary aspect of database security, namely to different forms of access control lists and to view-based protection issues ([16], [13], [14], [11]). Later the focus has moved to mandatory controls, integrity issues and security mechanisms which are designed to provide privacy ([2], [3], [4], [5]). The major current trends are to provide tools that support the designer during the different phases in the design of databases with security critical contents, to develop security semantics and classification constraints, to investigate the use of rules and triggers for various problems related to database security, to extend security issues to other data models and distributed and heterogeneous databases, and to investigate in physical design questions like transformation and recovery management as well as in storage structures developed with main focus on the support for security.

Information stored in data warehouse is often considered as valuable and important corporate resource. Many organizations have become so dependent on the proper functioning of their systems that a disruption of services or a leakage of stored information may cause outcomes ranging from inconvenience to catastrophe. Organization data may relate to financial records others may be essential for the successful operation of an organization, they may represent trade secrets or may describe information about persons whose privacy must be protected. For example health-care systems must provide patient information instantaneously to physicians who need it. But the same system should not permit unauthorized access to the data. As the degree of summarized data in data warehouse increases, the value of the data becomes increasingly higher. Summarized information that can assist an enterprise in making decisions is just as valuable to the competition. Controlling security and access to the data inside the data warehouse is still an evolving area of technology. Although there has been a big interest for data warehousing in last years, the area of the security and access control in data warehouse has stayed untouched until recently.

Since the data warehouse does not manage mission critical operational data, the nature of the security threat is not of causing damage to data but of disclosing corporate secrets and strategies [12]. Countering this threat involves containing access on a need-to-know basis. The security policy must also provide restrictions on drill-down capabilities and access control of specific summarized data tables and operational detail. Permissions must be also be managed for restrictions of resource usage such as ability to create temporary tables and ad hoc queries. The access control and security issue in data warehouse is also complicated by a number of factors [15]:

- The data warehouse is built primarily as an open collection of enterprise data. It can assist in decision

0-8186-8147-0/97 $10.00 © 1997 IEEE

making and can be used by analyst and operational staff in improving their operations and deriving strategic and sustained competitive advantage. The addition of security controls is against the need to be open.

- Users access data inside the data warehouse at different levels of summarization. The same user can start with highly summarized data and drill-down progressively into increasingly detailed data. Other users can operate at single level of summarization. It is difficult to manage access at the table and row level for data for each of these users.
- The nature of OLAP and data access tools in the data warehouse arena has been exploratory. Most users use the data warehouse by employing a discovery process. The addition of cumbersome security controls can make this very frustrating as users are prevented from processing further in their exploration.

Other threat scenarios are nuisance scenarios where hostile users tie up large amounts of resources -essentially making the data warehouse unavailable. Managing runway queries, creation of temporary tables and applying resource limits to user profiles can begin to address these challenges. The design of an access control and security plan is therefore an essential activity in the development process. Because of the nature of client/server applications, managing security from a single point of control is difficult. Users have user Ids and passwords which are often different for their workstation, network access, remote login into the server and remote login into one or more databases [12]. When a user departs a clean up crew has to remove access controls in a number of systems. Planning applications that manage the cleanup and removal of these multiple access controls can assist the task of managing security within the data warehouse.

In this paper we present a rule based user role profile to manage the security and access issues in a data warehouse environment. The data warehouse data model is represented by a star-schema which represents data in a central fact table with related dimension tables. The unique keys of each dimension table make up a compound key in the fact table. This representation can be seen as a cube. The hole data ware house is an N-dimensional huge cube. With our security model we try to define security constrains for each role in the data warehousing environment, which means each role defines a sub-cube of the data warehouse's N-dimensional cube. In the next section we give a summary of common security models; in section three we describe the main features of On Line Analytical Process (OLAP) and in section four we present our security model for a data warehouse environment and OLAP.

2. Database security models

Database security is concerned with ensuring the secrecy, integrity and availability of data stored in a database. Secrecy means the protection of information from unauthorized disclosure either by direct retrieval or by indirect logical inference. Secrecy must deal with the possibility that information may also be disclosed by legitimated users acting as an information channel by passing secret information to unauthorized users. Integrity is the protection of the data from accidental modification including the insertion of false data, the corruption of data and the destruction of data. Integrity constraints are rules that define the correct state of a database and thus can protect the correctness of the database during operation. Availability means to ensure data being available to authorized users when they need them.

