Research

Multilevel security for information retrieval systems – II

Bhavani Thuraisingham

The MITRE Corporation, Bedford, MA 01730, USA

Abstract

In this paper we describe multilevel security issues for hypermedia-based information retrieval database management systems. We first summarize the discussion in our earlier paper on the use of object-oriented approach for developing secure information retrieval systems and then discuss in detail how such a system can be augmented by a Linker to provide hypermedia database management system capabilities. In particular, we discuss various types of nodes and links, an architecture for a hypermedia-based information retrieval system, the various functions of such a system including information retrieval, browsing and update operations, and the interface to the underlying multilevel secure information retrieval system. Finally, the concepts are illustrated using an example.

Keywords: Information retrieval database system; Multilevel security; Object-oriented database system; Hypermedia system; Node; Link; Information retrieval; Browsing; Update operation; Multimedia document

1. Introduction

Due to the explosion in demand for information, retrieval techniques have become indispensable to the efficient usage of computerized library facilities. These facilities include the traditional information retrieval systems [6] and the more recent hypermedia systems [1]. However, easy access to information has also provided an avenue for malicious users to abuse these facilities; i.e. through breaches in security.

In our previous paper in Information and Management, we investigated multilevel security issues for information retrieval database systems. We first described the essential features of information retrieval systems and then discussed the multilevel security impact on these systems. The security issues were grouped into two categories. The first consisted of issues on multilevel data representation and the second consisted of issues on multilevel data manipulation. We proposed an object-oriented approach for representing and manipulating the information. The resulting sys-
tem was called a Trusted Information Retrieval Database Management System (TIR-DBMS).

For many new generation applications, it is important that the system not only provides the facility to support multimedia data such as text, images, voice, and video, it is also important to be able to link the various types of data efficiently so that users have access to large amounts of related but, in general, unstructured information within a short space of time, either by browsing or querying the system. In addition, users should also periodically update the files containing multimedia data so that information contained in them accurately reflect the real-world. In order for a TIR-DBMS to provide this facility, it must be augmented by a module called a Linker as shown in Figure 1. The resulting system is called a Trusted Hypermedia Information Retrieval Database Management System (THIR-DBMS). Figure 2 illustrates an example document stored in the information database; this can be accessed by users at different security levels. The darkened
structures and text represent Secret information while the plain structures and text represent Unclassified information.

This paper describes our research on augmenting the TIR-DBMS described in [5] with a Linker to produce a THIR-DBMS. The TIR-DBMS is summarized in section 2. The major contribution of this paper, which is security for linking the conceptual nodes, is given in section 3. For a background on hypermedia systems we refer to [3]. For a discussion on trusted database management system concepts we refer to [2].

2. An overview of TIR-DBMS

In this design of a TIR-DBMS, the operating system provides all mandatory access control. Also, multilevel documents are stored as multiple groupings in single level fragments. Mandatory access to the single-level fragments is therefore controlled by the operating system.

A system architecture for a TIR-DBMS based on an object-oriented data model is shown in Figure 3. The Linker's requests are mediated by the Linker Interface Manager (LIM). The LIM is responsible for parsing the requests and generating an internal representation. It interfaces to the Object Manager (OM), which consists of five major modules: the Schema Manager (SM), the Browser, the Query Manager (QM), the Transaction Manager (TM), and the Presentation Manager (PM). The SM is responsible for manipulating the object-oriented representation of documents. The user’s view of the multilevel database is object-oriented. The storage and manipulation of the structure of the documents is the responsibility of SM. The Browser is invoked when users ask to scan the various documents. The QM is responsible for query processing. That is, the user’s queries on objects are processed by the QM and translated into an appropriate language so that the Storage Manager (STM) can execute the query. Similarly, the TM is responsible for managing the transactions on objects. The PM is responsible for ordering the documents so that they can be presented in an appropriate format to the user. Note that the Browser, QM, TM, and PM all access the schema and structure information via the SM. The STM is responsible for storing and manipulating the data. It is the responsibility of the STM to decompose multilevel objects into single levels objects so that they can be stored in single level files or segments.

We assumed a security policy where a subject reads an object if the subject’s security level dominates the security level of the object and it writes into an object if the subject’s level is that of the object.

