SUBJECT SWITCHING ALGORITHMS FOR ACCESS CONTROL IN FEDERATED DATABASES

Jacqueline Yang, Duminda Wijesekera, Sushil Jajodia
Center for Secure Information Systems, George Mason University, Fairfax, VA 22033-4444

Abstract: Authentication in federated database systems present difficulties because the autonomously operated components may not know the identity of federation users. One proposed solution is subject switching, where the federation translates the federated users identity to that of an agreed upon component subject. This translation may be problematic, due to not having component subjects with the same accesses requested by federation users. Therefore, we propose using proximity measures between requested and provided accesses and present two policy neutral algorithms to find proximity minimizing matches between a federation subject and a collection of component subjects.

Key words: federated database, access control, database security

1. INTRODUCTION

Demand for seamless information sharing has inspired many computing resources to be interconnected in numerous ways into large-scale information systems. Federated databases can provide integrated yet autonomously controlled services for permanent logical data storage. A federated database is a set of autonomous and possibly heterogeneous databases that participate in a collective enterprise without centralized control. Consequently, any component database of a federated system could participate in more than one federation and could be a federated database system itself. Implied autonomy of component databases in federated databases facilitates heterogeneity in data models, query languages, concurrency control, transaction management and access control.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35587-0_24
M. S. Olivier et al. (eds.), *Database and Application Security XV*

Authentication of federation users presents problems in federated database systems because a user's identity may not be known to all component databases. One possible approach to deal with this problem is to permit federation users to assume the identity of the federation. This approach has the drawback that every federation user obtains all the access rights of the federation, unless the user accesses are restricted using some mechanism by the federation.

Another possible approach is to change the identity of the user to other federation subjects based on prior agreement between the federation and the components. This method, commonly refereed to as subject switching [JONS94], is the topic of this paper. In order to map a federation-wide subject to a component database, it is necessary to find a component subject with the same privileges, which may not be always possible. In this case, we propose two alternatives, called least under permissive and least over permissive. The former gives approximately the same privileges without giving more permissions than requested, while the latter gives the least amount over the requested permissions.

The organization of the remainder of this paper is as follows. Section 2 introduces the basic concepts, overall architecture and autonomy in access control of federated database systems, with a description of degrees of component autonomy. Section 3 presents the concept of and algorithms for subject switching. It further shows why subject switching is not always possible. As a remedy, Section 4 presents some metrics that can be used to approximately match access requirements of a federated subject to a collection of component subjects and provides some algorithms. Section 5 contains our concluding comments.

2. FEDERATED DATABASE SYSTEMS

A federated database system [HEIM85] consists of number of autonomous component database systems that control their interactions with other members in the federation in order to provide a substantial degree of information sharing. The DBMS of a component database system could be a centralized, distributed or another federated DBMS. Also any component database system can participate in more than one federation. In a federated database system, the components have control over the data they manage and their operations and determine the data that it wishes to share with other components. Federated database systems are categorized as loosely coupled or tightly coupled. Loosely coupled systems gives users freedom to join or leave a federation, and doe not provide federated schemas to be used at the federated level. Tightly coupled systems provide one interface at the

federation level for federated services and provide federated schemas for users. This paper considers tightly coupled systems.

Autonomy is important for components in a federated database system. Veijalainen and Popescu-Zeletin [VEIJ88] identifies three dimensions of autonomies: *design*, *communication* and *execution*, and Sheth and Larson [SHET90] proposed an additional dimension, *association* autonomy, which is the freedom for a component database to join or leave the federation at will. Jonscher and Dittrich [JONS94] proposed another dimension of autonomy, *authorization autonomy*, which is the ability for component databases to autonomously control their access control decisions.

