An Implementation of a Trusted and Secure DRM Architecture

Víctor Torres, Jaime Delgado, and Silvia Llorente

Universitat Pompeu Fabra, Passeig de Circumval·lació, 8,
08003 Barcelona, Spain
{victor.torres, jaime.delgado, silvia.llorente}@upf.edu
http://dmag.upf.edu

Abstract. Content providers and distributors need to have secured and trusted systems for the distribution of multimedia content with Digital Rights Management (DRM) to ensure the revenues derived from their works. This paper discusses the security mechanisms applied to the implementation of a DRM architecture, regarding the certification and verification of user tools integrity during their whole life cycle, the mechanisms for providing a secure and trusted communication between client tools and the server framework for authorisation, certification or verification purposes, and the mechanisms for the secure storage and resynchronisation of the reports that describe the actions performed by users during the tool offline operation. The presented architecture is being implemented in the AXMEDIS project, which aims to create an innovative technology framework for the automatic production, protection and distribution of digital cross media contents over a range of different media channels, including PC (on the Internet), PDA, kiosks, mobile phones and i-TV.

Keywords: Secure content management, multimedia content protection, digital rights management systems.

1 Introduction

In [1] [2] [3] we presented in a general way an architecture to manage multimedia information taking into account digital rights management (DRM) and protection. The architecture, called DMAG Multimedia Information Protection and Management System (DMAG-MIPAMS), whose name is after our group acronym DMAG [4], consists of several modules or services, where each of them provides a subset of the whole system functionality needed for managing and protecting multimedia content. The architecture is depicted in Figure 1.

In this paper we are going to give more details about a real implementation of that architecture which is being developed in the context of the AXMEDIS European Project [5]. In particular, we will concentrate on how communications and services in the architecture can be secured and trusted, and which mechanisms have been introduced to ensure that client tools act as expected and are not modified by malicious users.

R. Meersman, Z. Tari, P. Herrero et al. (Eds.): OTM Workshops 2006, LNCS 4277, pp. 312 – 321, 2006.

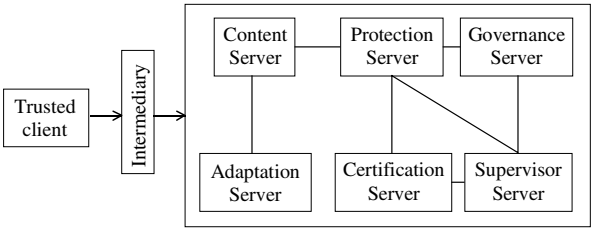

Fig. 1. DMAG MIPAMS architecture

In next sections we provide an overview of the AXMEDIS architecture. Then, we analyse the trust and security aspects and finally we provide a use case to understand how it works in a real scenario.

2 An Implementation of the Architecture

The architecture implemented in the AXMEDIS project consists on several independent modules that interact as web services when they are located in different machines or directly in other situations.

The general description of the AXMEDIS architecture main modules, depicted in Figure 2, is as follows:

- *Protection Processor.* This client tool module is responsible for estimating the client tool fingerprint, enabling or disabling the tool, verifying the tool integrity and unprotecting protected multimedia objects.
- *Protection Manager Support Client (PMS Client).* This client tool module manages and stores protection information, licenses, reports regarding the offline performed actions and other secured information in a local secure storage system called secure cache. It is responsible for authorising users to perform actions over objects with respect to digital licenses during offline operation. It also delivers protection information to the protection processor, if present in the secure cache, or requests it to the AXCS after a positive authorisation. It acts also as the intermediary module used by Protection Processor to contact AXCS to certify and verify tools.
- *Protection Manager Support Server (PMS Server).* This server side module is responsible for authorising users to perform actions over objects in an online environment and requesting protection information to the AXCS if needed. It acts also as an intermediary module to contact AXCS from PMS Client.
- *AXMEDIS Certifier and Supervisor (AXCS).* AXCS is the authority in charge of user and tool registration (Registration Web service), user and tool certification (AXMEDIS Certification and Verification, AXCV), user and tool management (e.g. status supervision, automatic blocking, deadline supervision, etc.), user and tool unique identifier generation and object metadata collection. AXCS is also responsible for saving the Protection Information related to protected multimedia objects as well as the actions performed on them (AXMEDIS Supervisor, AXS), the so-called Action Logs. Action Logs are the particular implementation of MPEG-21 [6] Event Reports [7] in the AXMEDIS context. AXCS also includes a

user Registration service, useful for registering new users in the system from distribution servers. All these data are stored in the AXCS database, which is accessed though the AXCS database interface module in order to keep the access independent from its implementation. Other functionalities provided by AXCS are those related to reporting and statistical analysis, which are performed by the Core Accounting Manager and Reporting Tool (CAMART module) by analysing the information stored in the AXCS database and collected in Action Logs. The integral modules of AXCS (see Figure 2) have been developed as web services or libraries.

