MULTI-LEVEL SECURE ACCESS AND EDITING SYSTEM FOR BUSINESS DOCUMENTS

Ade G. Ola, Virginia State University, aola@vsu.edu
Emmanuel E. Omojokun, Virginia State University, eomojokun@vsu.edu
Xue Bai, Virginia State University, xbai@vsu.edu
Adeyemi A. Adekoya, Virginia State University, aadeyemi@vsu.edu

ABSTRACT

As businesses continue to expand their operations across national boundaries, there is an increasing need for collaborative work between employees and business partners. However, if company documents and personal or identifying information of customers and employees are to be protected from unauthorized disclosure, access to stored documents must be controlled. This paper describes an Internet-based collaborative document sharing and editing system with security-leveled access control to stored documents. A fundamental goal of the system is to allow documents whose contents have varying sensitivity levels to be viewed and edited by authorized users with varying authority levels, while enforcing the required security constraints. The system allows documents in Office Open XML standard format to be segmented and stored in an embedded database management system, which controls access to documents. Once access is granted, users edit document sections using the corresponding familiar commercial off-the-shelf software.

Keywords: Business Documents, Collaborative Editing, Open Office, Java DB, Java Security

INTRODUCTION

There is an increasing need for employees and business partners of organizations across the globe to collaborate and access the same collection of documents. In such an environment, there is significant risk of unauthorized disclosure of identifying information of customers, employee data, and other pertinent trade documents. This paper describes an Internet-based collaborative document sharing and editing system with security-leveled access control to stored documents. A fundamental goal of the system is to allow documents whose contents have varying sensitivity levels to be viewed and edited by only authorized users with varying authority levels, while enforcing the required security constraints. The system allows documents in Office Open XML (OOXML) standard format to be segmented and stored in an embedded database management system, which controls access to documents. Once access is granted, users edit document sections using the corresponding familiar commercial off-the-shelf (COTS) software. The multi-level secure system will permit businesses, employees and contractors to work seamlessly on the same documents without compromising confidentiality of information.

The solution approach supports sharing and editing of composite documents containing sensitive information in collaborative environments. As designed, the Multi-Level Secure Document System (MSDS) will allow documents to be partitioned into sections with varying sensitivity levels and to be viewed and edited by users with varying authority levels. In order to meet the fundamental goal of allowing documents whose contents have varying sensitivity levels to be viewed and edited by authorized users, the system is designed to meet the following technical objectives:

- Support multi-level documents where a single document may contain multiple sections of varying classification and compartmentalization into subject areas

- Maintain the confidentiality of documents such that users may never view sections of documents for which they do not have clearance or approval

- Allow users to load, edit, and save a document without disturbing sections of the document for which they do not have sufficient authority or approval

- Permit users with appropriate authority or approval to demarcate documents or part of documents into security-level-based sections

- Permit multiple users to work collaboratively by allowing concurrent access to the same documents

- Support the formation of collaboration groups or coalitions

- Enable users to have transparent access to documents that are located on various networks

- Permit documents to be formatted using various commercial off-the-shelf software

SOLUTION APPROACH AND SYSTEM ARCHITECTURE

The solution approach is based on a multi-tiered architecture where documents in OOXML [2] format are managed by Java DB, which is a database management system embedded in the Multi-level Secure Document Java application. An Internet based interface system provides, through a single server, location transparent access to stored documents. Users who are organized into dynamic collaboration groups will have access rights based on their individual security levels, as well as rights inherited from the groups to which they belong. When authorized users gain access to the server, the documents and options displayed in the interface will depend on the security level of the individual requesting access. Whenever the user initiates a document access operation, the application filters the document and produces a constructed copy consisting of the appropriate sections. Users may edit documents in their workspace and save them back to the central storage.

The system has three tiers consisting of Web server, Java application, and Java DB for managing document data. From their web browsers, users connect to the application and subsequently gain access to relevant documents, through the web server. Access to documents is permitted only through the document application, which is invoked from the web browser. Figure 1 depicts the architecture of the system, and the components of the system are described in the following sections.

