AN ACCESS CONTROL SYSTEM FOR MULTIMEDIA APPLICATIONS
Alexandros Kadoglou and Ioannis Pitas
Department of Informatics, University of Thessaloniki
54006 Thessaloniki, GREECE
Tel.: +30-31-996304, FAX: +30-31-996304
e-mail: pitas@zeus.csd.auth.gr

Abstract - We present a system that can be used for access control for multimedia applications and services. The services that concentrate our interest include multimedia titles, software on the Internet, video on demand, Pay-TV and other existing and forthcoming multimedia services that are are and will be available through the Internet and other digital distribution channels. Based on the theoretical study regarding the access control system, we have built a software platform that provides all the required functionality. The software has been developed so that it can be used as the base for electronic commerce but can also be easily adapted to provide solutions to other problems related to on-line service distribution such as distance learning. The platform is based on the client – server approach. It supports the secure ordering, downloading and execution of services and multimedia applications as well as acting as a remote licence manager for applications located in the user’s computer.

I. INTRODUCTION

This paper presents an implementation of the access control system proposed in the framework of the European research projects ACCOPI (Access Control and Copyright Protection for Images) and OKAPI (Open Kernel for Access to Protected Interoperable interactive services). The projects deal with the problem of access control and copyright protection for broadcasted image related services.

The main objective is the development of a common, equitable and open access control system, adjusted to the European needs and requirements, that will be proposed as a European standard. The services that concentrate our interest include mainly Pay-TV, but also the existing and forthcoming multimedia services which are expected to share the same channels as digital TV for their diffusion in the future. We assume that these services have the basic characteristic of interactivity, which implies the existence of a return channel for the users to the service providers [1].

After the theoretical study regarding the access control system, there was a need for a software-based platform that would illustrate the concepts and the capabilities of the kernel of the system in a concise and simple way. Besides, the development of a software based platform before a real hardware implementation is very helpful, because several implementation problems and possible theoretical gaps can be found and resolved with lower cost. Also, some parts of the platform could be used with little or no modifications in the software of the real implementation.

The platform is based on the client – server approach. Dedicated servers and clients have been developed for this system. The clients are used to connect to the Service Provider’s server and view information on available services, order them and subsequently download / execute them on their personal computers. The server can perform all the actions expected from such a system. These include marketing the available services through the use Web pages and the accessing of demonstrations of its services, managing its service and customer databases, but most importantly restricting access to authorized users only (those who have ordered and paid the service). This access control is extended even to the user’s remote computer, i.e. to computer programs that have been downloaded by the user and are executed locally.

The system is oriented towards two types of services: a) accessing multimedia content available from the Service Providers and viewed with the use of special viewer software and b) use
of computer software for a specified (and restricted) time period and/or location. In the latter case
the computer software is downloaded to the user’s computer but is protected with a locking
mechanism that enables the Service Provider to act as a remote license manager.

Furthermore, the kernel supports features that enhance the security, data integrity and
management of the kernel, especially needed for the Service Provider. These include several
operations that allow the management of the available services, the logging of requests for services
and support for the list of customers that perform transactions with the SPv. All the data required
for the SPv’s operation can be stored in an external database. This was accomplished by using the
ODBC interface that abstracts the underlying database mechanism so that any commercial database
program that is compliant with ODBC can be used for information storage. Operations on the
database are fully supported through the SPv’s GUI interface with the capability to manipulate the
records by using display, add, remove and modify operations.

Finally, a graphical interface for the User module was developed in the Java language. The
functionality of the OKAPI kernel is provided by a library that the interface uses to perform the
necessary operations. The main reason for the adoption of Java in the porting process of the User
part of the OKAPI kernel was its ability to produce executables that can run without changes under
a vast variety of hardware platforms and operating systems as long as they provide a Java Virtual
Machine (JVM). By porting the User part of the kernel to Java, the need for development of
different versions of the program vanishes.

II. DESCRIPTION OF THE COMMON FUNCTIONAL MODEL

The platform is concentrated on the basic operations provided by the system, that make it
flexible and, at the same time, make possible or help the access control and the billing of the
services. In fact, what we have implemented is a part of the Common Functional Model (CFM) [2]
of the access control system, that is the devices and references related with the central functions: the
User, the Service Provider (SPv) and the Trusted Third Party (TTP). The presence of the TTP as a
trusted link between the users and the SPv’s is necessary to ensure the equitability and the security
of the system [3].