One major issue related to access control in tightly coupled systems is that if access to federated schemas are to be protected and the components are to retain their own access control decisions, then the identity and authentication of federation users have to be shared between components and the federation management [DeCa97]. Depending on the level of component participation in the identity and authorization of federated users, authorization autonomy can be further classified as (1) full authorization autonomy, (2) medium authorization autonomy and (3) low authorization autonomy [JONS94]. In full authorization autonomy, every federation user has to be known to each component, and has to be authenticated in order to access data. In medium authorization autonomy, the federation authenticates itself to components, and submits the federation user's identity and the request to the component. In low authorization autonomy, the federation authenticates itself and uses its own identity to access data. One of the major disadvantages of this arrangement is that unless controlled by other means, components databases now allow such a user to obtain all access rights of the federation, which is not an ideal situation. Even in the case of medium authorization autonomy, because component databases do not authenticate federated users, as suggested by Jonscher and Dittrich [JONS94], their identity can be changed in agreement between the federation and the components. This method is commonly referred to as subject switching.

In federations with full authorization autonomy, component databases share the identifiers and, consequently, there is no issue in mapping their identity to subjects known to components. If either medium or low authorization autonomy is preserved, subject identities need to be verified by the federation management only, as component databases trust the federation to validate the identity of its subjects. We assume that the federation management maintains mapping each federated (subject, component) pair to components pairs of the same type. This way, the federation need not apply for local access permissions for each federated user.

3. SUBJECT MAPPING ALGORITHMS

In order to map a federation wide subject to a component database, it is necessary to find a component subject with the same set of privileges, which may not be always possible. In that case, we propose an approximate matching algorithm where the local subject may have approximately the same privileges as the federated subject. Our algorithm is based on the federation management maintaining an access control list for each federated subject and federated and component subjects have a set of permissions and prohibitions (i.e., positive and negative permissions) per object. We also assume that the federation management maintains a list of subject IDs in every component database with their corresponding capabilities (i.e., combinations of access rights and prohibitions). In order to discuss our algorithms we need some definitions and measures of permission disparities between (subject, object) pairs.

3.1 Motivating Example

We motivate our work with a running example of a clinical trial, (such as testing a potential treatment regiment) which is carried out in a federation of two hospitals, and coordinated by a research center that must maintain an extensive database of outcomes of the trial. Independent of this study, all hospitals are required to maintain their autonomous databases, but will cooperate in carrying out the study. Thus, individual hospitals will be the components, and the research center will be the federated database. As there are different types of hospitals (such as teaching hospitals, regional trauma centers and rural hospitals), they may have different subject hierarchies. For example, teaching hospitals have medical residents of different seniorities, professors, and non-clinical researchers. We assume that all individuals who have a primary affiliation with a medical facility have local accesses assigned to them by their home institution, but in order to participate in the study they need to become a federation user. In addition, there are other federation users (such as members of regulatory bodies and state medical-ethics boards) that may need access to the study, but have no affiliations with participating hospitals. The researchers will be given different privileges to access the local data, depending on the requirements of the research. We assume there are two hospitals A and B having the same export schemas, *Patient, Treatment, Lab_Result* and *Complaint*. Hospital A in our example is a research hospital, and consequently, has a deeply nested hierarchy of database users for fine grain control of information. Hospital B is an affiliated rural hospital. Each of these hospitals created their local users with a set of assigned privileges as given below.

Hospital A: Large Teaching Hospital
{Staff_Physician: Patient:r, Treatment:rw, Lab_Result:r, Complaint:r}
{Medical_Resident: Patient:r, Treatment:rw, Lab_Result:r}
{Nurse: Patient:r, Treatment:r}
{Case_Worker: Patient:rw, Complaint:rw}
{Lab_Technician: Lab_Result:rw}
{Non_Clinical_Researcher: Patient:r, Treatment:r, Lab_Result:r}

Hospital B: Affiliated Rural Hospital
{Physician: Patient:r, Treatment:rw, Lab_Result:r, Complaint:r}
{Nurse: Patient:rw, Lab_Result:rw}
{Case_Worker: Patient:r, Complaint:rw}

Federation: Experimental Treatment Management
{Physician: Patient:r, Treatment:rw, Complaint:r}
{Researcher: Patient:r, Treatment:r, Lab_Result:r}
{Nurse: Patient:rw, Complaint:r}
{Regulatory_Supervisor: Patient:r, Treatment:r, Lab_Result:r, Complaint:r}
{Medical_Ethics_Supervisor: Patient:r, Complaint:r}

3.2 Access Compatibility Measures and Algorithms

Definition 1(Compatibility of Permissions and their Disparity Measures) Let L be a collection of access requests, i.e. $L = \{(o_j, -a_j) : j < J\}$ where o_j, a_j are objects and actions.

