Design of secure distributed medical database systems

Marie Khair[1], Ioannis Mavridis [2] and George Pangalos[2]

[1] Dept of Computer Science, Faculty of Natural and Applied Sciences, Notre Dame University, Louaize, Lebanon
mkhair@ndu.edu.lb
[2] Informatics Lab., Computers Div., Faculty of Technology, Aristotle University of Thessaloniki, 54006, Greece
imav@eng.auth.gr, gip@eng.auth.gr

Abstract. Security is an important issue in health care environments where large amounts of highly sensitive personal data are processed. It is therefore important that both the technical considerations and the security requirements (availability, integrity and confidentiality) are taken into account as main design objectives when designing a distributed medical database system. The aim of this paper has been to describe a step-by-step methodology for the design of a secure distributed medical database system. The methodology is based on the combination of mandatory and discretionary security approaches and uses hierarchies of user roles, data sets and sites in order to decide the secure distribution of the application. An experimental implementation of the proposed methodology in a major Greek hospital has shown the usefulness of the proposals as well as their effectiveness in limiting the unauthorized access to the medical database, without severely restricting the capabilities of the system.

1 Introduction

The design of a distributed database usually takes into account several objectives; for example the processing locality, the availability and reliability of distributed data, the workload distribution and storage costs versus availability, etc. However, using all these criteria at the same time is usually very difficult since it often leads to complex optimization models. So the trend has been to take only the most important objectives into consideration, for example maximizing processing locality, and to consider the other features as constraints ([3]). During the design of distributed database systems, parameters related to performance and reliability is usually taken into account in order to decide the allocation and replication scheme. Security is however another important issue which must also be taken into account from the very first steps of the design process, especially in security critical environments as is the health care one ([5,8]). For example, while replication seems to be generally advantageous because it can localize the access to the data, if not well handled it may create several problems related to the preservation of the confidentiality of the data residing on the sites.

The proposed design methodology takes into account and handles all three components of security (availability, integrity, confidentiality). Although adding yet another objective may further increase the complexity of the design procedure, this is necessary, especially in health care environments where large amounts of highly sensitive personal data are to be processed. The security problem becomes even more important when the distributed system includes mobile sites ([7]).

2 The Fragmentation and Allocation problem

As opposed to the case of centralized databases, the design of a distributed database usually includes two additional important design decisions: the fragmentation of the data and the allocation and replication of the fragments. During fragmentation data are divided horizontally, vertically or mixed in sets, so that the fragments can be allocated at different sites. A fragment consists of a group of tuples (in case of horizontal fragmentation) or attributes (in case of vertical fragmentation) which have the same properties from the viewpoint of their allocation ([3]). The correctness of the fragmentation depends on the disjointness (no fragment overlap), completeness (every subpart is covered), and reconstruction (nothing lost) of the whole data schema. During allocation it is decided where to fit the fragments, and also whether to support redundant or nonredundant allocation. In our case we start with nonredundant and continue with redundant allocation since replicating a fragment through several sites, if well handled, may increase the availability of the application.

The following method can be used to determine the nonredundant allocation of fragments [3]:

- A measure is associated with each possible allocation and the site with the best measure is selected ('best fit' approach).

Either of the following two methods can be used for the redundant allocation of fragments [3]:

- Determine the set of all sites where the benefit of allocating one copy of the fragment is higher than the cost, and allocate a copy of the fragment to each element of this set ('all beneficial sites').
- Determine first the solution of the nonreplicated problem, and then progressively introduce replicated copies starting from the most beneficial ('additional replication').

Since we support replication, a study has to be made in order to decide, for each site, whether to replicate the fragment under study or not. This can be done, for example, in the following two ways.