The Web-base User Interface

Users access the application and documents through a login process. After the authentication process, the Web server presents the document access interface, where the user can browse though a document list or perform a document search. The documents that are available to a user through browsing or a search query depend on the security level of the user. When two users with different security levels access the same document, they may see different paragraphs or sections of the document. The same underlying document may bear different names depending on the viewer; a document presented to the user may also consist of sections from multiple underlying documents.

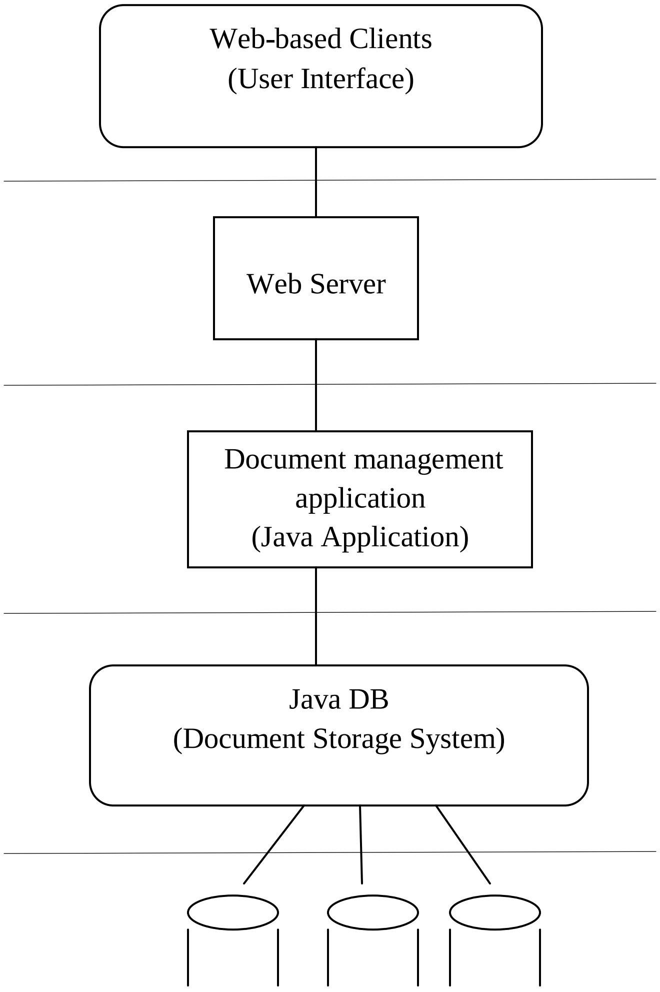


Figure 1: Overview Architecture of MSDS

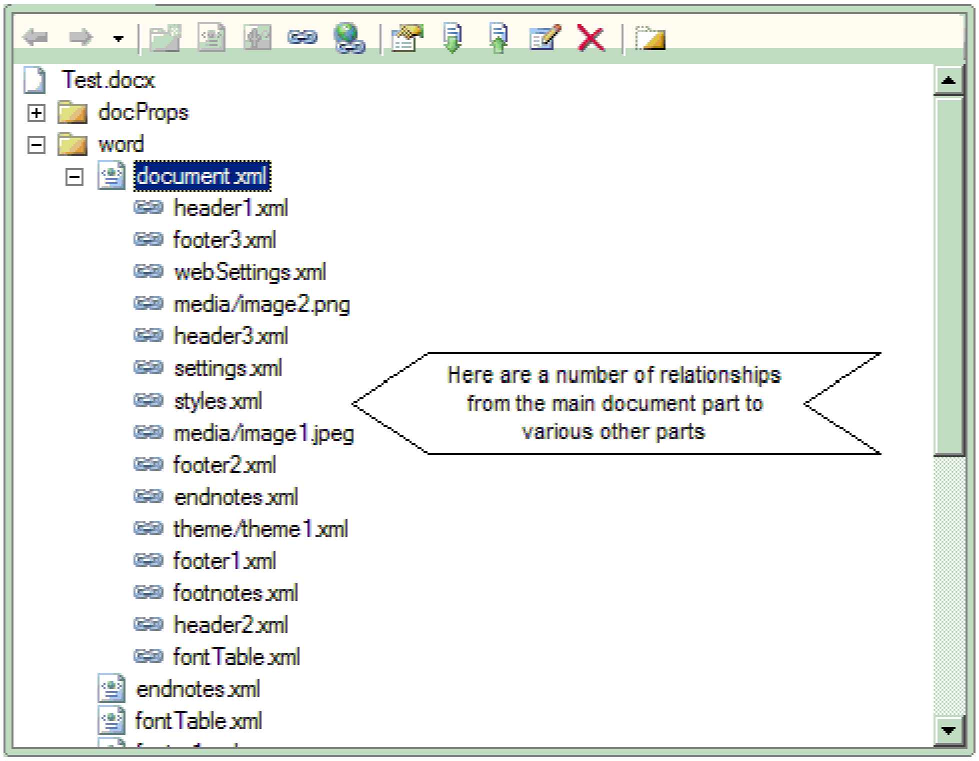
Web Server

The Web server controls access to the application through user authentication at login time, encryption of passwords, and presentation of web pages according to the user's profile. The server processes user requests by forwarding queries or operations to the application and by loading dynamic web pages. The server also runs the application, which allows users with varying roles to perform system and database management functions.

Document Management Application

The application consists of three main components, which loads document from storage, updates and stores document to storage, and allows subdocuments to be defined, respectively:

- The document generator retrieves document sections from storage and composes the sections based on defined document views and the requester's security level
- The document uploading component updates edited sections and paragraphs of the document as defined by the document views available to the user
- The View editor allows authorized individuals to define subdocuments and to assign security properties that dictate who may have access to the sections



Here are a number of relationships from the main document part to various other parts

Figure 2: Main Document structure

Document Format

The ISO standard (ISO/IEC 29500-1:2011) [3] defines a set of XML vocabularies for representing word-processing documents, spreadsheets and presentations, based on the Microsoft Office 2008 applications. It specifies requirements for Office Open XML consumers and producers that comply to the strict conformance category. The Open Packaging Conventions specification defines the structure of Open Office documents. Word 2007 document, for instance, consists of three major components:

Part items: Each part item corresponds to one file in the un-zipped package.

Content Type items: Content type items describe what file types are stored in a document part; for example, image/jpeg denotes a JPEG image. This information enables Microsoft Office, and third-party tools, to determine the contents of any part in the package and to process its contents accurately.