1. We say subject S_1 is a *negative cover* of subject S_2 with respect to L if or every object, prohibition pair (o, -a) of L, if S_2 has prohibition (o, -a) then S_1 also has (o, -a).
2. If S_1 is negative cover of S_2, then the cardinality of {(o, -a) of S_1} – {(o, -a) of S_2} is the *negative disparity* measure between S_1 and S_2.
3. We say subject S_1 is a *positive cover* of subject S_2 if for every object, permission pair (o, +a) of L if S_2 has permission (o, +a) then S_1 also has (o, +a).
4. We say that S_1 is a cover of S_2 with respect to L if S_1 a positive cover of S_2 and S_1 is a negative cover of S_2 with respect to L.
5. If S_1 is positive cover of S_2, then the cardinality of {(o, +a) of S_1} – {(o, +a) of S_2} is the *positive disparity* measure between S_1 and S_2.
6. We say that the pair *(positive disparity, negative disparity)* is the disparity measure between S_1 and S_2.

[]

Intuitively, S_1 is a positive cover of S_2 if (positive) permissions on an access request list L allowed for S_2 are also allowed for S_1. The number of permissions that are additionally allowed for S_1 is the positive disparity

measure of S_1 over S_2. Hence a positive cover of a subject dominates the subject with its permissions, while a negative cover of a subject dominates the subject with prohibitions. The pair (positive disparity, negative disparity) measures the collective permissions and prohibitions that a covering subject has over a covered object. The objective of our mapping algorithms is to minimize the disparity measure between requested permission for federated subjects and component subjects that they are mapped to.

If a perfect match (i.e., collection of component subjects with exactly the kind of prohibitions and permissions that are required by a federation subject) exits, then it can be found by searching. Failing a perfect match, we look for an approximate match. Appropriateness of the approximate match depends upon the security policy enforced. There are two alternatives: the first is to give the best possible permissions, and nothing more than requested and the other is to give the least amount over the requested permissions, named *least under permissive* and *least over permissive* algorithms respectively. In order to consider these notions, we first clarify our notion of a perfect match in the following definition.

Definition 2 (Perfect Match) Let S be a federated subject with an access request list L, and each component subject S_i in component database C_i has the access list $L_i = \{(o_j, a_j) : j < M_i\}$ for some integer. Let $L = U \{ L_i : 1 =< i =< n\}$. We say that component subject list $(S_1,\ldots,S_n)$ provides a perfect match for federation subject S if S_i is a cover with respect to L_i for every i.[]

Definition 2 exemplifies one of the basic simplifying assumptions of our work. That is, any federated subject S provides its request list L in terms of component requests, and any collection of component subjects $(S_1,\ldots,S_n)$ are considered to be providing all of them if subject S_i can provide its share L_i of L. Next two algorithms compute perfect matches for a given federated subject S with an access request list L. Section 4 proposes possible remedies in terms of best possible matches, when a perfect match fails to be found.

Algorithm 1 (Least Under Permitting Algorithm) Let S be a federated subject to be mapped to a collection of component databases C_i for $i < N$.

For each $i < N$:

Let $N[i,j]$ for $j < N_i$ for some integer N_i be the sequence of all negative covers of S in C_i, arranged in increasing order of their negative disparity.