Let i be the fragment index, j is the site index, k is the application index, f_{kj} is the frequency of application k at site j, r_{ki} is the number of retrieval references of application k to fragment i, u_{ki} is the number of update references of application k to fragment i and $n_{ki} = r_{ki} + u_{ki}$. Then:

i. The 'best-fit' approach

For a nonreplicated allocation, we place fragment R_i at the site j where the number of references (Reads and Updates) to R_i is maximum. The number of local references of R_i at site j is:

$$B_{ij} = \sum_{k} f_{kj} n_{ki} \qquad \textbf{(1)}$$

R_i is allocated at site j^* such that B_{ij*} is maximum.

ii. The 'All beneficial sites' approach

Using the 'all beneficial sites' method for replicating allocation, we place fragment R_i at all sites j where the cost of retrieval references (Reads) of applications is larger than the cost of update references (Updates) to R_i from applications at any other site. The B_{ij} is evaluated as the difference:

$$B_{ij} = \sum_{k} f_{kj} r_{ki} \ - \ C \times \sum_{k} \sum_{j' \neq j} f_{kj'} u_{ki} \qquad \textbf{(2)}$$

C is a constant which measures the ratio between the cost of an update and a retrieval access. Typically, update accesses are more expensive, since they require a larger number of control messages and local operations (thus $C \geq 1$). R_i is allocated at all sites j^* such that B_{ij*} is positive; when all B_{ij} are negative, a single copy of R_i is placed at the site such that B_{ij*} is maximum.

3 A step-by-step secure conceptual design phase methodology

In the proposed methodology ([5,9]), that is based on an RBAC-oriented approach ([10]) for health care environments, the secure conceptual design phase is usually the most critical in the overall process of designing a distributed secure medical database system. The secure conceptual design phase includes the following steps:

- *Step 1*: Identification of sites, subjects, objects, and permitted actions (identification process).
- *Step 2*: Assignment of security labels (labeling process).
- *Step 3*: Fragmentation and allocation of data (distribution process).
- *Step 4*: Processing of security constraints.
- *Step 5*: Definition of permitted actions.

In the remaining of this section we outline a step-by-step design methodology for the implementation of the secure conceptual design phase of a distributed database.

Step 1: Identification of subjects, objects, sites and permitted actions (identification process).

Substep 1.1: Identification of the subjects.
As was previously mentioned, we have chosen to support the concept of user roles for the representation of the security subjects. So, characterization of the different user roles within the application must be performed in the first place. This is performed by studying the duties of the users within the application and the possible grouping of these duties into a user role, that can have different sensitivities that depend on the need-to-know of the persons playing this role.

The responsibilities of the individuals are characterized into two distinct levels of abstraction for the development of a User Role Hierarchy (URH): user roles and user categories. User roles allow the security designer to allow particular privileges to individual roles. To represent similarities that exist among user roles, a user category can be defined. Different categories can be grouped into categories of higher level. The highest category, represented as the root, contains all user roles. Privileges that are supplied to each role are passed on to its categories. The grouping of user roles into user categories is very application dependent. This step can be further divided into the following substeps:

1. Define all the user roles that exist within an application.
2. Group the existing user roles under the corresponding user category depending on the task of the user in the application.
3. Repeat step (2), grouping all user categories into higher category, until reaching the root category.

As a result of these actions, the security designer obtains an initial characterization of the URH.

Substep 1.2: Identification of the objects.

The security objects are the target of the security protection, and are in another sense the data contained in the application. Generally, data are characterized by high complexity and heterogeneity in both the nature and the sensitivity levels of the different data sets included in them (especially in environments where sensitivity is not easily defined as for example in healthcare ones). Organizing these data in a structured manner is necessary for the development of the appropriate user views, which in its turn is a required step in the design and implementation of the application.

Data is grouped into data sets, that will be target for the labeling process later. Data sets represent data with a common use. These data sets are grouped into a number of data categories. A data category characterizes common characteristics among related data sets. In turn, the different data categories can be grouped into one or more data categories of higher levels.

Substep 1.3: Identification of the sites.

This step consists on studying the sites where data will be distributed. This study should identify the following characteristics:

- the function of each site in the context of the whole application,
- the technical characteristics of the site (storage capacity, processing power, etc.), especially if they may constitute a constraint,
- the type of connection of the site to the network (LAN, Internet, dial-up, etc.),
- whether the site is mobile or not and the degree of mobility, since this may have a significant impact on the design,
- the security threats to each site,
- any special requirements and conditions of the site.

Substep 1.4: Identification of the permitted actions.

Since we have chosen to support both DAC and MAC security policies, there are different types of accesses that can be executed by subjects to objects. These operations are not defined only by the sensitivity levels of both (for example the *-property of Bell-Lapadula), but they also depend on the needs of the user roles, with respect to the security policy.