In the following part we will summarize most common database security models. An extensive overview of these concepts can be obtained from [6] and [17].

2.1. Discretionary security models

Discretionary security specifies the rules under which subjects can create and delete objects, grand and revoke authorizations for accessing objects to others. They are fundamental to operating systems and DBMSs. Discretionary access controls (DAC) are based on the concepts of a set of security objects O, a set of security subjects S, a set of access privileges T defining what kind of access a subject has to a certain object, and in order to represent content-based access rules a set of predicates P. Applied to relational databases O is a finite set of values representing relational schemas, S is a finite set of potential subjects representing users, groups of them or transactions operating on behalf of users. Access type privileges are the set of database operations such as select, insert, delete, update, execute, grant or revoke. The tuple $<o,s,t,p>$ is called access rule and a function f is defined to determine if an authorization $f(o,s,t,p)$ is valid or not:

$f: O \times S \times T \times P \rightarrow \{\text{True, False}\}$

For any $<o,s,t,p>$, if $f(o,s,t,p)$ evaluates into True, subject s has authorization t to access object o within the range defined by predicate p.

Most systems supporting DAC store access rules in an access control matrix. In its simplest form the rows of the matrix represents subjects, the columns represents the objects and the intersection of a row and a column contains the access type that subject has authorization for with respect to the object.

Discretionary security is enforced in most commercial DBMS products and is based on the concept of database views. Instead of authorizing a user to the base relations of a system the information of the access control matrix is used to restrict the user to a particular subset of the data available.

2.2. Mandatory security models

While discretionary models are concerned with defining, modeling and enforcing access to information mandatory security models are in addition concerned with the flow of information within a system. Mandatory security requires that security objects and subjects are assigned to certain security levels represented by a label. The label for an object o is called its classification (*class(o)*) and a label for a subject s is called its clearance (*clear(s)*). The classification represents the sensitivity of the labeled data while the clearance of a subject its trustworthiness to not disclose sensitive information to others. A security label consists of two components: a level from a hierarchical list of sensitivity level or access classes (for example: *top secret* > *secret* > *confidential* > *unclassified*) and a member of a non hierarchical set of categories representing classes of object types of the universe of discourse.

Mandatory access control (MAC) requirements are formalized by two rules. The first one protects the information of the database from unauthorized disclosure and the second one protects data from contamination or unauthorized modification by restricting the information flow from high to low.

1. Subject s is allowed to read data item d if *clear(s)* >= *class(d)*.
2. Subject s is allowed to write data item d *if clear(s)* =< *class(d)*.

2.3. Adapted mandatory access control model

Adopting mandatory access controls to better fit into general purpose data processing practice and offering a design framework for database containing sensitive information are the main goals of the Adapted Mandatory Access Control (AMAC) model. Adapted mandatory security belongs to the class of role-based security models which assume that each potential user of the system performs a certain role in the organization. Based on their role users are authorized to execute specific database operations on a predefined set of data.

The AMAC security constraints are handled during database design as well as during query processing. During database design they are expressed by the database decomposition while during query processing they are enforced by the trigger mechanisms.

2.4. Personal knowledge approach

The personal knowledge approach is focused on protecting the privacy of individuals by restricting access to personal information stored in a database or information system. The main goal of this security technique is to meet the right of humans for informational self-determination as requested in constitutional laws of many countries. In this context, privacy can be summarized as the basic right for an individual to choose which elements of his private life may be disclosed.

The approach is built on the assumption that a person represented in the database knows everything about himself and if he wants to know something about someone else represented in the database that person must be asked. Knowledge about different persons can not be stored permanently and therefore must be requested from the person whenever the information is needed.

2.5. Clark and Wilson model

This model was first presented in [7]. It is based on concepts that are already well established in the pencil-and-paper office world. These are the notion of security subjects, security objects, a set of well-formed transactions and the principle of separation of duty. With other words, the users of the system are restricted to execute only a certain set of transactions permitted to them and each transaction operates on an assigned set of data objects only.

3. On line analytical processing

On Line Analytical Processing (OLAP) is an analytical processing technology which creates new business information from existing data, through a rich set of business transformation and numerical calculations. It is based on a multidimensional view of the business data in the data warehouse and presents a multidimensional logical view of the data. The view is independent of how the data is stored. OLAP is also referred to as Analytical Processing or Dimensional Analysis.