3. Design of a THIR-DBMS

3.1. Overview

As already stated, a THIR-DBMS consists of a TIR-DBMS augmented by a Linker. Here we consider the design of the Linker. It is responsible for connecting the various nodes containing multimedia data. The nodes are conceptual data objects. The Linker’s view of the database is a collections of nodes connected via various types of links. As far as the Linker is concerned, it does not matter whether the nodes contain voice, text, video, images, or graphics. That is, the Linker’s functions are independent of the type of data contained in the nodes. Either one or more nodes are stored in an object managed by the TIR-DBMS or one node is stored in one or more objects managed by the TIR-DBMS. If there is no one-to-one correspondence between the Linker’s view of the system and the TIR-DBMS’s view, then a mapping must be provided to translate the requests on the conceptual nodes into requests on the objects managed by the TIR-DBMS.

---

1 A Trusted Database Management System must ensure that users cleared at different security levels access and share a database in which the data are assigned different sensitivity levels without compromising security.

2 In [5] we discussed four possible architectures and then described the design of a TIR-DBMS based on the architecture where the operating system provides mandatory access control. This design has been influenced by our earlier work on multilevel security for multimedia systems [4].
There are three types of nodes: 3 basic nodes, organizational nodes, and inferential nodes. The basic nodes are: (i) text nodes for storing textual data, (ii) graphic nodes for storing drawings, (iii) sound nodes for storing sound, (iv) video nodes for storing moving pictures, (v) image nodes for storing images, (vi) mixed-media nodes for storing compositions of nodes such as audio/video data, and (vii) buttoned nodes which execute certain procedures.

Organizational nodes are (i) index text nodes, which contain links pointing to index nodes, (ii) index nodes, which contain links to concepts related to the index term associated with the index node, and (iii) object nodes, which describe the nodes (they can be considered as meta-nodes). Inferential nodes are the rule nodes, which contain rules. Note that inferential nodes are needed for an intelligent information retrieval system. Organizational nodes, such as index nodes, are necessary for fast access on a particular topic.

Links connect different nodes. There are three types of links: basic, organizational, and inferential. The basic links are (i) move to links, which move to a related node, (ii) zoom links, which expand on the current node, (iii) pan links, which return to a higher level 4 (or an enlarged) view, (iv) view links, whose activation depends on the user’s role or ID 5, and (v) button links, which are associated with button nodes to execute procedures.

Organizational links are (i) index links, which move either from an indexed node to concepts or from index text to index nodes and (ii) object links, which are associated with object nodes to describe the various nodes of the system. Inferential links are implication links that are associated with rule nodes for inferencing. Inferential links

---

4 Note that we are not referring to a security level here.
5 That is, view links may be a way to enforce discretionary security.
are necessary only for intelligent information retrieval systems 6.

3.2. Nodes

We assume that in a multilevel environment, each node, which is a conceptual data object, is assigned a security level. The question is whether a node should contain single level data or multilevel data. If a node has single level data, then all the information in that node is classified at the same level as that of the node. If a node has multilevel data, then the information in that node must be classified at or above the level of the node 7. Figure 4 (a) shows an Unclassified node which has single level data. That is, it has data classified at the Unclassified level. Figure 4 (b) shows an Unclassified node which has multilevel data. That is, in addition to the Unclassified data, it has Secret data (shown in bold letters): i.e., when a node has multilevel data, only a Secret user can read both the Unclassified and Secret data. An Unclassified user can only read the Unclassified data 8.

The main difference between single level and multilevel nodes (at the conceptual level) lies in the implementation. For example, a node with multilevel data can always be decomposed into multiple nodes with single level data. However, at least at the conceptual level, a node with multilevel data is more in line with the way users view the system. On the other hand, if nodes have single level data, then the mapping between a node and the corresponding internal data object/objects is less complex 9.

---

6 We do not address security issues for inferential nodes and links. However, these nodes and links may be essential, for example, to detect and or prevent security violations via inference in a multilevel environment.

7 It should be noted that this is entirely a conceptual representation. It will be seen that if a node at level L has data at level L* > L, then only a user at level L* or higher can read the data at level L* when clicking on that node.