1. If the sequence $N[i,j]$ is empty for some i, then subject S cannot be matched to the component database C_i.
2. Else (i.e., $N[i,j]$ is nonempty), choose the least integer $j < N_i$ such that $N[i,j]$ has the least positive disparity with respect to S_i. Then map S to subject $N[i,j]$ of the component database C_i. If none of the $N[i,j']$ for $j' < N_i$ is a positive cover, then there is no subject that can provide accesses that are requested by the federation subject. []

Example 1 Consider the federation users and the local users for Hospital A and Hospital B. By applying Algorithm 1, we have the following mappings for Hospital A, but no match exists for Hospital B.
(Federation.Physician, Hospital_A.Nurse)
(Federation.Researcher, Hospital_A.Non_Clinical_Researcher)
(Federation.Regulatory_Supervisor, Hospital_A.Non_Clinical_Researcher)

Algorithm 2 (Least Over Permitting Algorithm) Let S be a federated subject to be mapped to a collection of component database C_i for $i < N$.

For each $i < N$:
Let $N[i,j]$ for $j < N_i$ for some integer N_i be the sequence of all positive covers of S in C_i, arranged in increasing order of positive disparity.

1. If the sequence $N[i,j]$ is empty for some i, then subject S cannot be matched to the component database C_i.
2. Else (i.e., $N[i,j]$ is nonempty), choose the least integer $j < N_i$ such that $N[i,j]$ has the least negative disparity with respect to S_i. Then map S to subject $N[i,j]$ of the component database C_i. If none of the $N[i,j']$ for $j' < N_i$ is a negative cover, then there is no subject that can provide accesses that are requested by the federation subject.[]

Example 2 By applying Algorithm 2, we have the following mappings for both Hospital A and Hospital B.
(Federation.Physician, Hospital_A.Staff_Physician)
(Federation.Researcher, Hospital_A.Non_Clinical_Researcher)
(Federation.Nurse, Hospital_A.Case_Worker)
(Federation.Regulatory_Supervisor, Hospital_A.Staff_Physician)
(Federation.Medical_Ethics_Supervisor, Hospital_A.Case_Worker)
(Federation.Physician, Hospital_B.Physician)
(Federation.Researcher, Hospital_B.Physician)
(Federation.Regulatory_Supervisor, Hospital_B.Physician)
(Federation.Medical_Ethics_Supervisor, Hospital_B.Case_Worker)

As stated, Algorithms 1 and 2 may not find a perfect match for the given federated subject and access request list. Theorem 1 shows that these algorithms find them if they exist.

Theorem 1 Suppose S is a federated subject with an access request list $L = \cup L_i$, where $L_i = \{(o_j, a_j) : j < M_i\}$ be the collection of objects in component database C_i for some integer M_i. Algorithms 1 and 2 will find a perfect match for S with respect to L if and only if it exists. Furthermore:
1. The least under permitting algorithms will find a perfect match that minimizes the negative disparity measure.
2. The least over permitting algorithms will find a perfect match that minimizes the positive disparity measure.

Proof: Suppose there is a perfect match $(S_1,\ldots,S_n)$. Then, each S_i is a cover for S with respect to L_i. Therefore, N[i,j] found in step 1 of Algorithms 1 and 2, are non-empty for each i. Furthermore, in step 2:
1. Algorithm 1 finds the component subject with the least negative disparity because N[i,j] is ordered in the increasing order of negative disparity.
2. Algorithm 2 finds the component subject with the least positive disparity because N[i,j] is ordered in the increasing order of positive disparity.
Conversely, if either Algorithm 1 or 2 finds a set $(S_1,\ldots,S_n)$, then each element S_i is the least j that is a positive/negative cover of S in N[i,j] with respect to L_i. Accordingly, $(S_1,\ldots,S_n)$ is a cover for S with respect to $(L_1,\ldots,L_n)$.[]

4. APPROXIMATIONS: METRICS AND MAPPINGS

The previous section showed that finding a component subject with a given list of authorization requirements is not always possible. Consequently, we propose to find the *best possible approximations* that can be provided by respective component databases. Approximate matching requires a metric to measure the proximity of the provided accesses to those requested, and algorithms to find the optimal among the class of possibilities. We propose these in the current section.