An analyst's view of the enterprise's universe is multidimensional in nature. Accordingly, the analyst's conceptual view of OLAP model is also multidimensional. These multidimensional conceptional schema or user view facilitates model design and analysis, as well as inter and intra dimensional calculations through a more intuitive analytical mode. Accordingly, user is able to manipulate such multidimensional data models more easily and intuitively than is the case with the single dimensional models. Multidimensional structure offers a good conceptual fit with the way end-users visualize business data.

Most business people already think about their business in multidimensional terms.

Business data as a matter of fact, is multidimensional. It is interrelated and usually hierarchical. In multidimensional analysis data is represented as dimensions such as product, geography. Dimensions are related in hierarchies, for example, city, state, region, country and continent. Time dimension is a standard dimension with its own hierarchy, such as day, month, quarter, year. [1]

In OLAP data is stored in arrays. These arrays are a logical representation of the business dimensions. This multidimensional array structure represents a higher level of organization. The structure itself contains much valuable intelligence regarding the relationships between the data elements because business analyst's perspectives are imbedded directly in the structure as dimensions as opposed to being placed into fields. The Figure 1 displays three dimensional view of the Sales data. This representation is also called data cube.

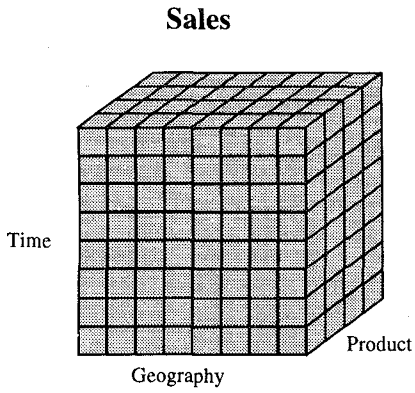

Figure 1: Data Cube

Users typically view the data as multidimensional cubes. Each cell of the data cube is a view consisting of an aggregation of interest, like total sales. The values of many of these cells are dependent on the values of other cells in the data cube. Users of data warehouses work in a graphical environment and data are usually presented to them as a multidimensional data cube whose one, two or even higher dimensional sub cubes they explore trying to discover interesting information. The values in each cell of this data cube are some business data measures of interest.

As an example consider a sales retail company. The operation data of the company is stored in a data warehouse. There are three dimensions we are interested in: *Time*, *Product* and *Geography*. The business measure of interest is the total *Sales*. So far each cell *(t, p, g)* in this 3-D data cube, The total sales and the quantity of product *p* that was sold in geography *g* in time period *t* are stored. Dimensions have the following hierarchy and the lowest granularity level of the data is represented first.

Time: *Day → Month → Quarter → Year*
Geography: *City → State → Region → Country → Continent*
Product: *Product → Product Line*

The business measure of interest, fact table, *Sales* with above dimensions has the following form:

Sales(TimeID,GeographyID,ProductID,$ sales,Qty)

Sales fact table with dimensions *Time*, *Geography* and *Product* will be used as example throughout this paper.

Users are interested in navigating through the data cube and dimension hierarchies and consolidated sales. For example what is the total sales of a given product to a given time period (t, p, ALL) or what is the total sales of a given product to a given time period to a given country (t, p, Country(g)).

In the following part we will give a summary of some data cube features.

3.1. Slice and dice

A data cube allows the end user to quickly range in on the exact view of the data required. This operation is also called *slice and dice* because the data is scoped down to a subset grouping. The reduced data cube can now be rotated and used in computations just as its larger parent was.

The end user may want to determine how sales in cities LA, NY, WA in days 2 and 4 of the month for products PC and TV was. Through the slice and dice operation the end user select desired positions along each dimension:

- for the *time* dimension day 2 and 4
- for the *geography* dimension WA, NY and WA
- for the *product* dimension PC and TV.

This is illustrated in Figure 2.

Because subset of the selected cube is derived from the data cube to which the current user is authorized to access, it is not very relevant for any security consideration. However the derived subset is a relevant object for security depending on actions taken.