The devices related with the user are the Terminal and the Access Control Unit (ACU). The
devices related with the SPv are the Access Information Directory (AID) and the Entitlements1
Collection. The device related with the TTP is the List of ACUs. What we have implemented for the
user is the combination terminal - ACU and the development of the operations provided by the
Operating System of the terminal. What we have implemented for the SPv is a server that responds
to user calls (e.g. for ordering / canceling entitlements) and an interactive operational console to
to control the server. What we have implemented for the TTP is a server that responds to user calls
(for registration of an ACU, blacklisting of an ACU) and SPv calls (for checking the validity of an
ACU, requesting the enciphering key of an ACU, blacklisting of an ACU), and an interactive
operational console to control the server [4].

III. OPERATIONS

In this section we shall describe the operations that are provided by the access control system.
These operations are the ones that we have so far included in the implementation.

---

1 The entitlement is a service that is available for ordering and accessing
A. User operations

- **Display Terminal Settings.** This operation gives the user information on the State, Country, Region, City, Geographical Co-ordinates and Lockout Threshold for the local terminal.
- **Change Lockout Password.** This operation changes the password that allows the modification of the lockout threshold. This threshold is used as a "parental lock" mechanism that prohibits access to entitlements that are rated higher than the current setting of the threshold.
- **Reset Lockout Password.** The lockout password is reset to be the same as the root password of the terminal.
- **Change Lockout Threshold.** The lockout threshold of the terminal is changed accordingly.
- **Create New ACU.** A new Access Control Unit (ACU) is created for the User.
- **Insert/Change ACU.** A new ACU is inserted into the terminal or the existing ACU is replaced by the new one.
- **Registration of a new ACU.** This operation registers the current ACU to the TTP.
- **Add new Service Provider.** An SPv is added to the list of accessible SPvs from the current ACU.
- **Remove Service Provider.** An SPv is removed from the list of accessible SPvs of the current ACU.
- **Display SPv Index of ACU in use.** The list of SPvs accessible by the ACU currently in the terminal is displayed.
- **Display ordered entitlements.** The entitlements that have are already been ordered and are accessible by the current ACU are displayed.
- **View entitlement info.** The user connects to an SPv and receives a list of available entitlements. Information on the ones that interest the user can then be obtained through the SPv’s Web server. Demonstration versions of the entitlements are also available for downloading through this operation.
- **Order Entitlements.** This operation performs the ordering of a new entitlement from an SPv. The user is presented with the list of accessible SPvs from which he chooses one. The selected SPv is contacted and a list of the available entitlement for ordering is received. Subsequently, billing information as well as the type of access required is sent to the SPv (options include unrestricted, password or HASP protected, time limited, node limited or any valid combination of the above). The operation concludes with the specification of the required time access period.
- **Cancel entitlements.** An already ordered entitlement is selected by the user to be cancelled. The SPv from which it was ordered is contacted and the entitlement is subsequently removed from its database as well as from the ACU.
- **Execute entitlement.** The selection of the entitlement to execute is made from the list of already ordered entitlements. The appropriate actions to access the entitlement are taken, such as downloading, installing and launching viewer applications, or launching local applications after the proper authorization has been received from the SPv.
- **Send blacklist request to TTP.** With this operation the user can notify the TTP for a stolen or otherwise compromised ACU.
- **Help.** Provides information on the order of operation that must be followed. Especially useful for first time users of the system.
- **Quit.** This operation stops the user module.
B. **SPv operations**

- **Display SPv's data.** Information for the SPv in question is displayed.
- **Display user's entitlements.** The operation lists all the ACUs that have ordered entitlements from the SPv along with the ISBN of each of the ordered entitlements.
- **Enable user's entitlement.** An entitlement already ordered by a user is enabled for accessing by specifying the method with which the entitlement was paid for.
- **Display available entitlements.** The entitlements that are available for ordering are displayed. The attributes that are displayed are the following: Entitlement id, Starting day, Starting month, Starting year, Ending day, Ending month, Ending year, Duration, Theme number, Level number, Provider id number and ISBN number.
- **Add new entitlement.** This operation adds a new entitlement to the SPv that will be available for ordering.
- **Delete entitlement.** Deletes an entitlement that was available for ordering from the SPv’s records.
- **Modify entitlement.** Modifies the properties of the entitlements that are available for ordering.
- **Display customers.** Displays the customers in the SPv’s records.
- **Add new customer.** Adds a new customer to the SPv’s records.
- **Delete customer.** Deletes an existing customer from the SPv’s records.
- **Modify customer.** Modifies the information about an existing customer in the SPv’s records.
- **Send blacklist request to TTP.** With this operation the SPv can notify the TTP for a stolen or invalid ACU.
- **Change password.** The password required to interface with the SPv server is changed to the one specified by the operator.
- **Stop SPv server.** Stops the SPv server.
- **Exit.** Stops the GUI program.