4.1 Metrics

In the absence of a perfect match, in mapping a set of access requirements, an attempt must be made to make the *best possible alternative* under the given circumstances. In doing so, we consider four differences. They are the permissions and prohibitions required by the federation subject, but cannot be matched, and the permissions and prohibitions provided by the component, but were not required by the federation subject. We propose the *access disparity metric* given in Definition 3 to quantify these differences.

Definition 3 (Access Disparity Measures for Approximate Mappings)
1. Suppose $<(o_i, a_i): i < k>$ is a vector of possible signed accesses in the given security domain. We assume that a_i has positive (+) sign for permissions, negative (-) sign for prohibitions and a 0 when neither is specified. We call such an access list a *complete access list* for a subject.
2. Let S, S' be two subjects with access lists $L = \{(o_i, a_i) : i < k\}$, and $L' = \{(o_i, a'_i) : i < k\}$. Then, let the vector difference between $<(o_i, a_i) : i < k>$, $<(o_i, a'_i) : i < k>$ be constructed as follows:

Under Prohibition = |{ i : a_i is a prohibition, L has (o_i, a_i) but not L' }|
Over Prohibition = |{ i : a_i is a prohibition, L' has (o_i, a_i) but not L}|
Under Permission = |{ i : a_i is a permission, L has (o_i, a_i) but not L' }|
Over Permission = |{ i : a_i is a permission, L' has (o_i, a_i) but not L }|
Disparity Measure = (under prohibition, over prohibition, under permission, over permission)[]

As will be shown shortly, Definition 3 is a generalization of Definition 1, and applies to any pair of subjects and not only to pairs where the second subject is a positive or negative cover of the first. One of the advantages of this metric is that it accurately reflects the nature of approximation in subject mapping. Details captured in this metric also make it more difficult to use.

Thus, we present a simpler, yet intuitive metric that can numerically quantify access differences between two subjects in Definition 4. The main differences between Definition 3 and Definition 4 are that the latter provides an aggregate over the former, and consequently makes no difference between permissions and prohibitions. This relationship between matrices of Definitions 3 and 4 are stated and proved in Theorem 2.

Definition 4 (Numerical Disparity Measure)

1. Suppose S is a subject, and $\{(o_i, a_i) : i < k\}$ is its complete access list. Let $\{(o_i, n_i) : i < k\}$ be constructed from $\{(o_i, a_i) : i < k\}$ by replacing:
   - $+a_i$ with +1.
   - $-a_i$ with –1.
   - Neither permission with 0.

   We call such a list the *numerical analog* of $\{(o_i, a_i) : i < k\}$.
2. Let S, S' be two subjects with complete access lists $\{(o_i, a_i) : i < k\}$, $\{(o_i, a'_i) : i < k\}$ and their numerical analogs $\{(o_i, n_i) : i < k\}$, $\{(o_i, n'_i) : i < k\}$. Then we say $\Sigma \{ | n_i - n'_i | : i =< k\}$the *numerical access disparity* between S and S'.[]

Numerical access disparity measure does not take any policy-based information nor assign any weights to the four types of differences existing between desired and provided permissions. For the relationship between metrics, Theorems 2 and 3 prove that Definition 3 is a generalization of Definition 1.

Theorem 2 (Comparing Disparity Measures to Approximate Measures)
Suppose S and S' are subjects with complete access lists $\{(o_i, a_i) : i < k\}$ and $\{(o_i, a'_i) : i < k\}$. Then the following hold:
1. If S' is a positive cover of S, then under permissions of (S, S') is zero, and over permissions of (S, S') is the positive disparity measure defined in Definition 1.

2. If S' is a negative cover of S, then under prohibitions of (S, S') is zero, and over prohibitions of (S, S') is the negative disparity measure given in Definition 1.
3. If S' is a cover of S, then under permissions and under prohibitions of (S, S') are zero and over permissions and over prohibitions are positive disparity measures and negative disparity measures given in Definition 1.

