

# A Proposal for Web-based Learning Environment Security

Reinaldo de Oliveira Castro  
E-mail: [reinaldo@dc.ufscar.br](mailto:reinaldo@dc.ufscar.br)

Mauro Biajiz  
E-mail: [mauro@dc.ufscar.br](mailto:mauro@dc.ufscar.br)

*Department of Computer Science Department of Computer Science  
UFSCar - Federal University of São Carlos UFSCar - Federal University of São Carlos  
Caixa Postal 676 Caixa Postal 676  
CEP 13565-905 / São Carlos – SP - Brazil CEP 13565-905 / São Carlos – SP – Brazil  
Phone/Fax: +55 +16 260-8232 Phone/Fax: +55 +16 260-8232*

## Abstract

*Distance Learning has been studied intensively in the past few decades due to the growing need for ongoing education brought about by the increasingly competitive job market. Because it offers a new form of student-teacher interaction, the Web plays a significant role in this scenario. Highly advantageous possibilities arise from this newest form of interaction, among them the absence of distance limitations and the application of audio-visual resources in the teaching/learning process. Moreover, obsolete procedures applied in distance learning can be reformulated based on technological advances, rendering it more effective and reliable. One of these procedures is identity confirmation on remote tests performed by students. The security architecture presented herein provides a reasonable solution for this problem. Furthermore, it creates a flexible access control method for Web-based learning (WL) environments.*

## 1. Introduction

The main purpose of computer security, a crucial issue ever since the first computer was created, is to provide protection against the illegal access of information. Advances in the field of computer science, driven by the high value our society places on information, have led to increasingly complex solutions for this protection [1, 2, 3, 4, 5, 6].

New security requirements have emerged as a result of the Web's growing popularity. Because the natural place for the purposes of WL environments [7, 8] is the Web, these environments must also adhere to these new requirements.

In this paper, we propose a security architecture for WL environments that meets the most up-to-date authentication and authorization requirements, providing an infrastructure that improves the environment's security

and facilitates access control management.

The reminder of this paper is organized as follows. Section 2 describes some basic concepts about authentication and authorization. Section 3 discusses our security architecture for WL environments. Lastly, section 4 presents the conclusions of our work.

## 2. Authentication and authorization

Authentication, which is formally defined as a process by which the entities involved in a communication establish their identities [9], generally involves the use of a username and a password. Banks, for instance, apply this mechanism in their automatic teller machine operations.

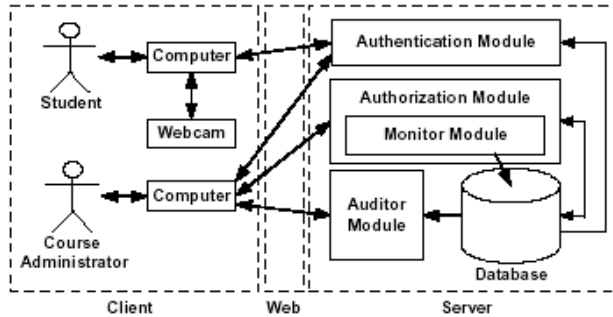
An authorization, which is the unit that controls the actions a user can perform in a computer system [10], keeps user visibility at a level congruent with the environment's rules. Bertino [11] formally defines an authorization as a triple  $\langle s, a, o \rangle$ , where  $s$  is an active entity which requires access to objects,  $a$  is an action type that defines the action (read, write, etc.) requested by that entity, and  $o$  is the object accessed and/or modified.

## 3. An architecture for Web-based learning environment security

The architecture described here behaves in two distinct manners, depending on which user interacts with it: a course administrator or a student.

The course administrator is responsible for the maintenance of a course and grants access rights to all the other users, i.e., students, teachers, etc. A student is a user enrolled in a course. The course administrator's and the student's view of the architecture are shown in Figure 1.

In the figure above, a course administrator and a student on the client-side interact through the Web, using the modules that comprise the architecture, i.e., the server-side. For the course administrator, this interaction



**Figure 1 – Architectural elements.**

may involve an authentication in the authentication module, an authorization insertion, an authorization deletion and an authorization search in the authorization module, or an audit in the auditor module, with the student as the target. The only possibility of interaction for the student is an authentication in the authentication module.

The functionality of each module is discussed in greater detail in the next subsections.

### 3.1. Authentication module

Authentication is usually the first process to which a user is subjected when interacting with a WL environment. In our architecture, this task is carried out in two different ways, i.e., *simple authentication* and *special authentication*.

**3.1.1. Simple authentication.** Every user environment normally uses simple authentication, which consists of two main steps. First, the Web browser requests the user's username and password. This information is posted in the server and its authenticity checked against the database. The user is prompted to reenter the information if the values do not match. Access to the environment is automatically denied for a while if this procedure is repeated three times. Provided he successfully completes the first step, the user then is presented with a random question. This question usually involves personal information about the user, e.g., 'What is your date of birth?'. Access to the WL environment is granted when the user successfully completes this last step. Rerouting to the first step is necessary if the last one fails. Because of its very low implementation cost, simple authentication is the most commonly used authentication mechanism.

**3.1.2. Special authentication.** Testing is a subject that has been a critical issue in all distance education theories. We propose herein a special, sophisticated form of authentication for remote tests, whose main purpose is to do away with the traditional 'face-to-face' testing methods and to reduce the possibility of student cheating

facilitated by the distance between student and teacher.

This special authentication requires some additional hardware and software:

- A web-camera (webcam), which is used to capture the student's image;
- A software program for the recognition of facial features.

