Asian Journal of Information Technology 5 (4) : 418-428, 2006

An Effective Security Interoperability Archetype for Secure Multilevel Databases

Awad M. Awadelkarim and Norbik Bashah Idris
Center for Advanced Software Engineering-The Information Security Group
CASE-ISG: Faculty of Computer Science and Information Systems,
University Technology Malaysia (UTM-KL, City Campus)
Jalan Semarak, 54100 Kuala Lumpur, Malaysia

Abstract: This study addresses the problem of security interoperability in an integrated heterogeneous database environment. It presents a universal framework of security management model for integrating and managing security features namely label-based access control of integrated heterogeneous relational legacy databases. The model is anchored in Rule-Based Algorithm and Global Security Policies, using XML as integration medium with richer features and above. Validation and implementation of the expedient model are also provided including the designated-principal benchmarks.

Key words: Database security, label-based access control, integration of databases, legacy databases, MAC, XML

INTRODUCTION

Database security addresses the problem of protecting sensitive information against intentional or accidental threats. It consists of granting the basic security services: confidentiality, integrity, authentication, authorization and availability. A significant security mechanism associated with authorization and confidentiality issues is access control. Correspondingly, the main purpose of database integration is to merge and share the information, besides, integration of heterogeneous legacy databases is important, as information is often required to be synthesized and aggregated especially for high-level management. This paper presents a Label-Based Security Management Model (LBSMM) for integrating security features of relational legacy databases. In such environment and from the security point of view, such integration presents several interesting problems and it effectively requires a new post-integration security management. Firstly, the autonomous legacy databases are assumed heterogeneous and as such have incompatible security features. Queries that may involve data from the legacy databases need to be managed by different security management systems. In addition, there are specified security features (labels) for each database schema and integration may require a new post-integration security management, i.e., what happened to the security features when those schemes are integrated: how the security attributes (labels) are applied, implemented, inherited and achieved when databases are integrated. In other word, our model addresses the integration problem of the security features when heterogeneous legacy databases are integrated.

The most vital part in the integration process is the resolution of conflicts between heterogeneous security sensitivity labels at the global level. Consequently, the resolved solution can lead to inconsistency of the security features, where subjects can lose access rights to objects after the integration. This situation requires a mechanism that can handle and manage the integrated labels at the global level. Thus, the LBSMM offers two types of security management namely Default Security Management (DSM) and Exception-handling Security Management (ESM) to fulfill such environment requirements and at the same time, to become an effective management method that provides secure and consistent access at the global level.

RELATED WORK

Numerous research publications have addressed the problem of security conflicts and proposed various general MAC-based (or label-based) security models[1-6]. More to the point, lately there has been a considerable interest in environment that support multiple and complex access control polices and use of XML as

Corresponding Author: Awad M. Awadelkarim, Center for Advanced Software Engineering, The Information Security Group, CASE-ISG: Faculty of Computer Science and Information Systems, University Technology Malaysia (UTM-KL, City Campus), Jalan Semarak, 54100 Kuala Lumpur, Malaysia

integration medium[7-12]. Previous work has mostly been tackled within the frame of federated databases with a global schema that secluded by access control and restrictions[13]. Thus, different proposals have been addressed various problems in this framework such as[10,13-17]. Alternatively, there has been considerable research interest in language-based approaches to access control[7,12,18-22] and the main goal is to provide a language that can support multiple access-control policies and achieve separation of policies from mechanisms. In addition, various papers addressed the labeling and re-labeling of the multilevel security policies in various environments. For instance, Li, Qian and Simon[2] introduced a security model of dynamic labeling providing a tiered approach to verification, including one policy that permits high-level subjects to make re-labeling requests on low-level objects. Additionally, Simon[14] proposes a schema whereby the secure canonical upgrade policies in multilevel secure object-oriented database management system stores can be supported within the Message Filter Model. Also, Lantian and Andrew[5] present an expressive language-based mechanism for reasoning about dynamic security labels; it formalizes computation and static checking of dynamic labels in a type system and proves its noninterference. Also, in[4], authors introduced a model for specifying security policies that deal with multi-policy security in large open distributed systems. Though, most of them cannot be directly used in the integrated environment. As well, e ven though high-level subject can alter the security level of low-level object, such changing is not recognized at a low-level. Besides, various models provide more sophisticated conflict resolution policies[1,10,11,15,23-25].

