Providing Dynamic Security Control in a Federated Database

Norbik B. Idris W. Alex Gray Robert F. Churchhouse

Department of Computing Mathematics
University of Wales College of Cardiff
United Kingdom
Email : scmnbi@cm.cf.ac.uk

Abstract

When data is being used in a federated database, the aim is to give a loose coupling of the data in the component databases so that a very dynamic and therefore flexible pattern of data sharing can be established. When security integration is performed this flexibility is curtailed by the resultant security level established at integration time which by default is the least upper bound between candidate security levels. Such overclassification of data implies that there will be authorised users who are debarred at the federation level to access the data. To circumvent this problem there is a need for a dynamic mandate type control for definite periods of the federated system's existence. An approach to establishing such temporary dynamic security control is described in this paper. It is an adaptation of Shamir's method [Sha79] for sharing a secret, and it aims to let users who are debarred at the default security level from access to particular data, gain access to this data under local control if an appropriate combination of current database administrator of the system are prepared to grant the access dynamically.

Proceedings of the 20th VLDB Conference
Santiago, Chile, 1994

1 Introduction

One of the aims of a federated database is to enable the loose and flexible coupling of autonomous databases to make data sharing easier between the constituent databases. As such, in a federated database environment, data is held in local databases which have retained their autonomy within the federation but whose user communities can access data at other sites and levels through the interoperability and integrating features of the federated database environment.

When a database is joined to a federation its security is also integrated with the security provisions in other databases to create a common security level for the data in the federation. The most important step in database security integration is the resolution of conflicts between integratable component databases. One feasible way to resolve such conflict is to choose the least upper bound value among candidate levels which will unfortunately result in overclassification in some candidate security levels. This means that some federated database users will be prevented from accessing data in databases where local security would let them access it. This restriction limits the access flexibility to data in the federation. It is important to realise that while the integrated security layer can act as barrier to the intended flexible sharing of a federated database, it is however a necessary protective measure and should only be overridden if the authorised security officers for each database grant permission for it to be overridden for a fixed period of time or for a specified type of access.

Thus, in a federated database we need two types of security - **static** and **dynamic**. Static security is created when the data is incorporated into the federation, and is permanently associated with it. Although its value can be amended in the future, it is the default security level for the data which applies to all

users unless it is overridden. Dynamic security on the other hand, is the ability to override the static security with a short term change in security level for a particular user granted by a local database's administration. These two types of security are needed in a federated database if full flexibility of access is to be attained while retaining local autonomy.

In our work so far on interoperability in distributed databases [HFG87, FGRC92, QFG92b, RGF89], we have concentrated on using meta-technique to manage the issue of heterogeneity as identified in [BLN86, BG92, B+89, OV91, SLCN88]. This technique has been extended to address the problem of heterogeneity in security [Thu92, IQG93, IT94], modelling it using object-oriented concept as suggested in [Hsi91, Thu90, TR92]. Because the methodology uses object concept it is possible to utilise experiences in other earlier works on security heterogeneity [LOP91, WS87].

In creating security for a federated database, we have looked at the problem of determining the static level of security required for data which is being integrated in the global schema. Following this effort, we then examined how having created the common security level for a federated database we can override its security in a flexible yet safe way to allow users temporary access to data that is debarred to them by the common security. This security override mechanism is known as the dynamic security in that it is a short term change in security for a particular user or group of users.

In this paper, we report on how we have adopted Shamir's method [Sha79] and used it as the basis in creating a dynamic level of security control in a federated database environment. This dynamic layer can co-exist with the static layer and give a more flexible security that reflects the multi-level structure and dynamic character of the data sharing in a federated database. At the same time it also enhances the availability of data to users of the federation subject to authorisation by appropriate owners of the data.

This paper is organised as follows. Section 2 discusses a typical structure of an organisation which has a hiearchical security control. Our methodology to create the federated database is given in Section 3. In Section 4 we summarise Shamir's technique and show how it can be utilised in providing the necessary dynamic security control. This is followed in Section 5 by a description of the way we have chosen to implement Shamir's scheme on top of our federated database environment. Section 6 gives a general appraisal of the dynamic security mechanism, before concluding and briefly touching on future extensions to the work in the Conclusion. Finally, the Appendix gives a mathematical example of key recovery for this system.

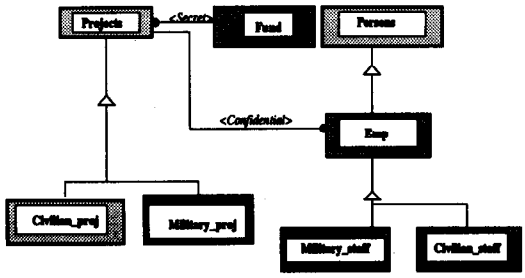
2 Structure of Organisation

We are interested in managing the protection of data in a multi-level secure environment which uses a federated database. In this environment different subjects (process or user) possess different levels of clearance to access the object (data in the database). These objects are also classified according to their security levels. The levels for both the subject and object are determined by the authorities in each component database of the federation. In addition, the security levels for objects accessed at the federated level may also have been resolved during the integration process which created the federation according to some agreed policies [IQG93] to create the default security. This has to be augmented with a flexible dynamic layer which continually changes but reflects the current user needs for temporary access to higher level secure data. This layer is ephemeral and continously changing and evolving. Such security relaxations are granted in this layer to an individual on a limited basis by the administration of a local database.