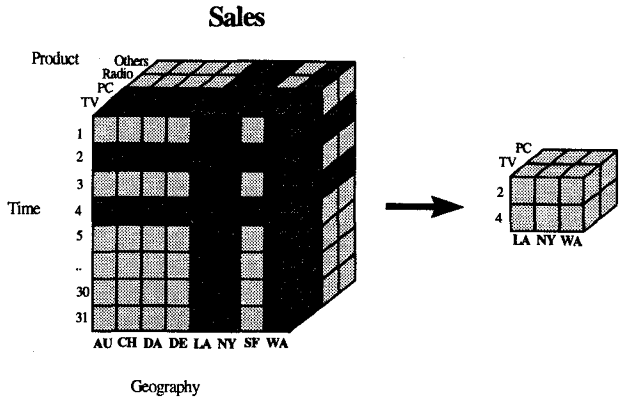

Figure 2: Slice and dice in a Data Cube

3.2. Drill-down, roll-up

In a business environment dimensions are hierarchical and there are multiple levels within a hierarchy. Hierarchies allows for very quick data manipulations and detailed analysis along different levels within all the dimensions of a data cube. Moving up and moving down levels in a hierarchy is referred to as roll-up and drill-down.

Each view, as defined by the chosen levels along each dimension, can then be rotated or sliced and diced. A user may first want to view sales at the city level and then may want to view the data region level. With drill-down and roll-up, the business and user has a free hand to navigate along the dimensions to see detailed or summarized data.

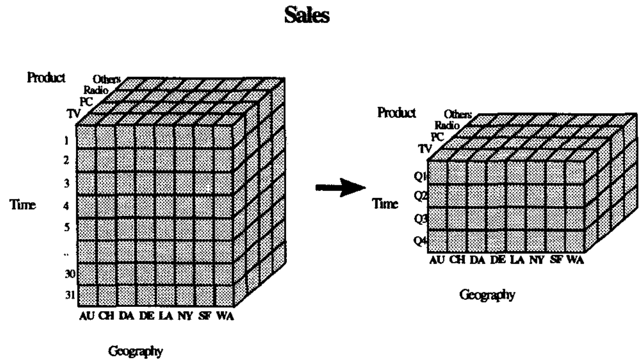

Figure 3: Drill-down and role-up in a Data Cube

For example, if the end user has a view of the data cube in basic dimension levels (*day, city, product*) and wants to see the data in *time* dimension quarterly. In order to achieve this, the aggregation of *time* dimension on *quarter* should be taken. Figure 3 shows this operation.

Due to nature of the business the detailed or summarized information derived by drill-down/roll-up can be very valuable for the organization and should be protected against any access attempt of unauthorized users. With the conventional database security models authorizations are defined only on the base data and definition of authorization on summarized data is not possible. For example in an organization a manager can access data in his region for any time period but not authorized to get data for the hole country in a selected period. This can not be expressed with current security models. In order to protect valuable company data against unauthorized users the OLAP security should also consider authorizations on aggregated and detailed data.

4. OLAP / Data Warehouse security model

The security problem of a data warehouse can be seen as a problem of DBMS that manage the data warehouse. Indeed the DBMSs that will house the data warehouse data do have their own built in security. The problem is that DBMS security is view based and designed for the operational environment, not the decision support systems. In view based security the data warehouse administrator or security administrator defines what data can be seen and manipulate through a view defined by the administrator. In this security concept there is the assumption that the DBA knows what is going to be done with the data once it is accessed. The view is then created to accommodate the known activity that will be done. In a Decision Support System(DSS) or OLAP environment we can control what the user is going to do with the data that has been retrieved from the data warehouse. The very essence of DSS is the lack of knowledge what will occur once the end user has retrieved the data from the data warehouse.

Another problem with view based DBMS security in the data warehouse environment is that the view based security could easily be bypassed. An end user merely has to do a direct disk dump of data and reformat the data to find out what the contents are. Views can not prevent an interpretation or aggregation of data once the data is dumped outside the control of DBMS.

The security model for the data warehouse should support all features of the OLAP concept described in section three. Definition of security constraints based on the granularity level is not sufficient for the data warehouse. It should be extended to support hierarchies of relations and roles played by the user. If we take *Time* as dimension with hierarchies *day→month→quarter →year* with the granularity *day*, the definition of access control for the granularity level will not be sufficient for the access control because of aggregations in *time* dimension. If a user of a conventional system is authorized to access only to *yearly* and *monthly* data and there is no pre-calculated tables for these aggregation levels, the user will not be able to access the required data because he is not authorized to access daily data from which the summary data is calculated. In an OLAP environment to access aggregated data even though the base data is not authorized to be accessed should be made possible.