8 We will refer to a node containing single-level data as single-level node and a node containing multilevel data as multilevel node.

9 In general by data objects or internal data objects we will mean the objects which store the data.

---

Figures 5 (a), and (b) show how the conceptual node with multilevel data could be stored as objects managed by the storage manager of the TIR-DBMS. In Figure 5 (a), the Unclassified data is stored at the Unclassified level and the
3.3. Links

Whether a link is basic, organizational, or inferential, it is assigned a security level and can be traversed by a user at a level at or above the level of the link. That is, an Unclassified user cannot access a Secret link. Links are bidirectional. A link can be a uni-link or a multi-link. A uni-link is a link that is associated with only one node. Examples include the buttoned links, zoom links, and pan links. Multi-links connect two or more nodes. Examples include move-to links and index links. The head of the link is the point where the link begins. The tail of the link is the point where the link ends. The following security property is enforced for links: The security level of a link must dominate the security level of the node which is associated with its head.

This property implies that a Secret node cannot have an Unclassified link emanating from it. However, an Unclassified link can point to a Secret node. Also, the security levels of the forward link and reverse link need not be the same. For example, there could be an Unclassified link from an Unclassified node to a Secret node. But the reverse link of this Unclassified link must be at least Secret.

Figure 6 illustrates zoom and pan links. Here, the node is Unclassified but the zoom link is Secret. Only Secret and TopSecret users can access this link. In this example, we assume that the
enlarged view is Secret. The pan link (which can be regarded in this case to be the reverse of the zoom link) is Secret. When the user clicks on this pan link, the original view will be returned.

Figure 7 illustrates multi-links. The index text node is multilevel. It has index links pointing to several index nodes. Secret entries have Secret links associated with them. Secret links point to

---

**Fig. 7. Multi-links.**

**Fig. 8. Modules of the linker.**
Secret index nodes. Index links are also associated with the index nodes. These links point to concept nodes which contain information about a particular topic. Move-to links connect different information on the same topic.

3.4. Information retrieval, browsing, and updates

The three major operations of a THIR-DBMS are information retrieval, browsing, and updates. Figure 8 illustrates the components of the Linker. They are the User Interface Manager, The Nodes and Links Manager, and the interface to the TIR-DBMS. The Nodes and Links Manager consists of the Query manager, the Transaction Manager (which is responsible for updates also), the Browser, the Presentation Manager, and the Schema Manager (which manages the information about the nodes and links). As can be seen, there is a mapping between the modules of the Nodes and Links Manager component of the Linker and the modules of the Object Manager component of the TIR-DBMS. Therefore, one could envisage the Linker to be embedded into the TIR-DBMS.

Information retrieval is facilitated by the index text nodes and the index nodes. When a user issues a request to retrieve a document, the Linker will scan the index text nodes using some key words specified in the query. From the index text nodes appropriate index nodes are accessed. From the index nodes, concept nodes (such as text, sound, video, and mixed media nodes) relevant to the query are accessed. If the Linker operates at the user's level, then only the nodes classified at or below the user's level are accessed in processing a query.

When the data contained in a node has to be displayed, the data in the corresponding data object (or objects) must be retrieved. The Linker issues a request to the TIR-DBMS which will then transform the request into the request on the data objects. Further mappings may be necessary so that the requests on the data objects are transformed into requests on the files which contain them. Eventually the corresponding files are retrieved and the contents are displayed. For the most part, the TIR-DBMS could operate at the user's level. However, this would mean that the correctness of the mappings between the requests on conceptual nodes and requests on files cannot be guaranteed. This will not result in a security violation, as the operating system controls access to the files. That is, even if the mapping specifies a file that a user cannot access, the operating system will ensure that the file is not opened. However, this means that the integrity of the system is compromised. The TIR-DBMS will return the data retrieved to the Linker together with any information necessary to combine the various pieces of data into a format that is readable by the user.