The basic well defined actions in most DBMSs are 'Select', 'Update' which includes 'Insert' and 'Delete'. In a medical database, deletion of sensitive data must be prohibited for follow up reasons.

Step 2: Assignment of security labels (labeling process).

Since our security policy supports MAC, the reading and writing of data by individuals is based on their authorized security clearance label. And since, the DAC security policy is supported too, a richer set of access modes that are specific to the particular type or category of information is granted to the individuals based on their need-to-know requirements. For this reason, we have first to define the security labels, assigned to users, data and sites, and then to define the type of permitted access.

The security label has two parts: The level and the category. The category of a user depends on his position in the User Role Hierarchy (URH). The highest category that corresponds to the root of the URH tree contains all sub categories and roles defined in the application. The category of data depends on its usage and corresponds to the needs of some categories of users. The category of a site depends on its use (what categories of users use it and what categories of data they need to use) and on its type (whether it is standalone, internet connected, mobile, stationary, etc).

The security levels assigned to each one of the users, data and sites are defined after an initial study that results in a rating of the clearance of users, the sensitivity of data and the trustworthiness of sites.

Substep 2.1: Assignment of security labels to the subjects

Having obtained an initial characterization of the user roles (URH), this substep allows us to assign a security label to each user role starting from the root and moving to the last level before the leaves (that form the user roles) of the tree. More specifically:

1. assign the category of the user role, as represented in the URH, then
2. assign a level to the user role that represents the level part of the label. This level depends on the trustworthiness of the user, and on his/her responsibility as defined in [5], [9].

By completing this substep, the security designer obtains an initial characterization (overall structure) of the User Role Hierarchy, and a primary assignment of labels to users.

Substep 2.2: Assignment of security labels to the objects

After the overall structure of the User Role Hierarchy (URH) has been established, the security designer can then proceed to assign data sets to the URH leaves which correspond to the data that needs to be accessed by the user roles. In this process, the designer uses the information accessing requirements as a guide for ensuring that the correct privileges are given. The procedure is repeated until reaching the root.

This step can be sub-divided into the following substeps:

1. Assign data sets starting with the user roles under a user category.
2. Move data sets shared by all user roles to their common user category.
3. Move data sets shared by all user categories to their common super categories. This procedure is repeated till reaching the root.
4. The label to be assigned to the data set then consists of :
   - a category that corresponds to the category of the node or the leaf already assigned in the URH.
   - a level that corresponds to the lowest clearance level of the user roles contained in the category.
5. Repeat this process for all the data sets within each user role, and then within their user categories.

As a security check for this substep, a stand-alone classification of the sensitivity of the data can be processed. This classification is dependent on both the content and the context of the data. In the case where the levels differ, the security designer must check and decide which level to assign. By completing this substep, the designer obtains an initial identification of privileges of each URH node and an initial label assigned to each data set.

Substep 2.3: Assignment of security labels to the sites

As in the case of users and data, each site should also be labeled with a level and a category. The security label of each site is derived from a function of physical and operational parameters of the site, as well as the type of connection and the grade of its mobility.

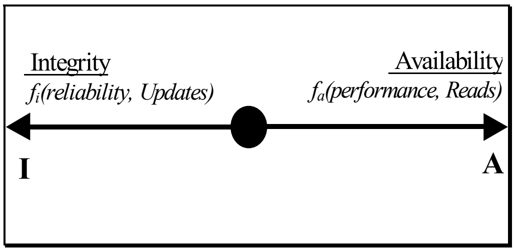

Fig. 1.

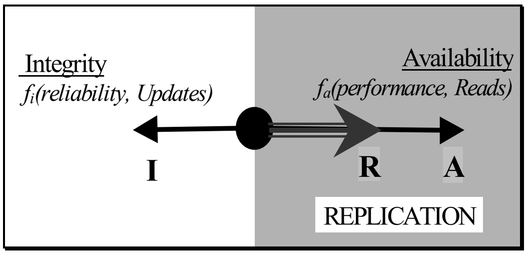

Fig. 2.

The category of the site depends on its use, i.e. the category of local users accessing the site (local users include all users having direct access to the site).