Proof:
1. Suppose S' is a positive cover of S. Then according to Definition 1, for every object o, permission +a, if (o, +a) is on the access list of S, then it is on the access list of S'. Consequently, according to Definition 3 under permissions for (S, S') is 0. Furthermore, over permission for (S, S') is $|\{ i : +a'_i$ is a permission and $+a_i$ is not a permission for $S\}|$, which is the positive disparity according to Definition 1.
2. Similar argument proves this fact.
3. Follows from combining cases 1 and 2 above.[]

Theorem 3 (Comparing Approximate Disparity Measures)
Numerical disparity measures defined in Definition 4 relate to disparity measures defined in Definition 3 as given by the following inequalities:
1. *Under Permission + Over Permission =< Permission Difference.*
2. *Under Prohibition + Over Prohibition =< Prohibition Difference.*

Proof:
Suppose the disparity measures for under permission, over permission, under prohibitions, over prohibition are respectively *Uper*, *Oper*, *Upro* and *Opro* for the subject pair (S, S') as defined in Definition 4. Suppose that these values are computed from complete access lists $L=\{(o_i, a_i) : i < k\}$ and $L'=\{(o_i, a'_i) : i < k\}$ for S and S' respectively.
Suppose that (o_i, a_i) is a permission that is not in the complete access list L' of S'. Then, in the computation of under permission *Uper* a_i contributes 1 to the count in Definition 3, but because a'_i could be either prohibition or 0, this a_i contributes either a 1 or a zero to the permission difference of Definition 4. Since the indices i for objects contributing to under permissions and over permissions are disjoint, we get the relationship *Under Permission + Over Permission =< Permission difference.* A similar argument applies for prohibitions.[]

4.2 Approximate Mapping Algorithms

In the absence of a perfect match for a given access request set and the policy that all prohibitions have to be enforced, Algorithm 1 can be changed to grant a subset of permissions that have been requested.

Algorithm 3 (Approximate Under Permitting Algorithm)
Let S be a federated subject to be mapped to a set of component databases $\{C_i : i < N\}$. Let $L_i = L^+_i \cup L^-_i$ be the prohibitions and the permissions required by S from the component database C_i, where $L^-_i = \{(o_j, -a_j): j < N^-_i\}$ and $L^+_i = \{(o'_k, +b_k) : k < N^+_i\}$.
Stage 1: *(Search for the component subject with minimal prohibitions)*
For each $j'=N^-_i$ down to 0:
For every subset M[i,j'] of size j' from objects $\{(o_j, -a_j) : j < N^-_i\}$:
Let N[i,j'] be the collection of negative covers of S with respect to objects in M[i,j'] arranged in increasing order of negative disparity of Definition 1.
Case 1: If N[i,j'] is empty then j'=j'-1.
Case 2: If N[i,j'] is non-empty, then choose S' from N[i,j'] such that it has the least positive disparity with respect to S and subjects in N[i,j'] according to Definition 1.
Stage 2: *(Search for component subjects with maximal permissions)*
For each $k'=N^+_i$ down to 0:
Let M[i,k'] be a subset of size k' from objects $\{(o_k, +a_k) : k < N^+_i\}$ and N[i,k'] be the collection of positive covers of S with respect to objects in M[i,k'] arranged in increasing order of positive disparity of Definition 1.
Case 1: If N[i,k'] is empty then j'=j'-1.
Case 2: If N[i,k'] is non-empty. Then, choose S' from N[i,k'] such that it has the least number of prohibitions for objects in $\{(o_k, -a_k) : k < N^-_i\}$.
Stage 3: *(Search for a subject with minimum numerical deviation)*
Let $T(i,1), \ldots, T(i, n_i)$ be a list of all subjects in component C_i, arranged in an increasing order of numerical disparity with S. Then choose T(i,1) as the subject in component C_i.[]