THE LBSMM ARCHITECTURE

When the various databases are required to integrate, heterogeneity becomes the core and focal concern. With the several variety of heterogeneity, our LBSMM handles the security heterogeneity that is label-based security features to ensure secured and homogeneous access at the global level. The LBSMM architecture consists of five basic components namely Objects' Security Features Integration (OSFI) unit, Integrated Security Features Management (ISFM) unit, Rule-Based Algorithm (RBA), Global Security Policies (GSP) and XML Repository. The three main players in the

Table 1: Typical label scheme

Industry	Sensitivity labels	Short numeric form used by the LBSMM
Defense/Military	Top secret, secret, confidential and unclassified	4, 3, 2 and 1
Financial services	Acquisition, corporate, client and operations	4, 3, 2 and 1
Judicial	National -security, sensitive and public	3, 2 and 1
Health care	Primary-physician, patient-confidential and patient release	3, 2 and 1
Business to business	Trade-secret, proprietary, company-confidential and public	4, 3, 2 and 1
HR and other systems	Highly-sensitive, sensitive, confidential and public	4, 3, 2 and 1

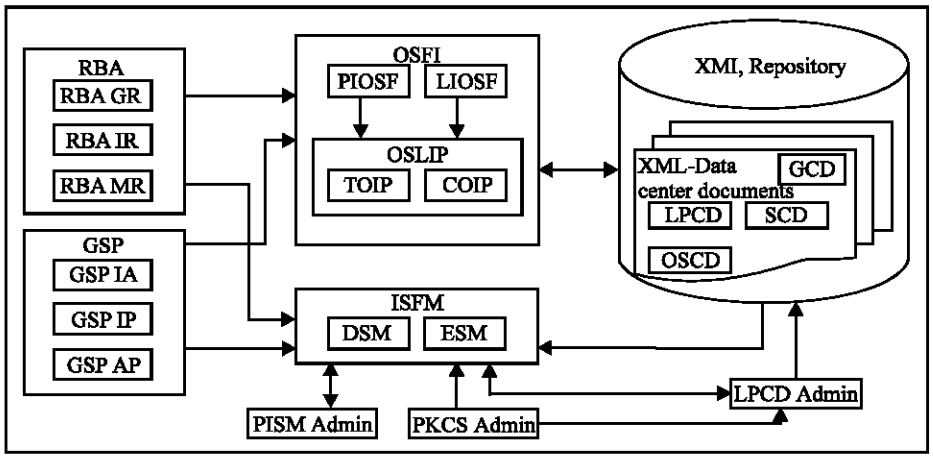

Fig. 1: The LBSMM architecture

```
<databaseA>
   <rowobj>
            <objid> … </objid>
            <objname> … </objname>
            <objtype> … </objtype>
            <objdesc> … </objdesc>
            <objsenlab> … </objsenlab>
            <objlink> … </objlink>
            <objrem> … </objrem>
   </rowobj>
  …
</databaseA>
```

Object Security Information
Obj_id: object identification indicates database, table, and row/ column that the object belongs to
Obj_name: stand for object name.
Obj_type: object type (char, integer, and so forth)
Obj_desc: object description (describe the object and its semantic).
Obj_sen_label: object's sensitivity label (values: 1,2,3, or 4).
Obj_link: the link between the object and the others (optional).
Rem: general remark about the object (optional).

Fig. 2: Object security information (XML DCD)

```
<databaseA>
   <rowsubj>
            <subjid> … </subjid>
            <subjtag> … </subjtag>
            <subjname> … </subjname>
            <subjdesc> … </subjdesc>
            <subjsenlab> … </subjsenlab>
            <subjrem> … </subjrem>
   </rowsubj>
            …
</databaseA>
```

Subject Security Information:
Subj_id: subject identification.
Subj_tag: subject tag denotes local database side that it belongs to.
Subj_name: stand for subject name.
Subj_desc: subject description (position/ranking) (optional).
Subj_sen_label: subject's sensitivity label (values: 1,2,3, or 4).
Rem: general remark about the subject (optional).

Figure 3: Subject security information (X-DCD)

```
<scdglobal>
   <databaseA>
  <objtab>
            <row>
  <objid> … </objid>
                        <objname> … </objname>
                        <objtype> … </objtype>
                        <objdesc> … </objdesc>
                        <objsenlab> … </objsenlab>
                        <objlink> … </objlink>
                        <objrem> … </objrem>
            </row>
                        …
    </objtab>

   <subjtab>
```

```
      <subjtab>
                  <row>
                  <subjid> … </subjid>
                  <subjtag> … </subjtag>
                  <subjname> … </subjname>
                  <subjdesc> … </subjdesc>
                  <subjsenlab> … </subjsenlab>
                  <subjrem> … </subjrem>
            </row>
      </subjtab>
   </databaseA>
   <databaseB>
   …
   </databaseB>
   …
   </scdglobal>
```

Fig. 4: The SCD schemetic

```
<oscdglobal>
   <grow>
  <gobjid> … </gobjid>
                        <gobjlevelintg> … </gobjlevelintg>
                        <gobjsenlab> … </gobjsenlab>
                        <gobjtag> … </gobjtag>
                        <gobjrem> … </gobjrem>
   </grow>
   …
</oscdglobal>
```

Objects' Security Centric Document:
gobjid: global object ID indicates the original integrated objects.
gobjlevelintg: global-objects'-level of integration
(values: 0 for physical integration and 1 for logical integration).
gobjsenlab: global objects' sensitivity label.
gobjtag: global-objects' tag.
gobjrem: general remark about the object (optional).