We propose a security model for data warehouse which has only roles as the security subjects. That means authorizations can only be granted to roles. A role is regarded as a job describing what has to be done regardless of who does it. Roles should have exactly those authorizations that are needed to fulfill the duties of the job. Users are existing persons working in the system. Each user in the data warehouse should have at least one role assigned, but he can have several roles. A user can play only one role at one time. This policy prevents authorization conflicts among the roles of a user and it seems to be no limitation to real-life situations as long as users can easily change the role they want to play.

Security objects are the passive entities of a security system that contain and receive information to be protected. In an OLAP environment these are dimension and fact tables and their attributes. Each security subject is authorized to a kind of action over security object. This action is defined as access type. Because of most OLAP data for the end user is read only in this concept we will have only one access type which is *Read.*

The authorizations are presented as rules to the OLAP. We use discretionary access controls(DAC) model that are based on a collection of concepts, including a set of security subject (S), a set of access types (A), and a set of security objects (O). In general, a security rule is a quadruple, (s,a,o,p), where subject *s* has the access type *a* to access security object *o* within the range of predicate *p*. Predicates are used to additionally restrict access to security objects dependent on several constraints.

Basically we have three different predicates to express security constraints.

1. ***Simple predicate (SP):*** SP(S,A,O), where S is the security subject, A is the access type and O is the security object under consideration.
2. ***Simple attribute predicate (SaP):*** SaP(S, A, O, Attr.), where S is the security subject, A is the access type, O is the security object and Attr. is the attribute of the security object under consideration.
3. ***Value based attribute predicate (VBaP):*** VBaP(S, A, O, Attr., Theta, V), where S is the security subject, A is the access type, O is the security object, Attr. is the attribute of the security object under consideration, Theta is the comparison operator, and V is the comparison value.

In order to grant any access right to a role any combination of above described rule definition types can be used. The list of these rules defines a subset of the data cube of the OLAP to which the role (security subject) is authorized to access. The rule definition can be done by the system manager or by the owner of the security object. The derived data cube of the role has its own dimensions and dimension hierarchies which are also a sub-set of the not restricted dimensions and dimension hierarchies. For example if we define some access rules that authorize to access only daily and yearly data of dimension *time* for a role. The dimension hierarchy of *time* for this role will be *day→year* instead of *day→month→quarter→year.*

In order to express the security requirements defined by means of the rules a decomposition of security objects into single level fragments is performed. The decomposition is based on the rule structure and results in a set of fragmental schemas in a way that no rule is defined over a subset of a resulting schema only. The decomposition is performed by using a vertical, horizontal, or derived horizontal fragmentation policy [17].

A *vertical fragmentation (vf)* (▬▬▬) is the projection of a relational schema (RS) into subset of its attributes. To keep the fragments lossless, the key is always present in each vertical fragment. A *horizontal fragmentation* (hf) (————) is partitioning a RS into disjoint fragments based on a predicate defined on rules. The predicate is expressed as a Boolean combination of terms, each term being a simple comparison that can be established as true or false. A *derived horizontal fragmentation (dhf)* (............) is partitioning a relational shame RSi by applying to it the same partitioning criterion as applied to RSj (i!=j). In the case of *dhf* the set of selection attributes is not subset of attributes of RSi. To perform dhf a relationship between RSi and RSj must exist [17], [6].

The sub-cube Ci ($Ci \in V$) defined on OLAP data represents the area of the data cube to which a corresponding role has access. Let F ($F = Ci \cap Cj$) be a fragment then F represents the area of the data cube to which two roles have access in common. If $F = Ci \setminus Cj$ the F is only accessible by roles having sub-cube Ci as their interface to the data cube. In this case, F represents data which is not contained in Cj and must therefore not be accessible for the corresponding role.

5. Example

The above defined security concept is illustrated in the following example. We will first define our rules and based on the rules the fragmentation policy is applied to the rules. Assuming an organization has the *Sales* business measure in its data warehouse and requires to represent the following roles for the security considerations.