Relationship items: Relationship items specify how the collection of document parts come together to form a document.

Figure 2, adapted from [10], shows a variety of parts that make up the main document part node (document.xml). Normally, zip files are used to implement the Open Packaging Conventions. However, we propose to store document paragraphs and other components in a Java DB (a SQL-based database management system). Using Java DB as the storage system allows us to store document parts in database tables, define views to expose only relevant document sections to different category of users, and to use DBMS access mechanisms to control access.

When an authorized user makes a request for a document, the Java application retrieves appropriate document part items and produces a constructed copy consisting of the appropriate sections. Because users may open any forwarded documents with COTS software (in this case Microsoft Office), they have to be presented in a valid Open XML format. Users may edit documents in their workspace and save them back to central storage, but the packaging and unpacking of the documents is transparent to the end user. Contentions arising from simultaneous requests for the same document paragraphs and sections are resolved at the table row level—row-level locking in Java DB is described in [9].

Java DB

At the lowest level, documents or document sections are stored and managed using Java DB. Java DB is based the open-source Apache Derby database. Java DB is lightweight at 2 megabytes and embeddable within desktop Java technology applications. The MSDS application leverages the capabilities of Java DB, which is a powerful database storage system with triggers, stored procedures, and support for SQL, Java DataBase Connectivity (JDBC) software, and Java Platform. The features of Java DB are summarized in [7].

ACCESS CONTROL AND SECURITY FEATURES

Access to documents is initiated from a web browser; users are authenticated and presented with an application interface based on their security levels and collaboration grouping. The level of access is also defined by these roles:

- *System Administrator* – This is the person who configures Java DB's system-wide behavior. Typically, this is a highly privileged user responsible for allocating machine resources, managing the network, configuring security, and actually launching the VM.

- *Database Owner* – This is the person who creates and tends the databases needed by a particular application; the System Administrator can also serve as the database owner

- *User* – This is a person authorized to use an application; that includes end-users, technical support engineers, and developers who maintain the application

Java and Java DB Security

Security and access control is provided by Java DB as well as outside of Java DB. A summary of the defenses against threats provided by Java DB and Java is given in [8]:

.