C. **TTP operations**

- **Display data for registered ACUs.** The list of all the ACUs that are registered is displayed along with information on whether a specific ACU is blacklisted or not.
- **Blacklist ACU.** The given ACU is marked as blacklisted in the TTP's records.
- **Change authentication password.** The required to interface with the TTP server is changed to the one specified by the operator.
- **Stop TTP Server.**

IV. **Key features of the Kernel**

A. **Application execution**

In the development of the kernel, emphasis was given to operations such as multimedia application **downloading, installation and execution.** These operations are crucial because the accessing of services that are available through the kernel may require the use of additional "viewer" applications, which must be retrieved and installed in the user's system. The type of the ordered entitlements determines the appropriate applications.

These applications can be one of two things: They can either be viewers for the proprietary data format of services provided by the SPvs, or they can be the entitlement itself. In any case, a mechanism must exist within the kernel that will allow the retrieval and installation of these
applications to the local computer system. Such a mechanism was indeed incorporated into the kernel.

A downloaded file may be a number of things. In the simplest case it is the executable for the application. However, it may also be an archive file, a self-extracting archive or some other application distribution format. Depending on the type of this file, the kernel takes the appropriate action to correctly install the application, based on predefined procedures. In the case of an archive file that is not supported directly by the kernel, external extracting utilities can be used.

When a user wishes to execute an entitlement and the appropriate executable cannot be found in the local system, the kernel assumes that this is the first time that the entitlement is accessed or that for some reason the application was deleted. In either case the kernel initiates the downloading and installation procedure described earlier. After the accessing of the service is over the application remains in the local system for future use.

B. Restricted application execution

In order to enhance the capabilities of the kernel we have allowed for a number of different access possibilities to the provided services. Apart from the obvious choice of unrestricted access, there is the possibility of a HASP or password protected service, which requires a one-time authorization from the SPv and the possibility of time, and/or location restricted access that requires authorization every time the service is accessed.

HASP and password protected access require some information about the ordering ACU and the terminal that is used. This information is used in the construction of the HASP or of the password and is given to the SPv during the ordering process. In the case of HASP protection, the HASP lock has to be physically sent to the User. In the case of password protection, when the service is first accessed the SPv is contacted and the User receives the appropriate password that unlocks the service.

The capability for time and/or location restricted access to services and applications is very important to this project. There are two possibilities here: First the service is located at the SPv and the user accesses it remotely and second the service is located at the user's computer and therefore local access is used. The first case can be easily handled since the SPv can refuse access to the services’ data if the appropriate entitlements have not been ordered by the requesting ACU.

The second case, however, requires that the SPv essentially acts as a license manager that grants or denies execution access to entitlements already located at the user's computer. The danger here is great since a pirate can try to unlawfully access or duplicate (copy) an entitlement. In order to protect the entitlements in this case, we have incorporated an execution restriction mechanism into the kernel.

This mechanism is implemented as follows: When someone wants to use a service for a specific period of time, he/she must contact the appropriate SPv that holds the entitlement and order it. During the ordering process the start and end dates of the intended usage period of the program are specified by the user. Once the ordering is complete the user can execute the program by choosing the "Execute" operation of the kernel. At this point the User module contacts the SPv and requests permission to run the application. If this is the correct time period based on the SPv's date and time, the permission is granted and the User module launches the application that should be installed in the local computer. If the proper executable cannot be found locally, then the application is downloaded from the SPv, is installed and subsequently executed without any intervention from the user, using the mechanism that has already been described.

As soon as the application program is successfully launched, the User module waits for connection from the application program on a specific TCP/IP port. Upon execution, the first thing
the application program does is to connect to the User module using that port and ask for the proper authorization message. The application will only continue execution if the correct message is received, otherwise it terminates. This method ensures that the entitlement (application) is protected against execution from outside of the kernel. If it is run by itself without the use of the User module, it will never receive the authorization message to execute and after the specified time-out period, it will terminate. Therefore, unauthorized access is prohibited.

Another possibility with this method is node limited access, where execution of an application is allowed only from a specific terminal or computer. In this case, the User module sends information about the terminal or computer during the ordering process. If the ACU that contains the entitlement and the application code are moved to another terminal and execution of the same entitlement is attempted, the terminal information will be different to that stored by the SPv and access will be denied.

C. Security issues

The security issues of the kernel were also addressed with the addition of cryptography capabilities. The exchanged messages can be encrypted using the publicly available PGP package that implements public key cryptography. We have also used another package, the SECUCDE security toolkit that is available for free for non-commercial applications, which provides a friendlier interface and better integration with the rest of the code. The cryptographic capabilities that are required for the kernel operation are provided equally well by both packages so we have developed the code so that either cryptography package can be used.