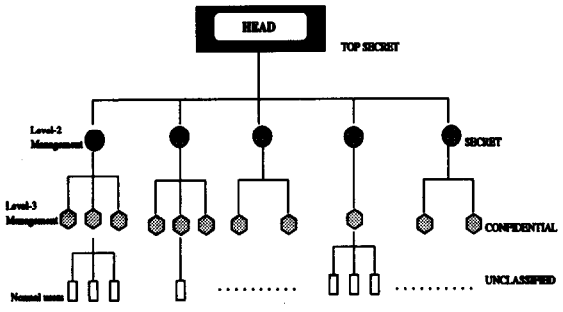
To access an object at the default level in this federation, a subject must have enough security clearance. The clearance level is usually associated with the relative importance of the subject compared to other subjects. It also implies the level of authorisation or responsibility placed upon the subject. This gives rise to a hierarchical control structure. Figure 1(a) shows the security levels given to object classes in a database modelled using OMT [R+91], and Figure 1(b) shows security clearances given to various subjects (database users). The latter forms a hierarchical structure of the staff of the organisation. The intensity of shade in both figures indicates the applicable levels of security i.e the darker the shade, the higher the security level[1].

In a federated database environment which provides static security, access right to a data object defaults to the clearance levels awarded to a subject at the time of registration. However in a federated environment which provides a dynamic mechanism, various configurations of the default clearance levels can be combined to override the static security for a user to be able to access to more secure data on a short term basis. This must nevertheless be done in a safe manner that does not compromise the local autonomy. Before going on to describe the implementation, we will first discuss briefly our integration scheme to create a secure federated database.

[1] In the actual implementation we have used colours to differentiate between the different security levels.



a) OMT model of a database schema with security classifications at Class and Association levels



b) Hierarchical structure of security clearances represented with shades

Figure 1: Correspondence between security levels in a database schema (Figure-a) and a hierarchy of clearance in an organisation (Figure-b). Note : the correspondence is shown by similarity in shades.

3 Creating the Secure Federated Database

A major obstacle in integrating databases in a federation is the problems arising from their independent design. For example, a single concept might have different representations and meanings in different local schemas. What is seen to be classified at one site may not need the same level of security at other sites. Security classification may also change over time. Thus the types and amount of heterogeneity occurring in the component schemas are complex and diverse. Differences in perspectives are due to different designers adopting their own individual viewpoints in modeling the same objects from the application domain.

The overall task of schema integration can be divided into two main parts: *schema analysis*, followed by *schema synthesis*. The first part is started by establishing correspondences and detecting possible conflicts between structural components or the semantic contents of local schemas in the so-called pre-integration phase. These conflicts are resolved by conforming the different schemas so that they can be compared, merged and restructured, before eventually being integrated to form a global schema during schema synthesis. Such activities represent the formal process of designing a global conceptual schema [BLN86, OV91].

To establish correspondences between objects, we need a method to compare their structural components and semantic contents. This will determine the level of equality and relationships between them with a view to forming an integrated equivalent object(s) at the global level. Most O-O database systems support such an 'object equality' comparison through the notion of object identity. Generally, the degree of equality between objects has been categorised into three levels: ***identity-equality***, ***shallow-equality***, and ***deep-equality*** [KC90, QFG92b]. The level of the equality implies the depth of comparison made between the objects. However, note that component database systems may independently assign their object identities. This means that an identical object stored in different component databases may be given diferent object identity and when this happens comparison can at best be based on deep-equality as well as human's confirmation.

Another important point in schema integration is that it cannot rely completely on object equality because there are times when it does not reflect the real world semantics (RWS) for certain types of object relationship [QFG92b, QFG92c]. The best integration can only be achieved when as much as possible of the real world semantics are available before performing it. There have also been proposals to include ***'keys'*** in order to increase the amount of knowledge available about the objects [PG88, Ste89]. Above all, we need a foundation to act as a framework in our integration process, and this is described in the next subsection.

3.1 Real world data modeling and schema integration

Using our earlier experiences we adopt the real world data modelling approach, based on the architecture shown in Figure 2, as the basis for O-O secrecy integration. This architecture is similar to the one proposed by the ANSI/SPARC committee on DBMS [Dat86].

The three levels in the model consist of the **rO** which represents the real world semantics, the **lO** or the logical object level which represents the user-defined classes/objects, and the **iO** which represents the internal implementation of the above two levels. The three levels can be seen as a type hierarchy moving upwards starting from the lowest level, i.e the internal level to the logical level and then to the real world level. The real world level thus represents the highest level of abstraction in the model.

Schema integration is an activity which creates a single integrated global schema from individual component schemas. For the purpose of detecting and resolving semantic ambiguities and conflicts, our experience shows that the information held in the real