The security level of the site depends on several parameters related to the site, for example the vulnerability of the O.S. and the DBMS, the physical environment, and nature of the site, etc. The vulnerability of the O.S. and DBMS can be rated by using national or international security evaluation criteria and methods ([9]). A very significant characteristic of the site is also whether it is mobile or stationary since vulnerabilities in both categories differ strongly ([7]).

Step 3: Fragmentation and allocation of data (distribution process).

Having identified the subjects, the objects, the sites and the various types of access and assigned the security labels, this step focuses on the study of how data will be distributed on these sites. This distribution process can be divided into 2 substeps: fragmentation and allocation.

Substep 3.1: Fragmentation.
In our model, the fragmentation of the global schema is driven by two factors: the different users' needs (data should be distributed to several sites), and the different sensitivities of the data (data should be confined only in some sites).

First, depending on the users' needs the data sets are examined. Since we support the need-to-know principle, no user should have access to data that is not needed to perform his task. This results to an initial vertical and/or horizontal fragmentation of the initial data sets ([1]).

Then, we begin to look for all the security constraints that have been formally defined during the requirements analysis. As soon as the security designer begins to add the security constraints (generally prohibited tasks), the conflicts' identification analysis process begin ([9]), to notify the designer when assigned/prohibited actions contrast. Different types of conflicts can be found depending on the type of constraints to be inserted. In fact, in order to prohibit an unauthorized access from a user to a data set, an upgrade of the part of the data set involved must be performed ([9]). Depending on the type of the security constraint, and since we have chosen the tuple level granularity, this can lead to one of the following actions:

1. If the part to be upgraded is the whole table, then all the table and all its rows are upgraded.
2. If the part to be upgraded is just some rows of the table, then just these rows are assigned a higher security label.

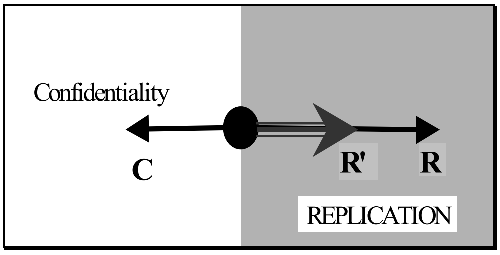

Fig. 3.

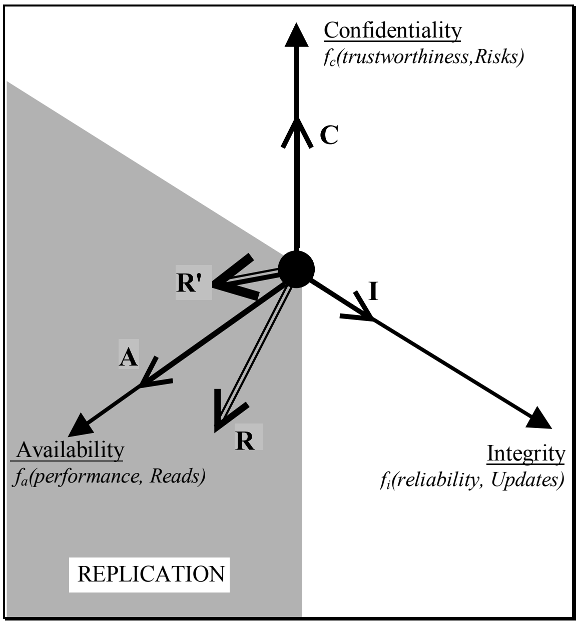
Confidentiality
f_c(trustworthiness,Risks)
C
R'
I
A
R
Availability
f_a(performance, Reads)
Integrity
f_i(reliability, Updates)
REPLICATION

Fig. 4.

3. In the case that the part to be upgraded is only some fields of a table, then this causes the initial table to be fragmented into two tables. One part is assigned a low and another a strictly higher level.

The final fragmentation and allocation scheme is obtained only after completion of step 4 below, when all the security constraints have been processed. As a result, each tuple of these fragments is assigned a security label and can be accessed by one or more user roles.

Substep 3.2: Allocation and replication

One of the important factors that may affect system performance is the replication of the fragments through the sites. The replication scheme determines how many replicas are created for each fragment, and to which sites these replicas are allocated. This affects for example the performance of the distributed system since reading locally is faster and less costly than reading from a remote site. The decision of whether to replicate or not is usually based on two factors: the number of reads versus the number of updates (or, as seen below, the availability versus the integrity (Fig. 1)).