Example 3 By applying Algorithm 3, we have the following mappings for both Hospital A and Hospital B. Comparing with Example 1, we have found the best approximate matches for each federation user.
(Federation.Physician, Hospital_A.Nurse)
(Federation.Researcher, Hospital_A.Non_Clinical_Researcher)
(Federation.Nurse, Hospital_A.Case_Worker)
(Federation.Regulatory_Supervisor, Hospital_A.Non_Clinical_Researcher)
(Federation.Medical_Ethics_Supervisor, Hospital_A.Nurse)
(Federation.Physician, Hospital_B.Physician)
(Federation.Researcher, Hospital_B.Physician)
(Federation.Nurse, Hospital_B.Case_Worker)
(Federation.Regulatory_Supervisor, Hospital_B.Physician)
(Federation.Medical_Ethics_Supervisor, Hospital_B.Case_Worker)

The difference between Algorithms 1 and 3 is that the former may not halt in case a perfect match cannot be found, whereas the latter always halts, doing the best approximate matching under the given circumstances. In

Algorithm 3, we first search for a match with as many number of prohibitions as requested by S, and if such subjects exist, we search for the one with the largest number of permissions requested. These two searches constitute steps 1 and 2 of Algorithm 3. If our search does not succeed, that implies that there is no subject in the corresponding component that does have either the required prohibitions or the required permissions. Then, we look for a component subject that has the least numerical disparity. This search constitutes Step 3 of our algorithm. The choice of the last alternative is somewhat arbitrary, and we are exploring better options for this case.

We can reverse steps 1 and 2 of Algorithm 3 and thereby obtaining the best approximate over permitting algorithm, as given below. Consequently, above explanation of Algorithm 3 applies to Algorithm 4 with the order of searching for permissions and prohibitions reversed.

Algorithm 4 (Approximate Over Permitting Algorithm) Let S be a federated subject to be mapped to a collection of component database C_i for i < N. Let $L_i = L^+_i \cup L^-_i$ be the prohibitions and the permissions required by S from the component database C_i, where $L^+_i = \{(o_j, -a_j) : j < N^-_i\}$ and $L^-_i = \{(o'_k, +b_k) : j < N^+_i\}$.

Stage 1: *(Search for component subjects with minimal permissions)*
For each $j'=N^+_i$ down to 0:
For each subset M[i,j'] of size j' from objects $\{(o_j, +a_j) : j < N^+_i\}$:
Let N[i,j'] be the collection of positive covers of S with respect to objects in M[i,j'] arranged in increasing order of positive disparity of Definition 1.
Case 1: If N[i,j'] is empty then j'=j'-1.
Case 2: If N[i,j'] is non-empty, then choose S' from N[i,j'] such that it has the least number of prohibitions for objects in $\{(o_j, -a_j) : j < N^-_i\}$.

Stage 2: *(Search for component subjects with maximal prohibitions)*
For each $k'=N^+_i$ down to 0:
Let M[i,k'] be a subset of size k' from objects $\{(o_k, -a_k) : k < N^-_i\}$ and N[i,k'] be the collection of negative covers of S with respect to objects in M[i,k'] arranged in increasing order of negative disparity of Definition 1.
Case 1: If N[i,k'] is empty then k'=k'-1.
Case 2: If N[i,k'] is non-empty. Then, choose S' from N[i,k'] such that it has the least positive disparity with respect to S and subjects in N[i,k'] according to Definition 1.

Stage 3: *(Search for a subject with minimal numerical deviations)*
Let $T(i,1), \ldots, T(i, n_i)$ be a list of all subjects in component C_i, arranged in an increasing order of numerical disparity with S. Then choose T(i,1) as the subject in component C_i.[]

Example 4 By applying Algorithm 4, we have the following mappings for both Hospital A and Hospital B. Comparing with Example 2, we have found the best approximate matches for each federation user.
(Federation.Physician, Hospital_A.Staff_Physician)
(Federation.Researcher, Hospital_A.Non_Clinical_Researcher)
(Federation.Nurse, Hospital_A.Case_Worker)
(Federation.Regulatory_Supervisor, Hospital_A.Staff_Physician)
(Federation.Medical_Ethics_Supervisor, Hospital_A.Case_Worker)
(Federation.Physician, Hospital_B.Physician)
(Federation.Researcher, Hospital_B.Physician)
(Federation.Nurse, Hospital_B.Case_Worker)
(Federation.Regulatory_Supervisor, Hospital_B.Physician)
(Federation.Medical_Ethics_Supervisor, Hospital_B.Case_Worker)