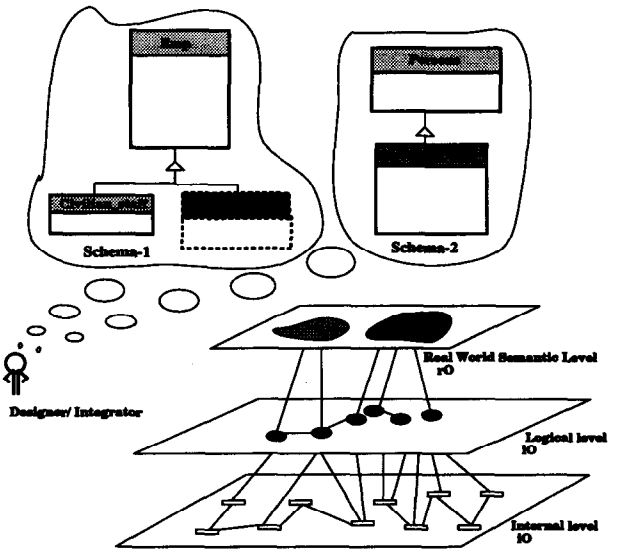


Figure 2: *The 3-level model architecture to facilitate schema integration*

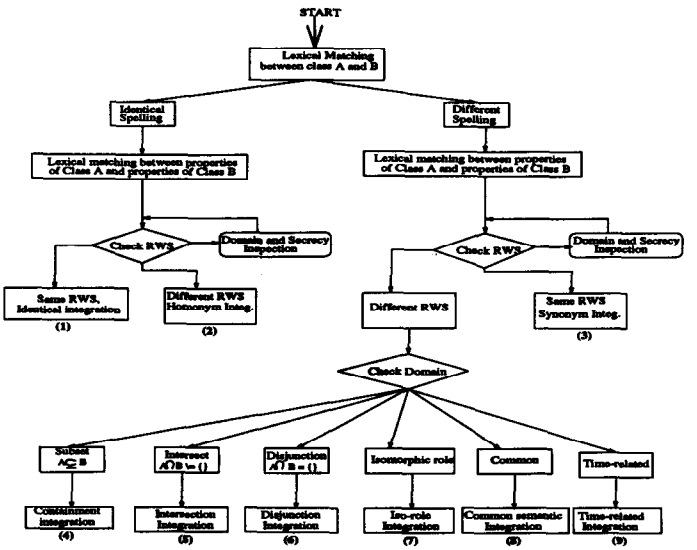


Figure 3: *Routes to show derivation of potential merging classes between two data objects*

world semantic level at the **rO** level emerges as a very important factor in successfully deriving the correct correspondences between objects/classes. Structural ambiguities can be detected and resolved using the **lO** level. The three types of object-equality mentioned earlier are used to derive correspondences between the **lO** and the **iO** levels.

Our experience also shows that introducing the concept of key at the **lO** level has a direct influence on the derivation of the correspondences between the **rO** and the **lO** levels. Here candidate keys have to be identified by the designer, and if such keys are not identical between the two classes, we need to utilise a mapping function between the two domains. In our work this takes the form of a meta-data dictionary. An interactive system has been developed which utilises a framework based on these concepts to assist the designer in constructing the integrated global schema.

3.2 Integration of security via schema integration

To create the secure federated schema, we extend the techniques used to create a global schema for a heterogeneous database as described in [QFG92b, QFG92c] to include security semantics. This integration process involves parsing the specification of the security semantics, building a knowledge base of security features, resolving conflicts among component databases, and making decisions about the static security at the federated level.

In our earlier work [IQG93] we identified the relevant secrecy semantics for objects at the class and association levels, and extended the DDL of an object-relational system (Postgres) to support these semantics. Further we built a parser to check these specifications and accumulate a knowledge base for the model. A graphic display, which employs different graphic attributes, is used to visualise the data model both at the local and federation levels. This would enable an integration manager to have a good overview of the final static model for the federated database and the local databases.

In deriving federated security level, the primary unit of integration is the 'class' since it is the most basic abstract level in the O-O data model. This level also reflects the most natural level at which the database designer logically sees and models the universe of discourse.

Thus in our system, we begin the process of integration by comparing object class names using lexical matching. The possible paths of merging are identified using an and/or graph (as shown in Figure 3) for directing class integration, and uses a supporting inference engine to search its knowledge space to find a solution in a data-driven reasoning approach. Each path through the graph terminates with a rule base which is triggered at the end of the inference process. Finally, the mergeability of the classes needs to be confirmed by the designer. This is done through an interactive process where the designer considers the real world semantics of both classes.

We illustrate only the result of an integration process described above by an example showing security semantics at the object class level. Figure 4, Figure 5 and Figure 6 show secure-OMT models of two participating databases being integrated to get a global

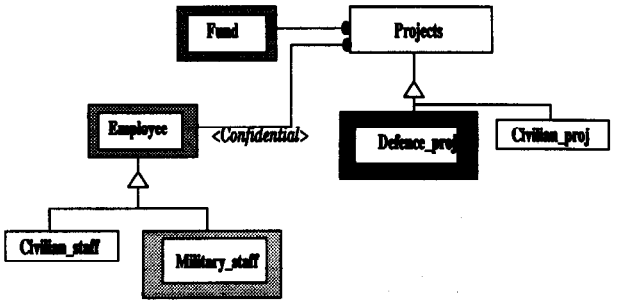


Figure 4: *Secure OMT model for Schema-1*

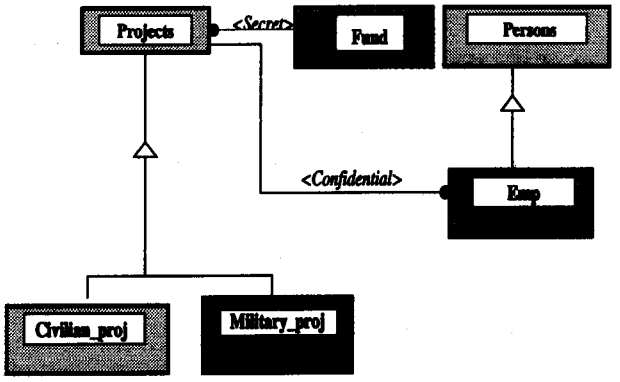


Figure 5: *Secure OMT model for Schema-2*

federal model. Note that the final secrecy level follows a policy of upgrading the lower of the two levels of the component databases [IQG93]. For example, since classes 'Projects' in both schema are identical and are thus mergeable but with conflicting secrecy levels, the global level will take the secrecy level of 'Projects' in Schema-2. Untagged class such as 'Projects' in schema-1 is by default given secrecy level 'Unclassified'.