Fig. 5: The OSCD schematic

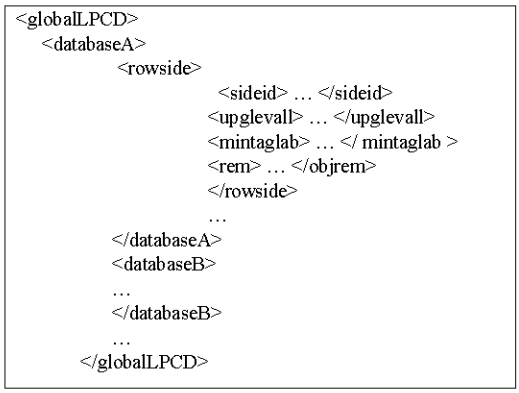

Fig. 6: The global LPCD schematic

Thus, the LBSMM controls and manages the objects access in three dimensions: Object-Label denotes and specifies the sensitivity of the object and determines the criteria that must be met for a subject to access that object. Subject-Label denotes the label authorization assigned to a subject. Exception-Label denotes an exception label that can be given to a particular subject to access certain object, which is beyond the subject's default-label authorization in an exceptional session. Various industries use different sensitivity-label schemas to implement label-based access control. Table 1 illustrates a variety of typical label scheme applied by industry and shows the short numeric form used by our LBSMM to indicate the exact sensitivity-label. Thus, higher numbers indicate more sensitive and lower numbers indicate less sensitive. The same numeric form (sequential number) used for subject sensitivity labels. This numeric form meets the LBSMM design and implementation requirements and at the same time, besides, it supports the model to be simply adopted and customized by any industry.

As known, the main purpose of the database integration is to merge and share the information. Thus, very simple example to illustrate the overall scenario of the problem in this way: assume there is three autonomous and heterogeneous databases that required for integration. They have also different security features, requirements and specifications. In addition, their subjects and objects are labeled into different sensitivity labels. Eventually, the scenario like so, numerous subjects have different level of authorizations (labels/clearance) request to access objects that are labeled according to their security-sensitivity. Thus, the LBSMM is concerned in integrating and then managing the integrated security features without taking into account the technique used to enforce the label-based access control at each local database site and its implementation mechanism as well. However, we also believe that such integration and management can never be fully automated, but can be acceptable with reasonable and consistent administration at the global level. Thus, we maximize the automation process in our model to such level that ensures appropriate security and interoperability in the integrated environment. Therefore, the LBSMM provides both Default and Exception-handling security management to fulfill the environment requirements. Normally, the default treatment presents the subjects' fixed-security labels obtained during the integration process, which are derived from RBA and GSP. When the exception treatment presents the temporary security label given to a subject for particular session or special-query handling upon a request and its approval. In general, Fig. 1 demonstrates and visualizes the overall scenario of the proposed LBSMM architecture and shows the relationships and the correlations among the model components. Therefore, now we move from the big picture to describe the details of each component in the proposed model architecture.

The XML Repository: In our model, XML repositories are used to store the all security information associated to the subjects and objects that mapped from each individual database, i.e., all the relevant security information and semantics of such security features will be stored in the XML repositories. Therefore, XML repositories support and facilitate the integration process and treatment. Additionally, the integrated security features also will be stored in such repositories. Subsequently, such repositories are involved in the post-integration security management procedure as well. In simple terms, the XML repositories can be visualized as the security features' knowledge (information) base of our proposed model.

Security Centric Document (SCD): SCD contains all the security information about the subjects and objects at the global level. During the integration process, the LBSMM maps all information related to the label-security features that provided by each local database site to the equivalent XML data-centric-documents (Fig. 2. and 3). Then, LBSMM integrates the all mapped-centric documents based on RBA and GSP in one comprehensive (global) structured document that is SCD (Fig. 4.).

Object Security Centric Document (OSCD): OSCD is XML structured data-centric-document that derived from previous global SCD based on RBA and GSP. It contains only information about the integrated security features of the objects such as the IDs of integrated objects, type of integration (physical or logical) and the new (global) objects' sensitivity labels (Fig. 5.).

Label-Permit Centric Document (LPCD): The LBSMM generates a Label-Permit (certification) to be utilized for upgrading process of the subject sensitivity label. Thus, the LBSMM maps the related label permit information that provided by each local database site to its equivalent XML data-centric-document. Then, the LBSMM integrates the all mapped centric documents based on RBA and GSP in one comprehensive (global) document that is LPCD (Fig. 6.).

Garbage Centric Document (GCD): In the LBSMM, when neither physical nor logical integration are applicable for certain object, the integration is denied and such object will be temporary sent to the Garbage Centric Document (GCD) to be excluded from that particular integration session (performance efficiency), however, after the integration session is completed, all the objects in the GCD will be retained to the SCD.