More precisely, if a fragment is frequently accessed in a read mode, a widely distribution is mandated in order to increase the number of local reads and to decrease the load on the network, which can be understood as increasing the availability of the whole system. On the other hand, an update of a fragment can be written to all or to a majority of the replicas (depending on the algorithm supported by the DDBMS). In this case, a wide distribution slows down each write and increases its communication cost. Therefore, if a fragment is sufficiently accessed in update mode, a narrowly distribution is mandated. This can be understood as balancing the cost of integrity that is increased by the replication with the cost of availability that is increased now by the non replication. So the decision whether to replicate or not should be driven by this two contrary forces (*A* for availability and *I* for integrity) that can be seen as vectors. The result of replicating or not is then depending on the area where the resultant *R* of the two vectors resides (Fig. 2).

There is however also another important factor to be examined. This is the confidentiality issue which is particularly important in environments containing sensitive data, as for example the health care ones. Confidentiality acts as a contrary force *C* against the resultant *R* of the figure 2. The final decision whether to replicate or not, taking into consideration the confidentiality requirements of the specific site, should be based on the area where the resultant *R'* of the two vectors resides (Fig. 3).

Concluding the above observations and placing all the facts together, we can say that the final decision whether to replicate or not should be driven by the combination of three forces (*A* for availability, *I* for integrity and *C* confidentiality) that can be seen as vectors.

The result of replicating or not is then depending on the area where the common resultant *R'* of the three vectors resides, as for example can be seen in figure 4 where the overall scheme of the decision making procedure has been represented in a two-dimension perspective and an initial estimation has been made of the cases where replication should be preferred (gray area).

When speaking about confidentiality in the above figures, we mean the need for confidentiality to be provided by the examined site in the case of allocating a specific fragment of data. In fact, the need for confidentiality depends mainly on both the estimated risks for exposure of sensitive data and the security level of the site. For example, if the risk for exposure of sensitive data is high and the site is not secure, the decision tends towards non replication even if it would have been beneficial in the reading procedure.

A summary of the objectives and depending factors for evaluating each of the three vectors presented in figure 4, is proposed in the following table 1.

The exact way in which the above objectives and factors participate in the final decision depends on the particular characteristics of the site. These characteristics contribute as a weight in the measurement of the three security components. For example, the weight of availability in a site dedicated for emergency cases should be higher than that of the confidentiality. A detailed study of how these weights should be assigned, as well as how confidentiality may be measured, is currently under further study.

Table 1. Summary of objectives and factors for replication decision.

	OBJECTIVE	FACTORS
CONFIDENTIALITY	Minimum Exposure Risks (maximum secrecy)	– Estimation of risks for exposure of sensitive data – Trustworthiness rate of the site
AVAILABILITY	Maximum Locality of READS (minimum remote access)	– Frequency of READ-accesses – Performance rate of the site
INTEGRITY	Minimum Remote UPDATES (maximum data consistency)	– Frequency of UPDATE-accesses – Reliability rate of the site

In our methodology we use an extension of the combination of the ‘best-fit’ and the ‘all beneficial sites’ methods for the allocation and replication of data, respectively. Following is a step by step explanation of how we use it, taking as objective the optimization of the three security components.

Substep 3.2.1: Allocation

We support the 'best-fit' approach for a nonreplicated allocation. This step can be seen as studying how to obtain the minimum remote access of the data in case of non replication. The value of B_{ij} in our methodology depends on the total number of reads and updates and on the trustworthiness of the site.

$$B_{ij} = \sum_k f_{kj} n_{ki} + f(\textit{trustworthiness of the site}) \qquad \textbf{(3)}$$

Substep 3.2.2: Replication

To apply replication, we must decide, for each site, whether to replicate the fragment under study or not. Using the 'all beneficial sites' method for replicating allocation, we place fragment R_i at all sites j where the resultant of the three vectors falls in the shaded area of the figure 4.

$$\begin{aligned} B_{ij} &= f_{confidentiality}(\textit{trustworthiness, Risks}) \\ &\quad + f_{availability}(\textit{performance, Reads}) \\ &\quad\quad + f_{integrity}(\textit{reliability, Updates}) \end{aligned} \qquad \textbf{(4)}$$

Steps 4 - 5: Processing of security constraints - Definition of permitted access types.