4 Shamir's Scheme

As mentioned earlier, the need for secret sharing scheme in our work is to circumvent the problem resulting from overclassification of security at the federated level when integrating pre-existing databases. In this section we discuss how the scheme can be adapted and at the same time relate the federated database with the organisation's structure shown in Figure 1(b).

A secret sharing scheme such as Shamir's has the following notion. Given a secret piece of information, we would like to construct n related pieces of information in such a way that any set of k of them will suffice to recover the original secret. However, no subset of $k-1$, or fewer, will reveal it. The pieces of information, called *shares*, can then be distributed privately via a secure channel to all the n participants in the

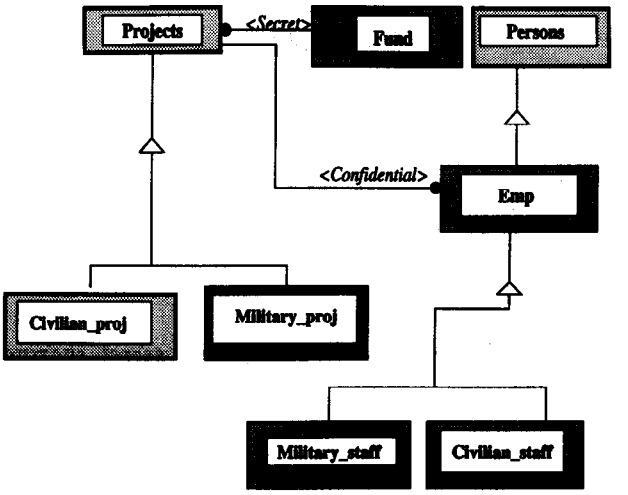


Figure 6: *Integrated secure OMT model of the federated schema*

scheme. This general model is either called k-out-of-n shared secret scheme or a (k, n)-threshold scheme.

Thus, any concurrence of less than k of the participants, including anyone unauthorised (and thus possessing no share), will have no better chance of computing the secret than will an outsider without any privileged information at all. In this sense Shamir's basic scheme is said to be **perfect** since each participant having a share has no advantage over outsiders in guessing the secret.

Apart from its perfect security, Shamir's scheme is also easy to understand and is versatile. When used together with an arithmetic module, it provides one with an efficient method for generating shares, decoding to recover the secret and changing the access structure of the system. Dawson and Donovan [DD93] showed that this basic scheme can also be adapted to solve many other problems associated with sharing a secret. One of these problems is managing multi-level shared control which is related to our work in database integration [HFG87, QFG92a, QFG92c, QFG92b, RFG91, RGF89].

4.1 Adapting Shamir's scheme to provide dynamic security in a federated database

A dynamic structure is one which allows various kinds of control to access the information in the organisation. The levels in an organisation's structure as shown in Figure 1(b) indicate the relative importance between these users. Generally, the more senior a user, the more clearance he should have. Shamir recognised this problem earlier and suggested a way to accomplish it by giving more shares to participants who have higher levels of clearance [Sim92]. It is not difficult to see that this method is rather awkward in the sense

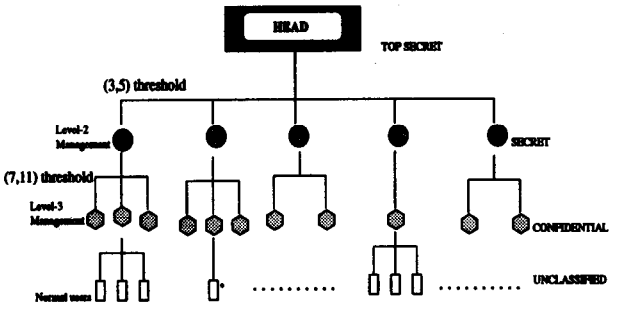


Figure 7: *Hierarchical structure with secret sharing thresholds.*

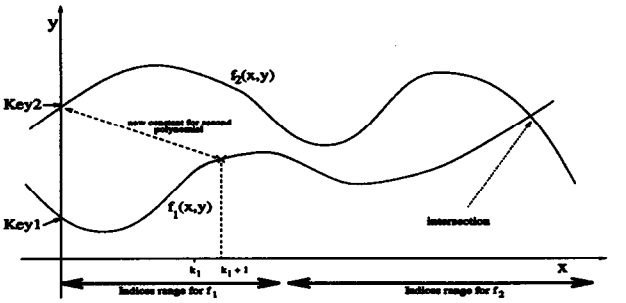


Figure 8: *Polynomials construction for mixed multi-level shared scheme*

that users with more priviliges have to memorise more than one share. This scheme would therefore not be welcomed by higher management levels in an organisation who would need to memorise and then use all their shares when granting override access to the federated database.

There are various kinds of multi-level structure which may be adopted by an organisation as its access structure. However, we will discuss only two kinds of multi-level scheme here.