Rule-Based Algorithm (RBA):RBA consists of a collection of procedures and rules that maintain the security and operability of the LBSMM. The procedures define and control the operation-flow and relationships between the LBSMM components. In addition, the rules also restrain the transitive integration and management streams of the security features at the global level. RBA provides general, integration and management rules.

General Rules (GR)

GR1: Each local database site should provide complete and approved security-information about objects, subjects and subjects' certification. The details of the proposed standard syntax of such information are described in Fig. 2, 3 and 6, in section 3.1. In fact, such rule (GR1) ensures that the relevant semantics of the security features for each individual database are accurately identified.

Integration Rules (IR): The integration rules monitor, control and verify all events done by OSFI in term of sequence and functioning to integrate the objects' security features. However, the implementation of the following rules will be entirely explained in section 3.4 (OSFI):

IR1: Objects Information obtained in GR1 should be properly mapped to the equivalent XML documents (Fig. 2.).

IR2: All mapped XML documents must be accurately integrated in one global SCD (Fig. 4.).

IR3: Objects are required for integration should be precisely determined and the objects' number should be confirmed as well, to decide which integration process type will be carried, i.e., TIOP or CIOP.

IR4: Integration class (Physical or logical) should also be accurately decided by OSFI.

IR5: The new sensitivity labels of the integrated objects should be exactly evaluated, computed and then determined by OSLIP.

IR6: Thus, the objects' security features must be well integrated and then, stored in the global OSCD.

Management Rules (MR): The Management rules monitor, control and verify all events done by ISFM in term of sequence and functioning to manage the integrated security features and handle the users' queries and requests at the global level. Though, ISFM, we will fully describe how the following management rules are implemented.

MR1: Subjects' security information obtained in GR should be properly mapped to the equivalent XML documents, Fig. 3.

MR2: All mapped XML documents must be correctly incorporated in one global LPCD (Fig. 6.) and SCD (Fig. 4.).

MR3: Query type and status (Default or Exceptional) should be well classified.

MR4: Default subject's sensitivity label, object's (s') sensitivity label(s) and all associated tags must be obtained (from SCD and OSCD) and accurately evaluated.

MR5: Thus, the query-management type is decided, i.e., DSM or ESM and then properly handled.

Global Security Policies (GSP): GSP consists of a collection of policies designed to support the RBA to standardize and homogenize the objects' accessibility and its procedures at the global level. On top, GSP for determining authorization as a basis for the access-control decision that made by our LBSMM to achieve the desired security level. Thus, the GSP can also curb the inconsistent integration and management occurrences of the security features in such environment. In addition, GSP offers a universal and flexible (adaptable) platform to implement the LBSMM in various industries such as military and health-care. GSP offers Integration and Access/Management Policies.

Integration Policies (IP)

IP1: Labeling Policies:

IP1_A: Objects with equal sensitivity labels: the same label will be assigned to the integrated (global) object.

IP1_B: Objects with different sensitivity labels: IF the difference = 1 then the higher label will be assigned to the integrated objects. Else (i.e., the difference <> 1) the integration is denied, to maintain the security consistency.

IP2: Object Upgrading Policy: Only one upgrading-level is allowed, when difference between the sensitivity-labels-values = 1. In fact, the upgrading values are: 0 = No level allowed, 1 = one level, or 2 = two levels and the default value is 1. However, IP2 only allows one-level-upgrade to maintain the security.

IP3: Tagging Policy: Each object and subject must be tagged to indicate which individual database it belongs to.

IP4: Denying Policy: If ((the objects' label-values are not equals) and (their difference <> 1)) then the integration of security features is denied.

IP5: Garbage Policy: When neither physical nor logical integration are applicable, such object must be temporary sent to the GCD.

However, how the above policies are applied during the integration process will be entirely explained in section 3.4.

Access Policies (AP):

AP1: Access Policy: Subjects must have Qualified Sensitivity Labels to access the exact objects, i.e., subject's label > = object's label.

AP2: ESM forbids all subjects' queries for Exception-handling when subject_label = 1 or 4.

AP3: ESM prohibits all subjects' queries for Exception-handling when ((objects_labels > subject_label) AND (object_tag = subject_tag)).

AP4: ESM permits subjects' requests for Exception-handling to be proceeded only when ((subject_label = 2 or 3) AND (subject_tag <> object_tag)).

AP5: Subject Upgrading Policy: Only subject with labels = 2 or 3 is allowed to be upgraded to 3 and 4 respectively. This to specify the minimum subject's sensitivity-label allowed for upgrading at a particular destination site. So, values can be 1,2, or 3. In fact, AP5 only allows 2 or 3, because subject's label = 1 is not allowed to maintain the security and 4 is non-upgradeable and to maintain subject's sensitivity clearance.

In section 3.5 (ISFM), we will fully describe how the above policies are applied during the post-integration security management, i.e., throughout the query-handling process.