- *Java Security* – Using a Java Security Manager and policy file, the System Administrator can restrict the permissions granted to user-written code. The System Administrator can also restrict the permissions granted to Java DB itself

- *SSL/TLS* – The System Administrator can require that SSL/TLS be used to encrypt network traffic between Java DB clients and servers, along the way raising an extra authentication hurdle

- *Encryption* - A Database Owner can require that the data for an application be encrypted before being stored on disk. This makes it expensive to steal and corrupt the data

- *Authentication* – Using usernames and passwords, a Database Owner can restrict access to an application's data

- *Coarse-grained Authorization* – A Database Owner can divide an application's users into three groups: those with no privileges, those with read-only privileges, and those with read-write privileges

- *Fine-grained SQL Authorization* – Via SQL GRANT/REVOKE commands, a Database Owner can further restrict access to fine-grained pieces of data and code

Thus, access is controlled at various levels:

- Authentication through user account and password control mechanism

- The operations allowed and documents displayed in the interface depend on the user’s security level

- Users are assigned to collaboration groups where they inherit group rights

- The sections of documents downloaded at user request are dictated by the user’s security level; the remaining sections of the document remain confidential

- The network locations of documents are transparent to the users; access is through the application which maps virtual documents to physical files.

RELATED WORK AND CONCLUSIONS

There has been considerable interest in document databases in which data is stored as JavaScript Object Notation (JSON) [4] documents and access provided most commonly through HTTP. A Microsoft Developer Network magazine article [5] gives an overview of the so-called NoSQL databases, including document databases such as Apache CouchDB. CouchDb [1] maintains each database as a collection of independent documents, and each document maintains its own data and self-contained schema. The work in [6] proposes the concept of "File View" for presenting different sections of files to various users, similar to the way relational database Views can present different views of an underlying database. The work reported in [11] defines the Secure Confidential Document Model (SCDM) in which a document may be chunked into logical blocks and access to documents traced. However, unlike the research and development reported in this paper, the focus of document databases, File View, and SCDM is not on collaborative editing.

This paper describes the architecture of a collaborative document management system with multi-level security. As businesses continue to expand their operations across national boundaries, the need for collaborative editing without exposure of sensitive personal identification information of employees and customers is expected to grow. The system is expected to improve productivity especially in corporations with global presence.

REFERENCES

[1] CouchDB. Apache CouchDB. http://couchdb.apache.org/

[2] Erika Ehrli (2006). Walkthrough: Word 2007 XML Format, Microsoft Corporation, June 2006 [Available online]: http://msdn.microsoft.com/en-us/library/bb266220(v=office.12).aspx

[3] ISO/IEC 29500-1:2011. The International Standards Organization (ISO) and International Electrotechnical Commission (IEC). [Available online, August 2012]: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=59575

[4] JSON. Introducing Jason. http://www.json.org/

[5] Lerman, J. "What the Heck Are Document Databases," MSDN Magazine, November 2011

[6] Liang, Y., Ai, Y., Dong, H. and Li, T., "File View: Secure Model in Intranet," International Conference on Networking and Digital Society (ICNDS' 09), 2009, pages 198 - 201

[7] Oracle Corporation. Java DB. [Available online, August 2012]: http://www.oracle.com/technetwork/java/javadb/overview/index.html

[8] Sun Microsystems (2008). Rick Hillegas, Java DB Security, June 2008. [Available online, August 2012]: http://www.oracle.com/technetwork/java/javadb/documentation/index-jsp-156831.html

[9] Tuning Java DB. [Available online, August 2012]: http://docs.oracle.com/javadb/10.8.2.2/tuning/index.html

[10] White, Eric (2009). Essentials of the Open Packaging Conventions, Microsoft Corporation, September 2009 [Available online, August 2012]: http://msdn.microsoft.com/en-us/library/ee361919(v=office.11).aspx

[11] Zheng, S. and Liu, J. “A Secure Confidential Document Model and Its Application,” Proceedings of the 2010 International Conference on Multimedia Information Networking and Security, pages 516-519