4.1.1 Standard multi-level control

A typical structure of an organisation is shown in Figure 1(b). Let us reproduce the structure, but this time with additions of security threshold parameters. This is shown in Figure 7. Now, assume that the Head is allowed to grant access to all the data in the federated database. Thus he does not need any secret key as long as he can be positively identified by the security system.

Going down to the second level, suppose the policy decides that the threshold here is $(3, 5)$, and in the third level $(7, 11)$. Note that this policy is quite consistent in the sense that staff from at least three of the five departments at the second level are required to concur in order to grant an enhanced clearance to a user. This reflects the amount of responsibility shared by them. We also assume that there is no privilege given to any user with clearance UNCLASSIFIED.

The above requirement can be achieved by adapting Shamir's scheme as follows. We need two different polynomials, but with a similar secret constant. Thus, let the two polynomials be,

$$f_2(x) = K + a_1 x + a_2 x^2 \qquad (1)$$

$$f_3(x) = K + b_1 x + b_2 x^2 + b_3 x^3 \ldots + b_6 x^6 \qquad (2)$$

It is not difficult to see that since the secret constant, K, is the same in both polynomials, any fulfilment of the threshold values $k_2 = 3$ or $k_3 = 7$ gains the enhancement. In other words, 3 second level participants or 7 third level ones are equivalent to the Head. This simple example shows the flexibility of the scheme. However, the access structure can still be configured to accept other types of policy. A slightly more complicated access structure is needed to allow for what we call mixed multi-level sharing, to create a mixed multi level control.

4.1.2 Mixed multi-level control

A more dynamic control is required in this situation as follows and we refer to Figure 7 again to explain this. Suppose there exist L levels of privileges and k_l shares are needed at level l to recover the secret key. However if a share from level l is not available, it can be accommodated by k_{l-1} shares at level $(l-1)$; and so on.

This enhancement is easy to implement by adapting Shamir's basic scheme as follows. Let there be (n_l) participants at level l subject to threshold system (k_l, n_l). Now, if there are only $k_l - 1$ participants, the missing share can be replaced by k_{l-1} shares from one level down. To achieve this, a polynomial, $f_l^{k_l-1}$, is constructed to handle threshold requirement at level k_l. Thus the secret key is obtained when $x = 0$ for this polynomial. Then we construct another polynomial $f_{l-1}^{k_{l-1}-1}$, but with its constant being the value $f_l^{k_l-1}(n_l + 1)$. Note that later on in the recovery attempt, this is the point which accommodates the group from the lower level. This construction is shown in Figure 8 with two polynomials.

There is a slight problem with this scheme in that there is a small chance that the share $f_l(x_i) = f_{l-1}(x_i)$ (intersection in Figure 8). If this happens $p_{l-1,i}$ can masquerade as $p_{l,i}$. To handle this Dawson and Donovan suggest a check that is made before shares are distributed. A better way is to insist that the indices for different privilege levels are mutually exclusive. Some 'gaps' (reserved indices) can also be introduced to ac-

commodate new participants at a particular level in future. This alternative is practical since L, n, and k are usually small (about 50).

The above are only two of the many cases which can be possibly modeled under the concept of multi-level shared control. They should be sufficient at this stage to give an understanding of the flexibility that can be achieved. The examples have also been simplified by giving multi-levels only to the object classes. The picture would be more complicated if we were to consider the levels of secrecy for other deeper semantics in the OMT.

Simmons [Sim92] notes that a multi-level scheme such as this where there exists more than a single class of capability cannot be perfect. This is because such a scheme disagrees with the definition of **perfect** given earlier in the sense that among all the authorised users, some are more privileged than others by virtue of their relative importance. However, in practice, such differences in capability are the norms in organisations, and we note that the disagreement is only due to the definition. In other words this form of perfect level of security where some users are indeed more privileged than others is what is required in multi-level shared control. If the group of participants which can replace a share in the level above it is seen as another participant in that level, the situation fits the definition again.

5 Security in a Federated Database System

Up until now we have shown how the theory and use of the adapted scheme could fit easily into the hierarchical management structure often found in an organisation using a federated database. In this section we explain very briefly the implementation of the mechanism on our system.

Simmons [Sim92] discusses in detail how to proceed with implementing a general scheme, while Dawson and Donovan [DD93] list down the basic requirements for a robust key management system to be accepted. We will refer to these in considering the appropriateness of our implementation of dynamic security in a federated environment with default static security created by integration.

5.1 Meta-security

The use of meta-techniques for schema integration is extended in this work in that it is used to create an integrated security knowledge base for all the privileged users in the system. This is done so that the security mechanism fits easily into the integration paradigm established so far [IQG93, QFG92a, QFG92c, QFG92b] for the static layer of security in the federated database. This is achieved by the Prolog fact:

clearance(User, Default, Temporal).

where '**User**' is a user's login name and '**Default**' is his initial clearance level. These would be determined when the user first registers into the system. '**Temporal**' is the currently upgraded clearance level for this particular interaction with the federated database. A temporal value of *null* indicates no new privilege granted.

The '**clearance**' fact above can be extended to build more knowledge with respect to the modes of operation (read, write, append) allowed for a subject on an object in the database. In essence, this security dictionary is similar to an access matrix used to manage object security in an operating system [Lan81]. Ways of extending this will become another interesting aspect of this approach to be studied in future.

The knowledge base is referred to and checked by the federated system every time a classified item of data is to be accessed. In this sense the mechanism forms a security layer with 'variable thickness' (due to the flexibility for enhancement), on top of the federated environment's static security layer.