Objects' Security Features Integration unit (OSFI): This section explains our security integration methodology and its various steps to achieve the security feature integration. In brief, the integration methodology involves the following steps:

- Designing, determining and recording the relevant required information about the security features for each individual database, then, the gathered information will be mapped to the XML repositories.
- Integrating the security features based on RBA and GSP. The integration process involves comparing, analyzing and evaluating of the syntactical (structural) elements and semantic particulars of the security features based on RBA and GSP. Thus, the integration information will be determined such as: the integration type (Physical or logical), the number of the integrated objects and their IDs, the new sensitivity-labels at the global level and other relevant information.
- Finally, the integration will be performed after resolving the security features conflicts based on RBA and GSP.

Therefore, the main function of OSFI unit is to integrate the objects' security features. Actually, It determines all the information needed for the integration

process and then generates the global OSCD. The LBSMM offers two types of integration (Physical and Logical) to ensure coherent integration. Physical based on the objects' syntax, names and types, when the logical based on the objects' semantic.

Physical and Logical Integration of the Objects' Security Features (P/L-IOSF): In physical integration, PIOSF compares and evaluates the object's name and type with the same attributes of the other objects found in the global SCD. If at least one of the two attributes is matched, then it refers to object description attribute to authenticate the semantic of the objects. After the semantic authentication is confirmed, the integration process is established. In the physical integration, There are two levels: identical physical integration that when the two above-mentioned attributes are matched and one-degree physical integration with one attribute matching. When the physical integration is not applicable. The LBSMM offers an alternative technique to be implemented that is logical integration. Logical integration mainly based on objects' descriptive attributes. LIOSF evaluates the semantic among the objects with reference to the objects' remark-text-values, which are located in the SCD. Once the matching process is approved, the integration process can be executed. Otherwise, the integration process is denied and the particular object will be sent temporary to the GCD.

Objects' Security Labels Integration Processor (OSLIP): As mentioned earlier, the most vital part in the integration process is the resolution of conflict between different labels at the global level. Thus, we classify the integration process based on the number of objects that are prepared for integration into two procedures: Two-Objects Integration Process (TOIP) that when only two objects are required to integrate and Cluster-Objects Integration Process (COIP) with group of objects, i.e., more than two objects, as an aid to facilitate the resolution conflict.

For the two selected objects, TOIP acquires the associated sensitivity-label values from the global SCD. Then, TOIP evaluates the two values based on RBA and GSP. Thus, if they are equal, the same value will be assigned to the new (integrated) objects' label, which is located in the global OSCD. If not, TIOP computes the difference between the two values, i.e., the subtraction-value and then applies the absolute rule to that value. If the end-result-value equals to 1, TIOP picks the higher label-value between the two values and then applies the upgrading policy. Subsequently, the higher chosen label-value will be assigned to the integrated objects' label. Then, tagging policy is applied. Otherwise, the integration of the two objects is denied. In addition, to maintain the security of the objects at the global level and to ensure optimal-secure integrated environment, our LBSMM forbids the integration of the objects with difference is not equal to 1.

For group of objects, COIP obtains the associated objects' security-labels-values from the global SCD. Then, COIP compares the values and if they are equal, the same value will be assigned to the global objects' label, which is located in the global OSCD, If not, COIP arranges the specific objects in a set of n distinct objects, (n indicates number of objects). Then, COIP performs *r-Permutation and Combination* of the set of n distinct objects. When r denotes the element ordering selected from n distinct objects. In our model r-value is constant, $r = 2$. So, the *2-combination* of a set of n distinct objects is:

$$C(n, r) = P(n, r) / r!$$
$$P(n, r) = n! / (n - r)! \text{ thus } C(n, r) = n! / (n - r)! * r!$$
$$\text{since } r = 2 \text{ thus } C(n, 2) = n! / (n-2)! * 2!$$

When the subsets of the *C (n, 2)* are built, COIP computes the difference, i.e., the subtraction value, between the two values of each subset and then applies absolute rule for that value. Then, COIP stores the end-result-value of each subset in an array and the index (subscript) of the array denotes the two objects that construct the subset. COIP examines the array and all subsets with cell-value = 0 or 1 will be integrated. COIP chooses the highest value among the selected subsets to be assigned to the integrated objects' label at the global level in the OSCD. Then the tagging policy is applied. Otherwise, when the cell-value is not equal to 0 or 1, COIP divert the particular subset to be handled by the TOIP. Eventually, the ultimate output of this processor will be sorted in the global OSCD.

Integrated Security Features Management (ISFM): This section explains our security management methodology and its various steps to achieve the post-integration security management. In short, our security management methodology involves: collecting, verifying and evaluating of the subjects' queries, access requests and their associated information and then, determining the type of the security management (DSM or ESM). So, post-integration security management will be handled and implemented. The main function of this unit (ISFM) is to manage the integrated security features at the global level. In fact, it handles the users'

queries and requests to control the objects' access based on RBA and GSP.