Similarly to Algorithm 3, Algorithm 4 may not halt in case a perfect match cannot be found. Instead, it will continue to search for a match with a smaller set of permissions requested. If it still fails, it will search for the one with minimal numerical disparity. Therefore, it will always find an approximate match for a federation user.

Theorem 4 (Properties of Algorithms 3 and Algorithm 4)
1. For any given federated subject S, Algorithms 3 and 4 terminate and provide a candidate for each component C_i.
2. If there are any candidates in some component C_i that are covers for S with respect to its access requests within C_i, then Algorithm 3 finds the one with the least minimum negative disparity, and Algorithm 4 finds the candidate with the least positive disparity.
3. If any component database C_i does not have a cover for S with respect to its access requirements, then both Algorithms 3 and 4 finds the candidate that has the least numerical disparity.

Proof:
1. Termination: In Algorithm 3, at the first stage, the component C_i is searched for a candidate subject S_i that has the smallest negative disparity with respect to the required accesses from C_i. In failing so, secondly C_i is searched for a candidate subject S_i that has the smallest positive disparity with respect to the required accesses from C_i. Failing to find any candidate at this stage, we get the candidate with the least numerical disparity. Hence the algorithm terminates. A similar argument applies to Algorithm 4.
2. Suppose component C_i has a cover for S with respect to the given access list L. Algorithm 3 uses Algorithm 1 as its first stage. Consequent to the

proof of Algorithm 1 at this stage, Algorithm 3 finds a candidate with the least negative disparity. A similar argument applies for Algorithm 4.

3. Suppose that there is no cover for S in component C_i. Then, Algorithms 3 and 4 shifts to Stage 3, where the component with the least numerical disparity is chosen.[]

5. CONCLUSIONS

We investigate the problem of mapping a federation user to a collection of component users in a tightly coupled federated database. Given a set of component subjects with their access capabilities and an access request list for a federation subject, it may not be always possible to map the latter to a collection of the former as required by subject switching. As a remedy, we suggest two algorithms, the least over permitting, and least under permitting algorithms. We have presented metrics to quantify the degree of disparity between the requested accesses and those that are possible to be granted. We presented two mapping algorithms that minimize these disparities.

ACKNOWLEDGEMENTS

This work was partially supported by National Science Foundation grant CCS-01113515.

REFERENCES

[DeCa97] De Capitani di Vimercati, S. and Samarati, P., *Authorization Specification and Enforcement in Federated Database System,* Journal of Computer Security, Vol.5, No.2, Pages. 155-188, 1997.

[HEIM85] Heimbigner, D., McLeod, D., *A Federated Architecture for Information Management,* ACM Trans. Off. Inf. Syst. 3,3, July 1985.

[JAJO01] Jajodia, S., Samarati, P., Sapino, M. L. and Subrahmanian, V. S. *Flexible Support for Multiple Access Control Policies* in Transactions of Database Systems, June 2001, To appear.

[JONS94] Jonscher, D., Dittrich, K.R., *An Approach For Building Secure Database Federations,* Proceedings of the 20th VLDB Conference, Santiago, Chile, 1994.

[SHET90] Sheth, A.P., Larson, J.A., *Federated Database Systems for Mansaging Distributed, Heterogeneous, and Autonomous Database,* ACM Computing Surveys, Vol.22, No.3, September 1990.

[VEIJ88] Veijalainen, J., and Popescu-Zeletin, R., *Multi-database Systems in ISO/OSI* in Standards in Information Technology and Industrial Control, Malagardis, N., and Williams, T., Eds North-Holland Pages 83-97, 1988.