5.2 Generating and distributing the keys

To construct a polynomial, a good random generator is needed to generate the secret key and coefficients. Our implementation uses the 'date:hour:minute:seconds' concatenation of the system's clock output to ensure an unpredictable seed for the generator. This guarantees that every polynomial constructed will be different and have random secret key and coefficients.

Having decided on the threshold values and number of participants to be included in the dynamic mechanism, they are then ordered, according to a simple serial ranking. The shares are then calculated using the rank number as the x-values. They are then distributed, assuming the existence of a secure channel to the users of the databases who are officers of the organisation.

5.3 Recovering the secret

For simplicity, we will describe this using the example in Section 4.1.1. Assume that the participants in level 2 in Figure 7 are pooling their shares to get privilege as the Head. According to the policy statement any 3 of them would suffice. A missing one is however replaceable by any 7 participants in level 3 in the organisation.

The system first requests each participant to enter his index and share value. Using the index, the system

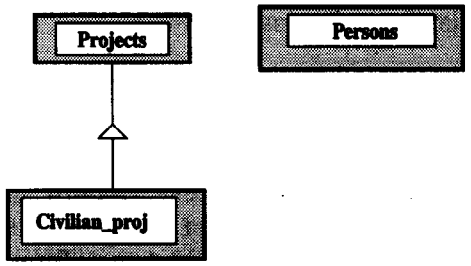


Figure 9: *Classes cleared to CONFIDENTIAL level before invoking shared control*

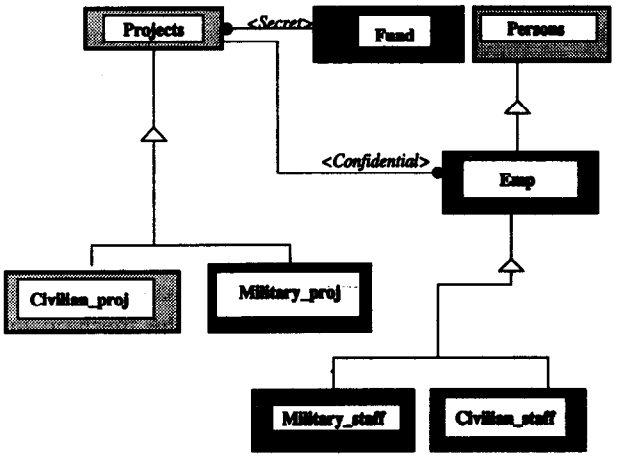


Figure 10: *Classes cleared after invoking shared control*

is able to identify which enhancement can be achieved by each index-share pair. We then construct k_2 simultaneous equations of the form f_2, and k_3 equations for f_3. Thus, suppose $p_{2,1}$, $p_{2,2}$ and $p_{2,4}$ pool their shares, three equations will be constructed as follows:

$$f_{2,1} = K + a_{2,1}(1) + a_{2,2,}(1)^2 \quad (3)$$

$$f_{2,2} = K + a_{2,1}(2) + a_{2,2}(2)^2 \quad (4)$$

$$f_{2,4} = K + a_{2,1}(4) + a_{2,2}(4)^2 \quad (5)$$

These equations are then solved using an arithmetic package to obtain the secret key and coefficients. The same procedure would be followed in order to solve other equations. Once the key is recovered, the system will update the security knowledge base to indicate a temporal enhancement of privilege for user requesting it for a database.

5.4 Example of a temporal enhancement

Once the secret key is recovered, the required user enhancement is granted by the participants involved in pooling their shares. We would like to emphasise that this enhancement is transient in nature and granted to a particular user. It must be invoked with a proper system log. This is discussed later in Section 6.1. The OMT models in Figure 9 and 10 show a database model of the multi-level secure database discussed in Section 7 and shown earlier in Figure 1(a), in terms of the classes which are cleared to a participant who has been given a default clearance of CONFIDENTIAL. Thus by default, he is only allowed access to the classes shown in Figure 9. If a temporal enhancement appeal is successful, he sees a bigger database environment as shown in Figure 10. Such an enhancement can only be made for the user requesting it by a group of users whose shares release the secret for the database.

6 Appraisal of Dynamic Security Control

Our implementation fulfils the basic requirements of a good robust system. The Lagrange interpolation method has a running order of $O(k\,log^2\,k)$ to recover the key. Denning notes that it is possible to achieve $O(k)$ using another algorithm for secret sharing by Asmuth and Bloom (see [Den82], page 183). However, she also notes that for small threshold values the difference will not be significant. Furthermore, as argued earlier, Shamir's scheme has many advantages compared to other algorithms including the one by Asmuth and Bloom.

At this stage we have not worked on the user-interface to initiate an invocation of the mechanism. A number of options are however available. Starting from the simplest way, a user wanting to enhance his clearance may contact (by telephone) privilege users and ask for their contribution towards the clearance into a pool. Alternatively, a message broadcast can be made to all privilege users who are logged on to the federated system, starting perhaps from the level which requires the least threshold value. Certainly, the latter is more preferable due to its speed. In either case, the system can be made to automatically manage its log and audit trials. These approaches will mean a user's clearance for data within a particular database can be altered by the database's administration for a limited time period. This preserves local autonomy and enables a dynamic security layer to be temporarily introduced.