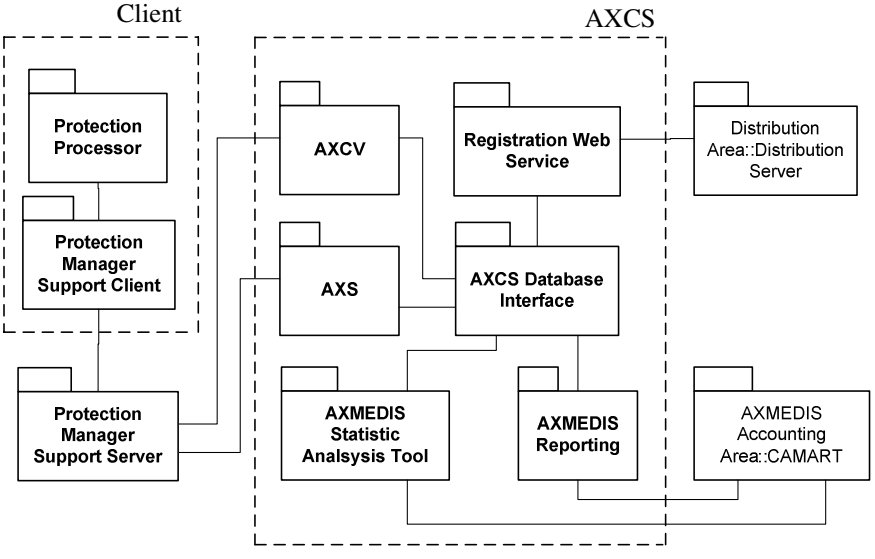

Fig. 2. AXMEDIS architecture regarding protection, rights management and accounting functionalities

2.1 Security and Trust on User Tools and Communication to Server

As we have mentioned in previous sections, in the client side we have different modules as Protection Processor and PMS Client which are devised to communicate with the server part by providing not only security to the transactions but also trust from the server side perspective. In the following sections we are going to describe the different mechanisms that the system includes to achieve the security and trust goals.

2.1.1 Registration of Users

All the users in the system must register, which enables their interaction with the system and system tools. User information is stored in the server side (AXCS) and is used for further verification purposes. After the user registration, the corresponding AXCS issues a user certificate that will be used to authenticate the user when performing some specific operations as the certification of tools (see section 2.1.2).

Every AXMEDIS user has associated a status that is used to determine whether the user is blocked or not in the system when interacting with the server part. The user status can be modified by the AXCS if some critical operation attempt is detected.

2.1.2 Registration of Tools

Tools using AXMEDIS framework must be verified to accomplish a series of guidelines, which are checked before registration is done. Once verified, each tool is registered for being used by AXMEDIS users. During registration phase, a fingerprint of the software tool is estimated so that its integrity can be checked later when the tool is installed and certified or verified on a specific device, as we will see in next sections.

2.1.3 Certification of Tools

The certification of a tool that uses AXMEDIS framework is a necessary step for that tool to work. Before a user is able to run and use a tool, the tool must connect to the AXCS to be certified as an "installed tool". Before installation, AXCS verifies the tool integrity by comparing its fingerprint to the one stored during the tool registration process and, once installed, extracts some information (tool fingerprint) concerning the installation of the tool and the device where it is installed.

A malicious user who tries to certify a tool whose fingerprint does not match the original registered tool fingerprint would be automatically blocked in the system so that he cannot continue performing other operations within the system. Moreover the tool would not be certified and thus it would not be operative.

Once a user successfully certifies a tool, any user of the system who owns a valid AXMEDIS user certificate can use it. Blocked users cannot use tools in the system.

To perform the certification of a tool, the tool connects to the AXCS via PMS Client and PMS Server web service. In order to have a secure communication, the client certificate is used to authenticate the user against the PMS Server.