The ability to browse is one of the main uses of the Linker. That is, the system should ensure that a user is able to scan all of the multimedia documents classified at or below his or her level. Suppose a user wants to browse the network of nodes to obtain all of the information about a particular mission. He or she would first issue a few key words. The Linker will scan the index text nodes and the index nodes and then display a window (which is the starting point of the browsing operation) to the user. A window is in fact a node in the conceptual graph constructed by the Linker. A user can traverse from one node to another by clicking on a link. The links are shown as buttons in a window. That is, a window displayed to the user could have one or more links associated with it. The user can then proceed by clicking on appropriate links that are visible to him or her. That is, he can only click on links that are classified at a level that is dominated by his or her level. If the node, which is pointed to by a link clicked by a user, is classified at a higher level than that of the user, then the window corresponding to that node is not displayed. The user could get some message such as “No more data available”.

A THIR-DBMS must support multiuser updates. Concurrency control techniques must be

\[\text{Note that if data or label integrity is to be maintained, then trust must be placed on the TIR-DBMS as well as the Linker. That is, the Trusted Subject-based Architecture needs to be considered to handle such requirements.}\]
enforced when multiple users update the objects at the same time. These techniques could be implemented at different levels. For example, if concurrency control is to be enforced at the conceptual level, then the Linker must support it. Although it is desirable to have a model of concurrency control at the conceptual level for efficiency reasons, it is not necessary. For example, even if the requests on a node are conflicting at the conceptual level, they need not be conflicting at a lower (such as internal) level. We illustrate this point with an example. Consider an Unclassified node which contains Unclassified and Secret data. Suppose an Unclassified user requests to update the Unclassified data and a Secret user requests to update the Secret data. The Linker will pass the requests to the TIR-DBMS which will then transform the requests into requests on the data objects. Since the data objects specified in the request are different, the two operations can be carried out independently. If a Secret user requests to retrieve the data contained in the node after the update operation is completed, he or she will obtain both the new Unclassified data and the Secret data contained in the node with multilevel data.

3.5. Revisiting example multimedia document

Figure 9 illustrates the logical representation of the example multilevel multimedia document. It is this representation that is managed by the Linker. The darkened structures and boldface letters represent Secret data and the rest represent Unclassified data. The Linker maintains standard information on the structure of a docu-

Fig. 9. Representing the example multilevel multimedia document.
ment. For example, a document consists of a cover, table-of-contents, introduction, set of sections, conclusion, and references. A document cover consists of a title, authors, organization, sponsor, authors' remarks, and document name. If any of this information is non-standard (i.e., varies from document to document), then an appropriate query is formulated by the Linker in order to retrieve it. The non-standard portions of the document include information on the title, authors, organization, sponsor, authors' remarks, authors' speech, document name, and the contents of the introduction. While browsing the document, if a user needs any nonstandard data, the browser will formulate an appropriate query and pass the request to the query manager. The query manager will then be responsible for processing the query and returning the result to the browser. In processing the query, the logical requests will have to be transformed into requests on internal data objects (and possibly files). These transformations are performed by the TIR-DBMS. Access to the files are controlled by the operating system.

Suppose an Unclassified user and a Secret user request to browse the document D11. The screen displayed to both users will have the information shown in Figure 10 (a). If we assume that

Fig. 10. Windows displayed to users at different levels.
this information is standard, the Linker has direct access to it. The buttons shown in Figure 10 (a) are the links. Suppose both users click on the Cover button. The window displayed is illustrated in Figure 10 (b). Next, suppose both users click on the Sponsor button. The Unclassified user will not get any information. That is, a message such as “Not available” could be displayed to him. The window shown in Figure 10 (c) will be displayed to the Secret user. In order for this window to be displayed, an appropriate request such as “Retrieve sponsor for document D11” will be formulated and passed to the query manager. The query manager will parse the query and perform query optimization and then give the modified logical request to the TIR-DBMS. The TIR-DBMS will then transform the request on logical data objects into internal data objects. If the request transformation process is not trusted, then it is possible for incorrect internal data objects to be specified in the request. As mentioned earlier, one cannot guarantee the integrity of the transformation operation if the process that carries out the transformation is not verified as functioning correctly. However, the system will

---

11 For the Unclassified user, the link points to a Null value whereas for a Secret user the link points to the window of figure 9c.
ensure that only data classified at or below the Secret level is retrieved.