If a set of users is available who can grant this temporal clearance they are contacted at this time to determine whether they will grant it. They will then simply enter their shares at their respective terminals for the security mechanism to recover the secret key and grant the requester the clearance. It may be possible that the set of users are distributed physically on

a sparse network.

Before concluding we suggest some further practical considerations about the system.

6.1 Practical considerations

- Size of secret key
  The size of the secret may be limited to 4 digits for easy memorisation by the participants. The choice of 4-digits should be safe enough as it is used in the banking sector to dispense secure PIN numbers, provided there is a limit on the number of attempts allowed (as discussed below). In reality, some of the shares will have less than 4 digits and will be much easier to memorise.

- Limited attempts at granting security upgrade
  In practice, attempts to invoke the threshold procedure must also be limited, in the same way as the automatic bank teller machine is supposed to swallow a card after three unsuccessful attempts. In any attempt, the system will insist entry of enough shares to be used in the recovery. This will avoid a user trying to simply guess the secret key and thus bypassing the recovery process.

- Choice of good random coefficients
  This is possible by using a good random number generator, and guaranteeing a different seed every time the system is invoked.

- System log and audit
  Like a normal security procedure, attempts to invoke the dynamic mechanism must be recorded for monitoring and auditing purposes. This helps to alert the security officer against any illegal access attempts.

7 Conclusion

The main aim for integrating databases in a federation is to pool and share information. When the integration process takes account of pre-existing security requirements there is a problem of overclassification at the federated level and mechanisms are needed to support security management particularly if flexibility of access is to be maintained.

In order to take full advantage of the created secure integrated environment of the federated database, and to address new needs in information accessing, we have extended the security control by implementing a dynamic mechanism which serves to enhance the standard static security management which is created when data is incorporated into the federation. This dynamic layer is created in such a way that local autonomy of the constituent databases is preserved while giving individual users a short lived alteration to their security clearance.

The prototype system we have built shows that there are many advantages obtained from this flexibility. So far Shamir's scheme has been easy to implement due mainly to its established algebraic foundation. More work needs to be done in looking into the security level and the improvements gained in terms of availability of information as a result of this enhancement to the federated database's security. On the other hand, we conjecture that the granting of transient privilege will also introduce a new form of inference problem. It will also need temporal and other constraints to limit the period and type of access granted dynamically.

Finally, it is up to the organisation to decide whether it is satisfied with a default standard security management of its federated database, or it needs to establish an enhanced dynamic layer of security which would then be available to users needing temporary access to items debarred by the federated database's default security layer.

8 Appendix - Example of Key recovery

Consider users with share indices {1, 3, 5, 6} pooling their shares of {19, 5, 1, 34} respectively to recover the key in a secret sharing system with threshold (4, 7) working in *modulus* 37. Thus the required polynomial is a cubic. Let it be

$$f(x) = K + ax + bx^2 + cx^3$$

then

$$f(1) = K + a + b + c \equiv 19\,(mod\,37) \quad (6)$$

$$f(3) = K + 3a + 9b + 27c \equiv 5(mod\,37) \quad (7)$$

$$f(5) = K + 5a + 25b + 125c \equiv 1(mod\,37) \quad (8)$$

$$f(6) = K + 6a + 36b + 216c \equiv 34(mod\,37) \quad (9)$$

From the above 4 equations we have :

$$(7) - (6) \Rightarrow 2a + 8b + 26c \equiv -14 \equiv 23(37) \quad (10)$$

$$(8) - (7) \Rightarrow 2a + 16b + 98c \equiv -4 \equiv 33(37) \quad (11)$$

$$(9) - (8) \Rightarrow a + 11b + 91c \equiv 33(37) \quad (12)$$

so :

$$(11) - (10) \Rightarrow 8b + 72c \equiv 10(37) \quad (13)$$

$$2(12) - (11) \Rightarrow 6b + 84c \equiv 33(37) \quad (14)$$

Reduce Equations (13) and (14) (*mod* 37) :

$$8b - 2c \equiv 10(37) \quad (15)$$

$$6b + 10c \equiv 33(37) \quad (16)$$

Now multiply Equation (15) by 5 and add to Equation (16),

$$40b - 10c \equiv 50 \equiv 13(37)$$

i.e :

$$46b \equiv 46(37)$$

or $b = 1$ which since $8b - 2c \equiv 10(37)$ gives $c = -1$ and since $2a + 8b + 26c \equiv 23(37)$ i.e,

$$2a + 8b - 26 \equiv 23(37)$$

$$2a \equiv 41 \equiv 4(37)$$

i.e

$$a = 2$$

Now,

$$K + a + b + c \equiv 19(37)$$

i.e

$$K + 2 + 1 - 1 \equiv 19(37)$$

i.e

$$K = 17$$

Hence the secret key is 17.

References

[B+89] E. Bertino et al. Integration of Heterogeneous Database Applications through an Object-Oriented Interface. *Information Systems*, IS-14(5), 1989.

[BG92] David Bell and Jane Grimson. *Distributed Database Systems.* Addison-Wesley, 1992.

[BLN86] C. Batini, M. Lenzerini, and S. Navathe. A comparative analysis of methodologies for database schema integration. *ACM Computing Surveys*, 18(4), December 1986.

[Dat86] C. J. Date. *An Introduction to Database Systems.* Addison-Wesley, 1986.

[DD93] Ed Dawson and Diane Donovan. Shamir's Scheme Says It All. (research report), Information Security Research Centre, Queensland University of Technology, Australia, 1993.