The certification process involves different operations in the AXCS:

- *Generation of tool certificate and private key.* AXCS Certification Authority generates a tool certificate. It is used to establish secure communications, via SSL providing secure web services, to the PMS Server by any user that manages the certified tool. In this way we ensure that only certified tools can interact with the server part in an authenticated manner.
- *Generation of tool unique identifier.* A tool unique identifier is assigned to that specific installation of the tool and is used to identify it when interacting with the server side. The identifier is generated following the UUID format [8] and inserted in the tool certificate.
- *Generation of tool activation code*. A tool activation code is used to enable the tool operation. Some cryptographic algorithms that depend on the specific installation are used to generate it and they are inserted in the tool certificate as a certificate extension.
- *Generation of tool fingerprint.* The tool fingerprint, as we have already said, concerns the installation and the device where the tool is installed. This fingerprint is used in further verification process to determine if the tool has been manipulated

or if the device has changed or, in other words, to ensure the tool is still trusted in further executions.

- *Storage of identifier, activation code, tool fingerprint and certificate*. All the previous information is stored in the AXCS database and will be used to authenticate the tools that connect to the server part and to verify their integrity, as we will explain in next sections.

On the other hand, the certification process supposes also different operations in the client side (PMS Client and Protection Processor):

- *Reception of tool certificate, private key, tool identifier and activation code*. Regarding the tool certificate, private key, tool identifier and tool activation code, tool identifier and tool activation code are included in the tool certificate in the following manner (see Figure 3): 1) The tool unique identifier is used as the certificate common name (CN) in the subject distinguished name (DN) field; 2) The tool activation code is inserted as a certificate extension.

```
Data:
  Version: 3 (0x2)
  Serial Number: 1000000493 (0x3b9acbed)
  Signature Algorithm: sha1WithRSAEncryption
  Issuer: O=AXMEDIS, OU=AXMEDIS AXCS CA, C=ES, CN=AXMEDIS
AXCS CA/emailAddress=axmedis@axmedis.org
  Validity
     Not Before: …
     Not After: …
  Subject:O=AXMEDIS, CN=ITO_cdecb4a1-dbcb-362c-a30d-bb936342996c
  Subject Public Key Info:
     Public Key Algorithm: rsaEncryption
     RSA Public Key: (1024 bit)
        Modulus (1024 bit): …
        Exponent: 65537 (0x10001)
  X509v3 extensions:
     X509v3 Subject Key Identifier: …
     X509v3 Authority Key Identifier: …
     1.3.6.1.4.1.25576.1.1.1: …
Signature Algorithm: sha1WithRSAEncryption
  …
```

Fig. 3. AXMEDIS tool certificate fields

The tool activation code extension is identified with the Object Identifier 1.3.6.1.4.1.25576.1.1, where 1.3.6.1.4.1.25576 is the Private Enterprise Number assigned by IANA to AXMEDIS Organisation and 1.3.6.1.4.1 corresponds to IANA-registered Private Enterprises [9] (see Figure 4).

Current assignation of the AXMEDIS tree corresponding to the 1.3.6.1.4.1.25576 branch is the following:

1.3.6.1.4.1. 25576.0: reserved
1.3.6.1.4.1. 25576.1: AXMEDIS PKI-X.509 related objects
1.3.6.1.4.1. 25576.1.1: AXMEDIS Tool certificate extensions
1.3.6.1.4.1. 25576.1.1.1: AXMEDIS Tool activation code (or enabling code)

Fig. 4. Assignation tree corresponding to the AXMEDIS IANA Enterprise number

The tool certificate and private key are finally packaged by AXCS in a PKCS12 [10] structure protected with a password linked to the user that performed the certification and delivered over the secure channel established using the user and server certificates.

- *Storage of certificate and private key and tool activation*. The PKCS12 structure is accessed by Protection Processor in order to extract the tool certificate and private key, which are finally stored in a local keystore, and also to get the activation code used to enable the tool.

2.1.4 Secure Communication

As we have already mentioned, all communications between client tools and the server part are performed over a secure channel, which is established by means of client and server certificates, thus having authentication of both parties. Whereas before client tool certification client tools need to use user certificates, after certification they use tool certificates to create the secure channel with PMS Server. PMS Server also establishes a secure communication with AXCS by means of its own server certificate issued by the AXCS CA. It is worth noting that the certificates issued to users, tools and servers have different certificate purposes.