After the identification, labeling and distribution processes, the different security constraints should be examined. For each constraint, checking must be performed to ensure its obedience. In case of conflict, the security designer should be notified to decide about the implementation procedure ([9]). Since we have chosen also to support DAC security policy, the permitted access types must be defined for all subjects to the data sets under their clearance. This definition depends on the need-to-know requirements of the subjects dominated by their clearance and their responsibilities in the application.

4 The experimental implementation

The AHEPA University Hospital, which has been used as our test-bed for designing secure database systems, is a general hospital which is part of the Aristotelian University of Thessaloniki. The following figures describe briefly the hospital: 16 clinics including the reference and hospitalization center for AIDS patients from all Northern Greece; 40 laboratories; a radiological department including M.R.I., C.T., U.S., D.S.A, X-rays, etc.; a nuclear medicine department (SPECT, Gamma-Camera); 705 beds; 520 medical doctors including consultants; 762 nursing personnel; 466 personnel for financial and general support; 28,000 inpatients per year; 2,500 surgical procedures per year; 107,000 outpatients per year; 2,345,000 laboratory tests per year (1993).

For the purposes of the experimental implementation a set of appropriate data flow diagrams (DFDs) and entity relationship models (ERMs) for the representation of data and functions, concerning the drugs delivery and the examinations orders and results in AHEPA, have been developed. For the identification of the sites, a general model of the Health Information System architecture has been utilized. Eight different user roles have been identified and grouped in a five-level-deep URH. Nineteen data sets have also been identified and have been grouped in a six-level-deep tree. An initial fragmentation of the data sets has been obtained, based on the different user needs and sensitivities of the data. Further fragmentation has been provided by the examination of the security constraints. The allocation and replication of the fragments has been based on the 'best fit' and 'all beneficial sites' approaches, respectively. Specific work is currently taking place on the area of measuring the trustworthiness, performance and reliability of the sites. An additional study has also started on the implications of the location and specific situation parameters on the design.

5 Conclusion

During the design of distributed database systems, parameters related to performance and reliability are usually taken into account in order to decide the allocation and replication scheme. Security is however another very important issue that should also be taken into account when designing a distributed medical database system.

This paper describes a step-by-step methodology for the design of a secure distributed medical database system. The methodology is based on the combination of mandatory and discretionary security models and uses hierarchies of user roles (RBAC based approach), data sets and sites in order to decide the secure distribution of the application.

An experimental implementation of the proposed methodology in a major Greek hospital has shown the usefulness of the proposals as well as their effectiveness in limiting the unauthorized access to the medical database, without severely restricting the capabilities of the system.

6 References

1. Ozsu, T., Valduriez, P.: Principles of distributed database systems. Prentice Hall (1991)
2. Bell, D.: Distributed database systems. Addison Wesley (1993)
3. Ceri, S., Pelagatti, G.: Distributed Databases: Principles and Systems. NY, McGraw-Hill (1985)
4. Castano, S., Fugini, M., Martella, G., Samarati, P.: Database security. Addison Wesley (1994)
5. Pangalos, G., Khair, M.: Design of a secure medical database systems. IFIP/SEC'96, 12th International Information Security Conference (1996)
6. Wolfson, O., Jajodia, S. and Huang Y.: An Adaptive Data Replication Algorithm. ACM Transactions on Database Systems, Vol. 22, No. 2 (June 1997) 255-314
7. Mavridis, I., Pangalos, G.: Security Issues in a Mobile Computing Paradigm. Communications and Multimedia Security (CMS'97). Vol.3 (1997) 60-76
8. Fugini, M.: Secure database development methodologies, in Database security. Landwehr (ed.) (1988)
9. Pangalos, G., Khair, M., Bozios, L.: An integrated secure design of a medical database system. MEDINFO'95, The 8th world congress on medical informatics, Canada (1995)
10. Ferraiolo, D. and Kuhn R.: Role-based access controls. 15th NIST-NCSC National Computer Security Conference. Baltimore, MD, October 13-16 (1992) 554-563