[Den82] D. E Denning. *Cryptography and Data Security.* Addison Wesley, 1982.

[FGRC92] N J Fiddian, W A Gray, A Ramfos, and A Cooke. Database meta-translation technology: integration, status and application. *Database Technology, Pergamon Press Ltd.*, 4(4):259–263, 1992.

[HFG87] D. I. Howells, N. J. Fiddian, and W. A. Gray. A source-to source meta-translation system for relational query languages. In *Proceedings of 13th International Conference on Very Large Databases (VLDB)*, pages 227–234, 1987.

[Hsi91] D. K. Hsiao. Object-oriented approach to security policies and their access controls for database management. Technical report, Naval Postgraduate School, Monterey, California, USA, September 1991.

[IQG93] N. B. Idris, M. A. Qutaishat, and W. A. Gray. Integration of secrecy features in a federated database environment. In Thomas Keefe, editor, *Proceedings on DATABASE SECURITY VII: Status and Prospects, Results of IFIP WG 11.3 Workshop on Database Security*, pages 84–106. IFIP, August 1993.

[IT94] Norbik Idris and Bhavani Thuraisingham. Object-oriented Model for Federated Database Environments. In *Proceedings of IMSS Conference on Intelligent Information Systems, Washington DC*, June 1994.

[KC90] S. Khoshafian and G. Copeland. Object identity. In S. Zdonik and D. Mair, editors, *Readings in Object-oriented Database Systems.* Morgan Kauffmann, 1990.

[Lan81] C. E Landwehr. Formal Models of Computer Security. *ACM Computing Surveys SIGOPS*, pages 247–278, Sept 1981.

[LOP91] H. Lu, B. C. Ooi, and H. H. Pang. Multilevel Security Control in Multidatabase management systems. In *Proceedings*

1st IEEE International Workshop on Interoperability in Multidatabase Systems (IMS91), pages 359–363, Kyoto Japan, April 1991.

[OV91] M. Tamer Ozsu and Patrick Valduriez. *Principles. of Distributed Database Systems.* Prentice-Hall, 1991.

[PG88] N. Paton and P. Gray. Identifications of database objects by key. In *Proceedings of 2nd International Workshop on Object-oriented Database Systems*, Bad Munster am Steinberg, FRG, Spetember 1988.

[QFG92a] M. A. Qutaishat, N. J. Fiddian, and W. A. Gray. Association merging in a schema meta integration system for a heterogeneous object-oriented database environment. In P. M. D. Gray and R. J. Lucas, editors, *Advanced Database Systems, 10th British National Conference on Databases, BNCOD 10*, pages 209–226, Aberdeen, Scotland, July 1992. Springer Verlag.

[QFG92b] M. A. Qutaishat, N. J. Fiddian, and W. A. Gray. A schema meta integration system for a heterogeneous object-oriented database environment- objectives and overview. In Jarko Leponiemi, editor, *NordDATA '92 Conference*, pages 74–92, Tampere, Finland, June 1992. PITKY, Finland.

[QFG92c] M. A. Qutaishat, N. J. Fiddian, and W. A. Gray. A schema meta integration system for a heterogeneous object-oriented database environment - Impementation in Prolog. In *Proceedings of 1st International Conference on the Practical Application of Prolog*, London, 1992.

[R+91] James Rumbaugh et al. *Object-oriented Modeling and Design.* Prentice-Hall, 1991.

[RFG91] A. Ramfos, N. J. Fiddian, and W. A. Gray. A meta-translation system for object-oriented to relational schema translations. In M. S. Jackson and A. E Robinson, editors, *Proceedings 9th British National Conference on Databases*, pages 245–268, 1991.

[RGF89] A. Ramfos, W. A. Gray, and N. J. Fiddian. Object-oriented to relational inter-schema meta-translation. In *Proceedings of NSF International Workshop on Heterogeneous Databases, Northwestern University, Illinois*, pages 22–38, 1989.

[Sha79] Adi Shamir. How to Share a Secret. *Communication of the ACM*, 22(11):612–613, 1979.

[Sim92] G. J. Simmons. An Introduction to Shared Secrets and/or Shared Control Schemes and Their Applications. In Gustavus J. Simmons, editor, *Contemporary Cryptology - The Science of Information Integrity*, pages 441–498. IEEE Press, New York, 1992.

[SLCN88] A. Sheth, J. Larson, A. Cornelio, and S. Navathe. A tool for integrating conceptual schemas and user views. In *Proceedings of 4th International Conference on Data Engineering*, 1988.

[Ste89] J. Stein. Mistaken identity (preliminary report). In *Proceedings of 2nd International Workshop on Database programming Languages*, Oregon, USA, 1989.

[Thu90] Bhavani Thuraisingham. Security in object-oriented database systems. *Journal of Object Oriented Programming*, 2(6):18–25, March/April 1990.

[Thu92] Bhavani Thuraisingham. Secure interoperability of trusted database management systems. *ACM SIGSAC - Special Interest Group on Security, Audit & Control*, 10(2 & 3):–, Spring/Summer 1992.

[TR92] Bhavani Thuraisingham and Harvey H. Rubinovitz. Multilevel security issues in distributed database management systems-iii. *Computers & Security*, 11(7):661–674, 1992.

[WS87] C. Y. Wang and D. L. Spooner. Access Control in a Heterogeneous Distributed Database System. In *Proceeding 6th IEEE Symposium on Distributed Software and Database Systems*, pages 84–92, Williamsburg VA, March 1987.