2.1.5 Verification of Tools

Verification of tools is devised to cover two functionalities. First, it provides a means to ensure that client tools have neither been manipulated nor corrupted. Moreover, verification is used to resynchronise all the actions performed by users during offline operation, that were stored in the local secure cache.

Verification of tools is performed periodically by the Protection Processor and every time the user tool resynchronises the offline performed actions with the server part. It consists on the verification of the estimated tool fingerprint in the moment of the verification against the tool fingerprint stored in AXCS database during the certification of the installed tool.

Regarding the tool integrity verification, if AXCS detects that critical parts of the tool or the device have been manipulated, it can adopt the pertinent measures as, for example, blocking the specific installed tool for which the verification failed.

Regarding the resynchronisation of offline performed operations, AXCS executes an algorithm to determine whether the received list of operations, which are called Action Logs in the AXMEDIS context, is complete with respect to the previous received operations. This integrity check is feasible thanks to the calculation of a fingerprint on the performed Action Logs, which is computed by PMS Client during the tool operation. This fingerprint is sent to AXCV when resynchronising the offline Action Logs and is verified by AXCV using the algorithm depicted in Figure 5.

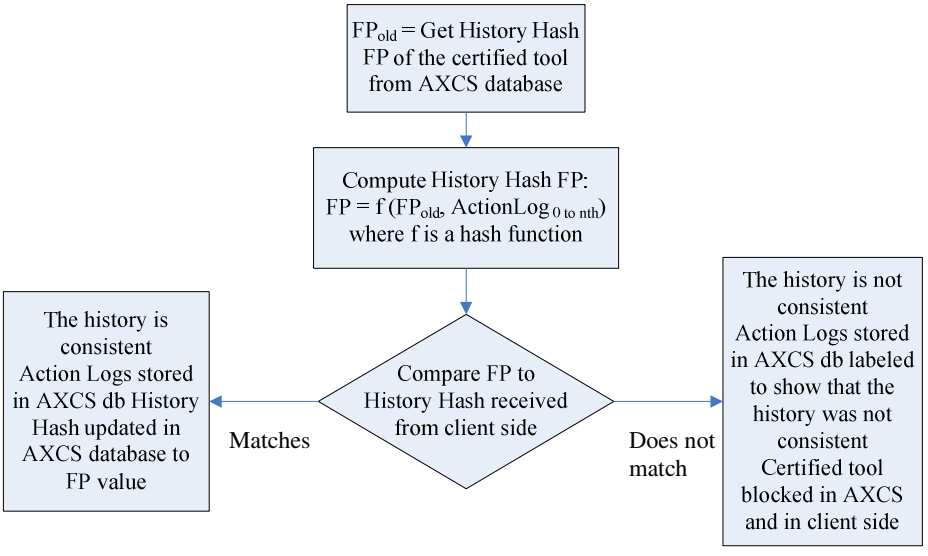

Fig. 5. Algorithm to determine the integrity of the received Action Log list in AXCV

The history fingerprint (FP) computation is also performed in the client side for each action performed in the online or offline operation, so that, once synchronised it must hold the same value in both PMS Client and AXCS. When an online operation is performed, this value is immediately synchronised in the server side. When any offline actions are performed, an Action Log associated to each of them is stored in the local secure cache, where PMS Client computes and separately stores FP value after the operation.

3 Use Case

In this section we present a scenario to illustrate how the proposed architecture and the processing of protected and governed multimedia content are related. It describes content consumption.

Imagine that a user has purchased online a license that grants him the right to play a movie during a certain period of time. The acquisition of the license could be performed in various ways. On one hand, the user could have obtained the license in the same place where he purchased the content. In this case, if the license needs to be customised for a particular user, the content distributor must request the license to the corresponding protection and governance servers. On the other hand, the user could have obtained the content through a P2P network, or other online or even offline distribution channels. In this latter case, the content must have some metadata that identifies the content server with which the user must interact to purchase the appropriate license.

The aforementioned user has, installed in his device, a specific tool or plug-in that manages the protected and governed objects of the proposed system and that is able to display them in the appropriate way.