The encryption can be used during the transmission of control messages between the modules as well as for the secure downloading of applications, thus protecting the services themselves.

The use of encryption was dictated by the need for security during the sensitive transactions between the User and the SPv modules. Such transactions include ordering entitlements, sending billing information as well as downloading entitlements. We have chosen to use public key cryptography where a pair of two keys are required: a public key that is used to encipher a message and the respective secret key, known to the recipient of the message only, which is used to decipher the message. The obvious advantage of this method is that the public key can be distributed freely over an insecure channel.

In order to ensure the secure transmission of messages a number of different keys must be created and maintained by the various actors. The different keys and their usage are briefly described below.

• Each ACU is assigned a pair of public and secret keys that are only known to the TTP and the ACU itself. The keys are created by the User program during ACU creation and the public key is subsequently transmitted to the TTP.
• Each SPv is assigned a pair of public and secret keys that are only known to the TTP and the SPv itself.
• Each ACU is also assigned a unique pair of public and secret Programmer Distribution Keys (PDK) from each SPv that it can access. These are created by the SPv when a new ACU contacts the SPv for the first time. The PDK is securely transmitted to the ACU using the ACU’s public key that the SPv retrieves from the TTP. The transmission of the ACU’s public key from the TTP to the SPv is encrypted using the SPv's public key.
• The PDK is used for the encrypted transmission of the Service Key (SK) from the SPv to the ACU which is unique to the SPv but can be the same for different ACUs.
• The SK is used for the encrypted exchange of messages between the SPv and ACU as well as the encrypted transmission of services.

D. Marketing of services

We have also introduced an operation that allows the marketing of the entitlements available for ordering from the different Service Provider. This way the marketing and promotion of entitlements is incorporated into the kernel, thus making it more complete and more appealing for commercial applications.

The promotion of the entitlements is accomplished through the use of WWW pages that the SPv maintains on its own web server. Each of the available entitlements for ordering is described and promoted on a separate page. The maintenance of these pages is outside the scope of the kernel.

When this operation is selected, the User module connects to the SPv, receives the list of the available entitlements for ordering and presents it to the user. At this point the User module can launch an external Web browser that is directed automatically to the WWW page that contains the promotional information for the entitlement that interests the potential buyer of the entitlement. The location of the page is retrieved from the SPv at that moment.

The Service Provider also has the ability to promote its entitlements by providing demonstration versions that are available to the user without having to previously order the entitlement. The downloading and execution of these demonstrations requires the same data structures and the use the same mechanism as the complete version of the entitlements.

E. Kernel portability

The main reason for the adoption of Java in the porting process of the User part of the OKAPI kernel was it’s ability to produce executables that can run without changes under a vast variety of hardware platforms and operating systems. By porting the User part of the kernel to Java, the need for development of different versions of the program vanishes.

To be more specific, the GUI part of the program which is the one that poses the most difficulties to be rewritten for different operating systems was developed in Java. The other functions that had to do with communication and processing remained coded in C++, as it is easy to port them to another platform with some slight changes and a recompilation. By linking the Java and C++ parts of the programs emerges a highly portable User version. By using this version, a User can connect to a TTP or SPv server using almost the same binaries on every machine that has Java support. Users that execute the program will have all of its abilities at their disposal, as if it had been developed in a platform-dependent language.

V. CONCLUSIONS

We have developed a system that can be used for access control for multimedia applications and services. This system is suitable to be used as an electronic commerce platform and supports all the operations that are expected from such a system. These operations include obtaining new digital user IDs, user registration, ordering, cancelling and accessing of services, as well as marketing operations for service promotion using the Web. Also the direct downloading, installation and restricted execution of services is supported. Furthermore, it introduces the notion of the remote licence manager which operates through the access restriction mechanism provided by the platform. All these operations are performed securely under the protection provided by the public key encryption algorithms that are used on the the exchanged messages and data.

The platform is based on the client – server approach with dedicated servers and clients for each participating entity that provide the required functionality. It has proven to be stable and
efficient during the trials it was subjected to and can be used for restricting access to services effectively and reliably. Portability issues have also been addressed by developing the User part in the Java language and can therefore be executed on a vast variety of hardware.

As a result, the system can be used in real situations. Its importance and applicability becomes even greater by the fact that it can be easily adapted to provide solutions to other problems related to on-line service distribution such as distance learning.

ACKNOWLEDGEMENTS

The work reported in this paper was performed for the European ACTS projects ACCOPI and OKAPI.

REFERENCES