- *Role1* has full access to all data represented in Sales fact table and its dimensions. The rules for *Role1*:

  ⇒ SP(Role1, Read, Sales)

⇒ SP(Role1, Read, Time)

⇒ SP(Role1, Read, Product)

⇒ SP(Role1, Read, Geography)

- *Role2* has access to daily and monthly data in cities only for product "mouse" of Sales business measure. The rules for *Role2*:

⇒ SP(Role2, Read, Sales)

⇒ SaP(Role2, Read, Time, Day)

⇒ SaP(Role2, Read, Time, Month)

⇒ SaP(Role2, Read, Geography, City)

⇒ VBaP(Role2, Read, Product, ProductName, "=", "mouse")

- *Role3* has access to daily data only in city "Vienna" where $ sales are less then USD100 of Sales business measure. The rules for *Role3*:

⇒ VBaP(Role3, Read, Time, Sales, $ sales, "<", 100)

⇒ SaP(Role3, Read, Time, Day)

⇒ VBaP(Role3, Read, Geography, Region, "=", "Vienna")

⇒ SP(Role3, Read, Product)

- *Role4* has access to yearly data only for all product lines. The rules for *Role4*:

⇒ SP(Role4, Read, Sales)

⇒ SaP(Role4, Read, Time, Year)

⇒ SP(Role4, Read, Geography)

⇒ SaP(Role4, Read, Product, ProductLine)

After security policy rules has been applied, a matrix is generated which represents all fragments generated and their associations to each role of the OLAP. The assignments of the roles to the fragments will be as following:

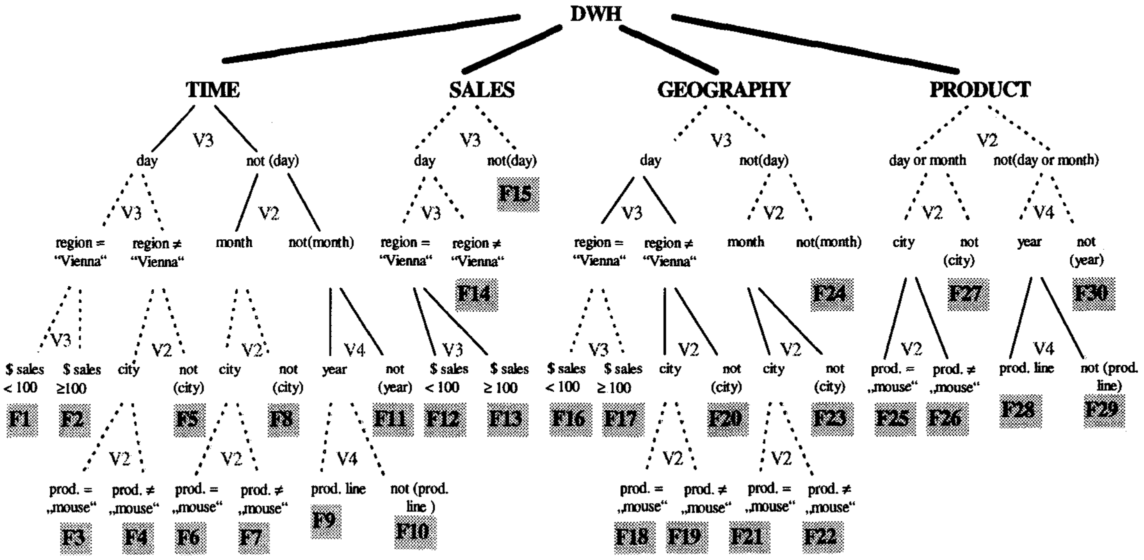

Figure 4: Example of a fragmentation for OLAP

F1={Role1,Role3}
F2={Role1}
F3={Role1,Role2}
F4={Role1}
F5={Role1}
F6={Role1,Role2}
F7={Role1}
F8={Role1}
F9={Role1,Role4}
F10={Role1}
F11={Role1}
F12={Role1,Role2,Role3, Role4}
F13={Role1,Role2, Role4}
F14={Role1,Role2, Role4}
F15={Role1,Role2, Role4}
F16={Role1,Role3,Role4}
F17={Role1, Role4}
F18={Role1,Role2,Role4}
F19={Role1, Role4}
F20={Role1,Role4}
F21={Role1, Role2, Role4}
F22={Role1,Role4}
F23={Role1, Role4}
F24={Role1,Role4}
F25={Role1, Role2, Role3}
F26={Role1,Role3}
F27={Role1, Role3}
F28={Role1, Role3, Role4}
F29={Role1, Role3}
F30={Role1, Role3}

The assignments of the fragments to the roles will be as following:

Role1={F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,F11,F12,F13, F14,F15,F16,F17,F18,F19,F20,F21,F22,F23, F24,F25,F26,F27,F28,F29,F30}
Role2={F3,F6,F12,F13,F14,F15,F18,F21,F25}
Role3={F1,F12,F16,F25, F26, F27, F28, F29,F30}
Role4={F9,F12,F13,F14,F15,F16,F17,F18,F19,F20,F21, F22,F23, F24,F28}

The fragmentation is shown in Figure 4. For example if a user plays Role2 and wants to query monthly data for product "mouse". The security system will generate a list of fragments that are needed to satisfy the query and this list will be compared with the list of the fragments that Role2 has access rights. If all fragments needed to satisfy the query is in the list of the Role2 fragments, the query will be performed. If not, the user is not authorized to retrieve some data from data warehouse, so an empty result is returned.

6. Conclusions

In this article we proposed a security approach based on adapted mandatory access control for OLAP - cubes. The advantage of this kind of security handling is its flexibility of assigning roles to different virtual sub-cubes. Hence it is quite straight forward to assign a number of roles to one particular person (or group of persons) without losing consistency with respect to the security policy.

This paper provides a first step toward the realization of an AMAC-based security concept for OLAP and data warehouses.

Further research is necessary in investigating multiple hierarchies in the dimensions of the OLAP-cube (e.g. {day -> week -> year} versus {day -> month -> quarter -> year}).

An in-depth analysis of performance and usability issues is very relevant and only possible as early as this approach is tested and implemented. Two of the authors (N. Katic, M. Stolba) are working on the implementation of this approach.

7. References

[1] R. Agrawal, A. Gupta, S. Sarawagi, *Modeling multidimensional databases*, IBM Research Report, Almaden, USA, 1996

[2] D.E. Bell, L.J. LaPadula. *Secure Computer System: Unified Exposition and Multics Interpretation.* Technical Report MTR-2997, MITRE Corp. Bedford, Mass., 1976.

[3] K.J. Biba. *Integrity Considerations in Secure Computer Systems.* ESD-TR-76-372, USAF Electronic Systems Division, 1977.

[4] J. Biskup, H.H. Brüggemann. *The Personal Model of Data: Towards a Privacy Oriented Information System.* Computer Security, Vol. 7, North Holland, 1988.

[5] J. Biskup, H.H. Brüggemann. *The Personal Model of Data: Towards a Privacy Oriented Information System.* Proc. 5th Int. Conf. on Data Engineering (ICDE'89), 1989.

[6] S. Castano, M. Fugini, M. Mertella, P. Samarati. *Database Security*, Addison Wesley, 1992.

[7] D. D. Clark, D. R. Wilson. *Comparison of Commercial and Military Computer Security Policies.* IEEE Security and Privacy Symposium, 1987.

[8] D. Bulos, OLAP Database Design: *A New Dimension in Database Programming and Design,* Vol. 9, No. 6 June 1996 32-37

[9] W. Essmayr, F.Kastner, G.Pernul, S.Preshuber, A.M. Tjoa, *Authorization and Access Control in IRO-DB*, Proc. of Int. Conf. on Data Engineering (ICDE'96), Luisiana, USA, 1996.

[10] W. Essmayr, F.Kastner, G.Pernul, S.Preshuber, A.M. Tjoa, The Security architecture of IRO-DB, Proc. 12th of IFIP Int. Conf. on Information Security (IFIP/SEC'96), Greece, USA, 1996.

[11] E.B. Fernandez, R.C. Summers, C. Wood. *Database Security and Integrity.* Addison-Wesley, Reading, MA, 1981.

[12] H.S. Gill, P.C. Rao, *Computing Guide to Data Warehousing,* Que Corporation, 1996.

[13] G.S. Graham, P.J. Denning. Protection Principles and Practices. Proc. AFIPS Spring Joint Conf. 1972.

[14] M. A. Harrison, W. L. Ruzo. Protection in Operating Systems. Comm. of the ACM Vol. 19(8), 1976.

[15] W.H. Inmon, J.D. Welch, K.L. Glassey, *Managing the Data Warehouse*, Wiley Computer Publishing, 1997

[16] B.W. Lampson. *Protection.* Conf. on Information and Systems Sciences, 1971.

[17] G. Pernul. *Database Security.* Advances in Computers, Vol. 38, p. 1-72, 1994.