The use case begins when the user downloads a protected and governed movie, opens it with his favourite player, which includes the appropriate plug-in and tries to watch it (Play movie). Although the plug-in has not been manipulated, the system needs to verify its integrity and certify it before allowing its operation.

Figure 6 shows the steps involved in the content consumption use case, which are the following:

1. The viewer requires unprotecting the movie to an internal trusted module, the Protection Processor.
2-3. Protection Processor estimates the installed tool fingerprint and connects to AXCS through PMS Client and Server in order to certify the tool.
4-5. AXCS successfully verifies user data and status and tool integrity with respect to registered tool.
6. AXCS sends Protection Processor a PKCS12 structure that contains tool private key and tool certificate with the tool identifier and activation code.
7-8. Protection processor stores tool certificate and private key in a local repository, extracts activation code and enables tool operation.
9-11. Before the authorisation, Protection Processor always calls verify method to check tool integrity and resynchronise offline Action Logs. In order to call it, it must reestimate the tool fingerprint and extract user and tool information from pertinent certificates.
12. PMS Client gets action logs from secure cache and contacts AXCS through PMS Server. (Note that in this case, as it is the first usage, there will not be any action logs)
13-17. AXCS verifies user and tool data with respect to certified tool Fingerprint, computes and verifies the operation History Fingerprint and stores received action logs in the AXCS database.
18. The result of the verification is sent to Protection Processor.
19. Protection Processor asks for authorisation and for protection information to PMS Client. As the user is working online, PMS Client contacts PMS Server.
20-21. PMS Server contacts AXCS to retrieve the object protection information.
22. PMS Server performs the license-based authorisation using its license repository.
23-24. As the authorisation is positive, PMS Server sends the pertinent Action Log to AXCS, which stores it in its database.
25. PMS Server notifies PMS Client the successful authorisation
26-27. PMS Client updates and stores the operation history hash fingerprint and the object protection information in the local secure cache.
28. PMS Client notifies Protection Processor the successful authorisation
29-31. Protection Processor requests the Protection Information to PMS Client, which retrieves it from the local secure cache.
32-33. Protection processor is capable of unprotecting the protected object so that the player can finally display the film to the user.

It is worth noting that, once the tool is certified, only verification process is done when the user wants to consume multimedia content. Steps 2 to 10 are no more executed after tool certification.