When a student wishes to take a remote test, the authentication module checks the webcam's availability. The process stops if the webcam is turned off and continues if it is turned on, in which case the web page containing the test is presented to the student. An image of the student's face is sent to the server every time he interacts with the test page. The server performs the face recognition process using the facial feature recognition software program. If the features do not match the student's facial image stored in the database, the server redirects the student to another page, which informs him that the remote test has been cancelled. If no problem is found in all comparison generated by the interaction with the test web page, the student has his test validated.

### 3.2. Authorization module

The existing WL environments do not work flexibly with access control management, because the authorizations are granted to roles rather than to individual users. Therefore, a problem will occur if the course administrator wishes to grant an access right to an individual user.

Our architecture solves this problem by means of the authorization module. This module allows access rights to be granted to both roles and individual users, based on the concept of explicit/implicit authorization to reduce the management effort involved in the greater number of individual authorizations. An implicit authorization does not exist physically, but is deduced through an explicit authorization.

Because the database stores only explicit authorizations, is necessary for the authorization module to check the explicit authorizations in its three domains (*S*, *A* and *O*) to identify implicit authorizations. The following subsection explains how this works.

**3.3.1. Role, action type and object domains.** The elements of each domain of an authorization fall into a natural hierarchy that permits the monitor module to identify implicit authorizations. An example of these hierarchies is illustrated in Figure 4.

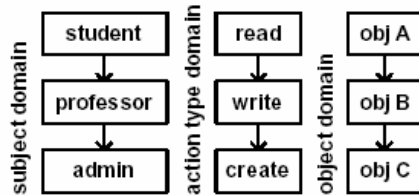


Figure 2 – Domain hierarchies.

In the above figure, a user assuming the role of *professor* implicitly inherits all the access rights of the students. If a user has the right to carry out the *create* actions, he also has the implicit right to perform the *read* and *write* actions. A user who works with *obj B* can also work with *obj A*.

After analyzing the possible combinations in the three domains, the authorization module grants the user access rights if an explicit or implicit authorization is found. Otherwise, the access right is denied. Thus, this module reduces the effort involved in the management of individual authorizations.

### 3.3. Monitor module

Every time a user requests access to the authorization module, the monitor module stores several items of information in the database for later analysis, including the date and time of the request, the time the user spent in a protected object, and the objects accessed previously and subsequently. Thus, the monitor module keeps a ‘log’ of the user’s actions.

### 3.4. Auditor module

In some cases, the course administrator wishes to know, for instance, what actions a user takes to carry out the tasks of a course or the number of times he has accessed a particular object. The auditor module is responsible for performing these tasks. Its two main objectives are to aid evaluations of the user and to create statistical user profiles. The first objective is achieved by an analysis of the above-mentioned set of information, which produces results such as “*Student X was given an evaluation Y because he/she accessed the required material Z times*”. The second is achieved by exhaustive analysis of the database. The resulting statistical information profiles would show data such as “*Student X routinely accesses the math class at about 2:00 p.m., remaining there for about 36 min*”.

## 4. Conclusions

This article presented a WL environment security architecture proposed for use in different WL environments. Its main goals are to provide a reasonable

solution for the problems of student’s remote testing and to create a flexible access control method. In addition, it produces useful information to aid in user evaluations and to create statistical user profiles.

## 5. Acknowledgement

This research has been funded by CAPES.

## 6. References

- [1] Fernandez, E. B., Larrondo-Petrie, M. M., and Gudes, E., “A Method-Based Authorization Model for Object-Oriented Databases”, In Proceedings of the OOPSLA- 93, Washington DC, 1993, pp. 135-150.
- [2] Gal-Oz, N., Gudes, E., and Fernandez, E. B., “A Model of Methods Access Authorization in Object-oriented Databases”, In Proceedings of the 19<sup>a</sup> VLDB Conference, Dublin-Ireland, 1993, pp. 52-61.
- [3] Fernandez, E. B., and Chien, P. D., “Authorization in Multimedia Conferencing Systems”, Communication and Multimedia Security, 1995, pp. 133-147.
- [4] Lampson, B., Abadi, M., Burrows, M., and Wobber, E., “Authentication in Distributed Systems: Theory and Practice”, ACM Transactions on Computer Systems, Vol. 10, Number 4, 1992, pp. 265-310. [5] Fernandez, E. B., Summers, R. C., and Wood, C., *Database Security and Integrity*, Addison-Wesley Publishing Company Inc, 1981.
- [6] Jackson, K. M., and Hruska, J., *Computer Security Reference Book*, CRC Press, 1992.
- [7] Lucena, M., and Salvador, V., “Learn@Web: An Integrated Environment for Cooperative Learning”, V Workshop of Informatics in the School, XIX Symposium of Brazilian Computer Society, Rio de Janeiro-Brazil, 1999, pp. 743-758.
- [8] Lucena, C. J. P., “AulaNet – An Environment for the Development and Maintenance of Courses on the Web”, In Proceedings of the International Conference on Engineering Education, Rio de Janeiro-Brazil, 1998,
- [9] Chin, S. K., “High-Confidence Design for Security”, Communications of the ACM, Vol. 42, Number 37, 1999, pp. 33-37.
- [10] Sandhu, R. S., and Samarati, P., “Access Control: Principles and Practice”, IEEE Communications Magazine, Vol. 32, Number. 9, 1994, pp. 40-60.
- [11] Bertino, E., and Martino, L., *Object-oriented Database Systems*, Addison-Wesley Publishing Company Inc, 1993.