As mentioned earlier, our LBSMM offers two types of security management (DSM and ESM) to provide consistent label-based access control management at the global level. In DSM, the default treatment presents the subjects' fixed-security labels obtained during the integration process. In fact, the default security labels are established when the objects are integrated and are coupled with them. However, these fixed-security labels can be revised and updated periodically and when required. DSM-labels apply to all received queries excluding those with exception-handling requests. DSM only allows access to objects if a subject has a qualified (authorized) sensitivity label based on the RBA and GSP. If not, DSM either forwards the query to be handled by the ESM or denies the access. In ESM, the exception treatment presents the temporary security label given to a subject for particular session or special-query handling upon a request and its approval. The ESM-labels can temporary override the DSM-labels for a particular session to ensure comprehensive accessibility (Access Rule Consistency) and flexible security management at the global level. In fact, ESM treatment recovers the inconsistency of the integrated security features such as the authorized users who are lost the access rights to objects due to the conflict resolution of the security features after the integration. Thus, the tow specified types of security management are needed to fulfill the integrated environment requirements and provide secured and homogeneous access.

Default Security Management (DSM): DSM automatically generates a Query-Request (QR) for each subject's query or object's access request. The associated QR consists of Subject Identification, Subject Default Sensitivity Label and Query-request's Status (Default or Exception). In addition, the DSM determines the required objects and their sensitivity labels. Thus, if the received QR is valid, the associated query will be moved forward. In fact, DSM only allows access to objects if a subject has a qualified (authorized) sensitivity label that fulfilled the default-access rules and policies, which affirmed in the RBA and GSP.

Exception-handling Security Management (ESM): As indicated earlier, the most crucial part in the integration process is the resolution of conflict between different labels at the global level. Consequently, the resolved solution can lead to inconsistency of the security features, where subjects can lose access rights to objects after the integration. This situation requires a mechanism that can handle and manage the integrated labels at the global level. Thus, ESM treatment recovers the inconsistency of the integrated security features such as the authorized users who are lost the access rights to objects due to the conflict resolution of the security features after the integration.

In fact, in ESM, the subject has Non-qualified (illegitimate) sensitivity-label to access the exact objects. Thus, initially, ESM evaluates the subject's tag and the objects' tags from the SCD and the OSCD respectively. If they are marked with the same tag, i.e., the subject and the required objects formerly from the same local individual database, immediately the ESM denies such access based on the access rules and policies that stated in the RBA and GSP. If not, ESM requests the subject to provide a Label-Permit (LP) for that exceptional query.

Therefore, ESM automatically generates a Label-Request (LR) for that query. The associated LR consists of the Subject Identification, Object Identification(s) that required for access and the Request Time Stamp, i.e., time this request was issued. The subject submits the LR to LPCD-Admin requesting for a Label Permit (LP). Then, LPCD-Admin refers to the SCD, OSCD and LPCD, which placed in the XML repository, to validate and confirm the credentials-ability of upgrading the subject-sensitivity label based on the rules and policies that declared in the RBA and GSP. Thus, if the upgrading request is approved, LPCD_Admin issues the Label Permit to the particular subject. The LP consists of two parts: Confidential part and Subject part. The confidential part includes: Subject Identification, Upgraded (new) Sensitivity Label, Object Identification(s) allowable for access, LP's Time Interval i.e., the LP's lifetime and this to prevent replay after LP has expired. The Subject Part consists of the Confidential-part plus the Subject Identification, Subject Default Sensitivity Label and the Request Status (Approved or Rejected). Then, LPCD_Admin sends the complete LP to the exact subject. Later, the subject presents the received LP to the ESM to access the required objects. Then, ESM verifies the submitted LP and if it is valid, such access is permitted. Otherwise, the access is denied by ESM.

RESULT AND DISCUSSION

There are numerous anticipated and imperative considerations (e.g. risks and mitigations) that need to be carefully addressed and handled when supplying and implementing an ultimate solution such as the LBSMM

for the security interoperability problem in such multifaceted and complex heterogonous environment rather when integrating and managing the varied and assorted security features of such legacy databases:

The continuous trace-ability of a security feature to the corresponding data is very important. Regardless of whether such security integration is performed in conjunction with the database integration or after the database integration is completed, we need a two-way relationship between security features and databases. This is because both databases and security features may be changed after the integration is achieved. To effectively manage changes, we need to easily determine which security features are affected by the changing data and make appropriate changes to the corresponding security features and vice versa. Therefore, the proposed model (LBSMM) can provide the preeminent accurate security interoperability at the global level. In fact, in addition to the systematic method that used to model (i.e., analyze, design, implement, evaluate, etc), supply and specify the proposed model (LBSMM) and all its linked components, security policies, rules, procedures and so forth, the using of the XML with its above-mentioned preferred capabilities are playing principal roles to support, facilitate and assure such continuous trace-ability consideration.