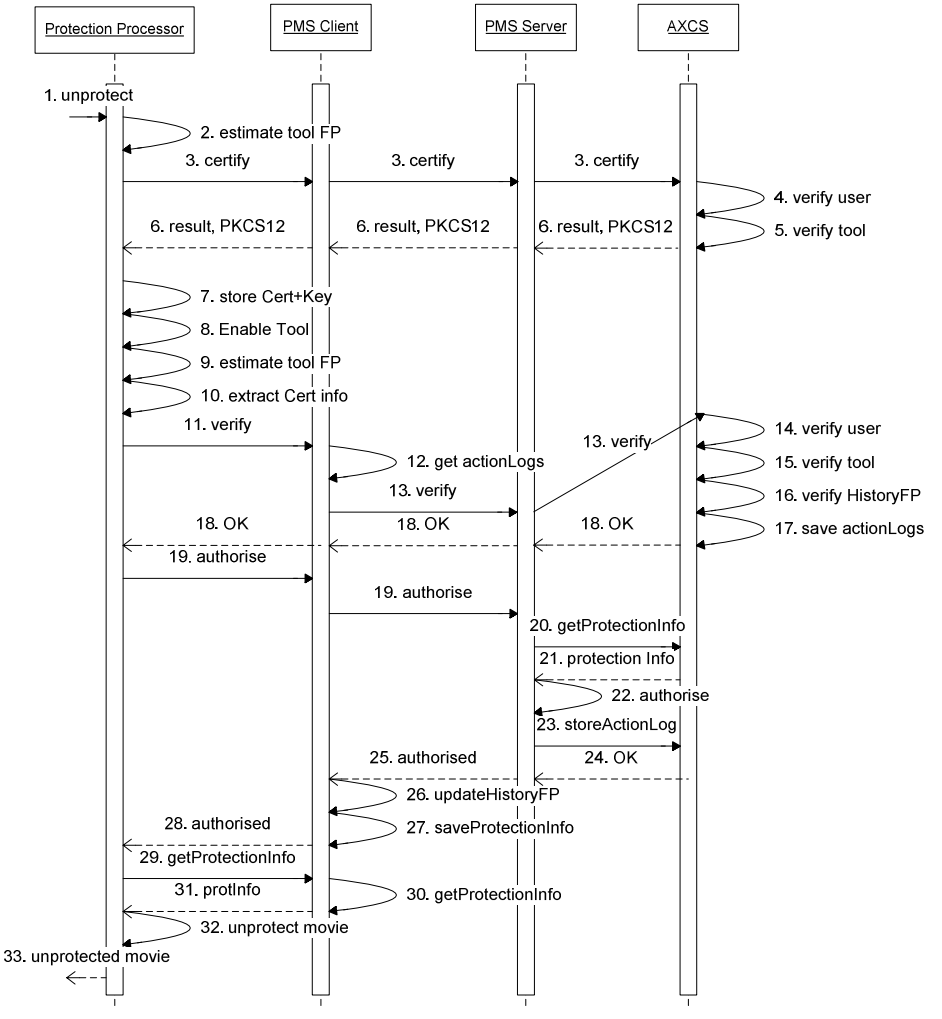

Fig. 6. Use Case

4 Conclusions

In this paper we have presented a possible implementation of the DRM architecture presented in [1] [2] [3], which is being developed in the context of the AXMEDIS European project [5]. In particular, we have concentrated in the aspects that provide security and trust to the interaction between user tools and the server part of the system, such as the registration of users and tools, the certification of tools, the establishment of secure channel communications using both client and server authentication and the verification of tools during their whole life operation. We have also provided a use case to illustrate the whole content consumption process.

Several demonstrators over different distribution channels (Satellite, PC, Kiosk, etc.) have been produced within the AXMEDIS project in order to validate the proposed solution and show its potential usage. Moreover, a public framework will be provided for the adoption of the AXMEDIS solution. Demonstrations of single tools and also of the framework are provided at AXMEDIS conferences and sometimes on the AXMEDIS portal [5]. The framework can be accessed by all affiliated partners.

The next steps to be tackled involve the integration with other existing content production and distribution tools in order to facilitate interoperability of both content management systems and multimedia and cross media protected objects.

Acknowledgements. This work has been partly supported by the Spanish administration (DRM-MM project, TSI 2005-05277) and is being developed within AXMEDIS [5], a European Integrated Project funded under the European Commission IST FP6 program. A special mention should be done to DSI-DISIT [11] for their collaboration in the work presented in this paper.

References

1. Torres, V., Rodríguez, E., Llorente, S., Delgado, J.: Trust and Rights in Multimedia Content Management Systems. Proceedings of the IASTED International Conference on Web Technologies, Applications, and Services (WTAS 2005). ACTA Press, Anaheim Calgary Zurich (2005) 89-94
2. Torres, V., Rodríguez, E., Llorente, S., Delgado, J.: Use of standards for implementing a Multimedia Information Protection and Management System. Automated Production of Cross Media Content for Multi-Channel Distribution (AXMEDIS 2005). First International Conference on. IEEE Computer Society, Los Alamitos Washington Brussels Tokyo (2005) 197-204
3. Delgado, J., Torres, V., Llorente, S., Rodríguez, E.: Rights and Trust in Multimedia Information Management. 9th IFIP TC-6 TC-11 Conference on Communications and Multimedia Security (CMS 2005). Lecture Notes in Computer Science, Vol. 3677. Springer-Verlag, Berlin Heidelberg New York (2005) 55-64
4. Distributed Multimedia Applications Group (DMAG), http://dmag.upf.edu
5. Automatic Production of Cross Media Content for Multi channel Distribution (AXMEDIS), IST 2004 511299, http://www.axmedis.org/
6. MPEG 21, http://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm
7. ISO/IEC, ISO/IEC FDIS 21000-15 – Event Reporting
8. A Universally Unique IDentifier (UUID) URN Namespace, http://tools.ietf.org/html/4122
9. Internet Assigned Numbers Entity (IANA) Private Enterprise Number, http://www.iana.org/cgi-bin/enterprise.pl
10. PKCS #12 v1.0: Personal Information Exchange Syntax Standard. RSA Laboratories, June 24, 1999, http://www.rsasecurity.com/
11. Distributed Systems and Internet Technology Lab - Department of Systems and Informatics DSI-DISIT, University of Florence, http://www.disit.dsi.unifi.it/