Since the links are, in general, bidirectional, users can return to a previous window. Suppose the Unclassified and Secret users both click on the button authors' remarks. The authors' remarks will then be displayed to the users. The returned window contains only one button called Speech. Suppose both users click on this button. Since the voice data is classified at the Secret level, the Unclassified user will not be able to listen to the speech. A message such as "speech not available" could be displayed. The authors' speech will, however, be retrieved from the internal data objects and played for the Secret user.

Next we consider the update operation. Suppose a TopSecret user wants to insert some video data about the author. Also, suppose this user wishes to insert the video data into a logical node which is pointed to by a link emanating from the node containing "voice data". If we assume that write-at-your-level policy is enforced, the Linker operating at the TopSecret level updates the logical representation of the document. The new representation is illustrated in Figure 11. Note that we represent TopSecret information by even darker structures and boldface italic letters. A request of the form: "insert video data from capture device C to follow the voice data in document D11" is formulated and given to the update processor. The update processor will parse the request and give it to the multimedia manager which is responsible for transforming the logical request into request on files. The security level of the file specified will depend on the storage mechanism used. For example, the voice as well as the video data could now be combined and stored in a TopSecret file. A Secret user retrieves the voice data via a trusted subject. On the other hand, data at security level L could be stored in a file at level L.

4. Summary

The explosion in the quantity of documents that are being produced in almost all enterprises today has resulted in computerizing the library facilities. This has required the development of sophisticated information retrieval systems. However, this also means that there is a greater chance of abuse of the information by untrusted users of the system. Many systems provide little or no form of security. In our previous paper, we first stated the data representation and data manipulation requirements of information retrieval system applications and then discussed the security impact. In particular, we discussed issues on developing a multilevel data model for representing these applications and architectural issues for a TIR-DBMS. We utilized an object-oriented approach for developing a TIR-DBMS.

In this paper we discuss the issues on augmenting the TIR-DBMS with a Linker to produce hypermedia data management capability. The resulting system is called a THIR-DBMS. The Linker enables users to browse, query, and update documents. We utilized a conceptual model for the linker and discussed the security issues for the nodes and the links of the model. We also addressed the relationship of the Linker to the TIR-DBMS. The paper provides the direction for designing information retrieval systems that support the data processing requirements of next generation systems.

Disclaimer

The views and conclusions contained in this paper are those of the author and do not reflect the policies of the MITRE Corporation.

References

Bhavani Thuraisingham is a lead engineer with the MITRE Corporation where she is working in realtime database systems, massive database management, and database security. Her interests also include heterogeneous database integration, multimedia information systems, and object-oriented design and analysis techniques for developing various information systems applications. Her previous work at MITRE includes the design and implementation of a secure distributed database system, database inference controller, and secure multimedia/object-oriented database system. She is also a co-director of MITRE’s Database Specialty Group, leads MITRE’s Database Research Program and serves in the Corporate Technology Area Council in Database Systems. Prior to joining MITRE, she conducted research and development activities at Honeywell Inc. where she her work included the design of the secure database system Lock Data Views, the design of a network operating system for space station applications, and also the application of object-oriented technology for developing next generation process control systems and for integrating heterogeneous data dictionaries. Before that she was at Control Data Corporation where she worked on the product development of CDCNET. She was also an adjunct professor of computer science and a member of the graduate faculty at the University of Minnesota. Her work has been published in over forty journal papers and numerous other conference/workshop papers. She is also an inventor of a U.S. Patent in a data management field. Dr. Thuraisingham gives tutorials in object-oriented database systems, heterogeneous database systems, secure database systems, and realtime database systems, has co-edited a book on secure database systems for North Holland and one on secure object-oriented systems for Springer, serves on editorial boards of the Journal of Computer security and the Computer standards and interface journal, and has served as the program chair or as program committee member for several conferences. Dr. Thuraisingham received the M.S. degree in Computer Science from the University of Minnesota, the M.Sc degree in Mathematical Logic from the University of Bristol, U.K., and the Ph.D. degree in Computability Theory from the University of Wales, Swansea, U.K. She is a member of the IEEE Computer Society, ACM, and the British Computer Society.