The accomplishment of optimal-inclusive integration and conflict resolution of the designated heterogeneous security features is a tedious and repercussion-prone process rather vital and challenging objective in such integrated surroundings. Thus, in LBSMM, we propose and offer two complementary integration techniques (Physical and Logical) that ensure optimal-extensive and practical security features integration at the global level. Besides, providing two balancing methods (Two-Objects and Cluster- Objects) that examine and classify the objects involved in each integration process as an aid to facilitate the resolution conflicts. On the whole, accomplish and provide ultimate mechanism that can accurately resolve the sensitivity-labels conflicts that arise during the integration process and at the same time, cautiously identify and handle the implications and inconsistencies that result from each resolution to attain the desired pragmatic security integration.

With regards to the post-integration security management at the global level, separation of the exceptional (dynamic) treatments from the default (static) treatments is incredibly significant. The global security policies, rules, procedures, labels, permissions needed to handle and perform a task are well defined and relatively stable and continual during the execution of a system. However, the implications of the conflicts resolution can lead to more vulnerabilities and inconsistency of the security features at the global level, where the exceptional treatments can overcome and take place. Thus, such sort of separation, i.e., dynamic treatments from static treatments, helps encapsulate static handlings in the system (as Default Security Management "DSM" procedure), leaving dynamic handlings flexible for administrators to change and control (as Exception-handling Security Management "ESM" procedure). In this way, the LBSMM provides the best practical design, implementation and enforcement support for post-integration security management in such milieu.

The solidarity and compliance of LBSMM policies and rules with the realistic security and privacy policies of the high-level (global), as well as the autonomy-level are extremely crucial. During the modeling and implementation stages of the LBSMM, we address this concern first by associate each procedure or operation with the policy or rule that govern its use and execution. Then, we link and handle each component, unit, or process based on the designated policy or rule that positioned in the GSP and RBA to restrain its function and implementation for achieving the desired level of security and operability. Furthermore, we standardize and homogenize the linkage definitions and associations between the GSP and RBA (e.g. from each policy at least on rule is derived from and vice versa), which consequentially cascaded and effected on the reminder modules and procedures. Moreover, to address this concern, we extend the security-control management portion by adapting PKCS to offer and handle additional fundamental security aspects. Altogether, reflect and point up evidently the LBSMM solidarity and compliance for such concern.

Other considerations such as simplicity, flexibility, scalability and adaptability, modularity, reliability, etc are very essential for such framework (LBSMM), which can be addressed and underlined through two vital attainments:

- The methodical and systematic approaches that used to design, model, specify and implement the LBSMM components, procedures, policies, rules, etc.
- Using of preferred, supreme and privileged tools such as XML and Java in such multifaceted surroundings that utterly comply with such concerns.

VALIDATION AND IMPLEMENTATION

We validated our expedient model (LBSMM) as follows:

- A healthcare pilot-study has been conducted in earlier stages to confirm and test the feasibility and soundness of the proposed design.
- A Comparative evaluation of the related work has been carried out, which reflected the strengths and advantages of the proposed model and
- We used principal benchmarks such as minimality, expressivity and other modeling criteria to examine and evaluate the LBSMM for conformance with the desired level of security, consistency, interoperability, modularity, simplicity, flexibility, etc, in such surroundings with high confidence, as well as the original revisited objectives. The XML-Java-Based system prototype of the LBSMM has been implemented and tested to check and validate the correctness and accurateness of the system's operations, functions and actual outputs. Furthermore, we introduced and presented XML-EBNF-Grammars that specify, represent and describe the LBSMM components and elements (e.g., subjects, objects, labels, security policies and rules) in XML syntax, in order to standardize and validate all the XML documents, sheets and files that create and compose the model elements, policies, rules and procedures.

CONCLUSION

In this study, we have presented a universal structure of security management model for integrating label-based access control security features of heterogeneous legacy relational databases using XML. The Model handles the integration and management of the security features at the global level to ensure consistent and secure access. The model is composed of five major components namely Objects' Security Features Integration unit (OSFI), Integrated Security Features Management unit (ISFM), Rule-Based Algorithm (RBA), Global Security Policies (GSP) and XML Repository. We have proposed two types of integration process namely physical and logical to ensure optimal-comprehensive integration. We have also categorized such integration process based on the number of objects that are prepared for integration into two (TOIP and COIP) as strengthening to assist the resolution conflict. In addition, we have proposed two sorts of security management: Default and Exception-handling Security Managements (DSM and ESM) to fulfill such environment requirements. Furthermore, we plan to extend the LBSMM in other research to handle the security management of the Web services document as demand increase for Internet substance. On the other hand, we plan also to expand the LBSMM to cover other security aspects of integrated databases such as data integrity, confidentiality and authentication.

REFERENCES

1. Jiang, Y., C. Lin and Z. Tan, 2004. Security analysis of mandatory access control model, Proceedings. The IEEE International Conference on Systems, Man and Cybernetics. Hague, Netherlands, pp: 5013-5018.
2. Foley, S., L. Gong and X. Qian, 1996. A Security Model of Dynamic Labeling Providing a Tiered Approach to Verification, Proceedings. The Symposium on Security and Privacy, Oakland, CA, IEEE Computer Society Press. pp: 142-153.
3. Bertino, E., S. Jajodia and P. Samarat, 1996. Supporting multiple access control policies in database systems proceedings. The 1996 IEEE Symposium on Security and Privacy, IEEE, pp: 94-107.
4. Bidan, C. and V. Issarny, 1998. Dealing with multi-policy security in large open distributed systems, proceedings. The 5th European Symposium on Research in Computer Security (ESORICS 98), pp: 51-66.
5. Zheng, L. and A. Myers, 2004. Dynamic security labels and noninterference, Technical Report, Computer Science Department, Cornell University.
6. Beres, Y. and C.I. Dalton, 2003. Dynamic label binding at run-time, proceedings. New Security Paradigms Work-shop 2003 Ascona Switzerland, ACM, pp: 39-46.
7. Bertino, E., F. Buccafurri, E. Ferrari and R. Rullo, 1999. A logical framework for reasoning on data access cont-rol policies, Proceedings. 12th IEEE Computer Securi-ty Foundations Workshop, AS, CA, pp: 175-189.
8. Bertino, E. and B. Catania, 2001. Integrating XML and databases, IEEE Internet Computing J., 5: 85-88.
9. Bhatti, R., E. Bertino, A. Ghafoor and J. Joshi, 2004. XML-Based Specification for Web Services Document Secu-rity, IEEE Comput. Soc. J., pp: 41-49.
10. Dawson, S., S. Qian and P. Samarati, 1998. Secure interoperation of heterogeneous systems, Proceedings. IFIP-14th International Conference of Information Security, Budapest, Vienna.
11. Gardarin, G., A. Mensch, T. Tuyet and L. Smit, 2001. Integrating heterogeneous data sources with XML and Xquery, Research Report, e-XMLMedia, Bourg La Reine, France.
12. Woo, T. and S. Lam, 1998. Designing a distributed authorization service, Proceedings. IEEE Info-com.
13. Dawson, S., S. Qian and P. Samarati, 1999. Providing Security and Interoperation of Heterogeneous Systems, Research Report, Kluwer Academic Publishers, Boston, pp: 1-29. 1999.

14. Agrawal, R., A. Evfimievski and R. Srikant, 2003. Information sharing across private databases, Proceedings. ACM SIGMOD, San Diego, CA,.
14. Foley, S., 1997. Supporting secure canonical upgrade pol-icies in multilevel secure object stores, Proceedings. The 13th Annual IEEE Computer Security Applications Conference, IEEE, pp: 69-80.
15: Chandramouli, R., 2000. Business process driven framework for defining an access control service based on roles and rules, Research Report, Computer Security Division, ITL NIST, Gaithersburg, MD 20899.
16. Fillingham, D., 1998. Exploration of the use of Partition RBAC for Medical Applications, Research Report, U.S. Department of Defence.
17. He, Q., 2003. Privacy Enforcement with an Extended RBAC Model, Research Report, North Carolina State University, USA.
18. Jajodia, S., P. Samarati and V.S. Subrahmanian, 1997. A logical language for expressing authorizations, Proceedings. IEEE Symposium on Security and Privacy, CA, pp: 31-42.
19. Cholvy, L. and F. Cuppens, 1997. Analyzing consistency of security policies, Proceedings. IEEE Symposium on Security and Privacy, CA, pp: 103-112.
20. Jajodia, S., P. Samarati, V.S. Subrahmanian and E. Bertino, 1997. A unified framework for enforcing multiple access control policies, Proceedings. ACM International SIGMOD Conference, pp: 474-485.
21. Jajodia, S., P. Samarati, L. Sapino and V. Subrahmanian, 2001. Flexible support for multiple access control policies, ACM Transactions on Database Sys. J., 26: 214-260.
22. Ryutov, T. and C. Neuman, 2000. Representation and evaluation of security policies for distributed system services, Proceedings. IEEE DARPA Information Survivability Conference, CA.
23. Lunt, T., 1989. Access Control Policies for Database Systems, C. Landwehr, Editor, Database Security 11: Status and Prospects, North-Holland, pp: 41-52.
24. Rabitti, F., E. Bertino, V.V. Kim and D. Woelk, 1991. A Model of Authorization for Next-Generation Database Systems, ACM-TODS, 16: 89-131.
25. Shen, H. and P. Dewan, 1992. Access Control for Collaborative Environments, Proceedings. The ACM International Conference on Computer Supported Cooperative Work, pp: 51-58.
25. Wang, L., D. Wijesekera and S. Jajodia, 2003. Towards Secure XML Federations, Research Report, Center of Secure Information Systems, George Mason University, USA.