Environmental constraints management in digital right licences

Julien Thomas, Frédéric Cuppens, Nora Cuppens-Boulahia

Environmental Constraints Management in Digital Right Licences

Julien A. Thomas (julien.thomas@telecom-bretagne.eu)*
Frédéric Cuppens (frederic.cuppens@telecom-bretagne.eu)*
Nora Cuppens-Boulahia (nora.cuppens@telecom-bretagne.eu)*

Abstract: In the past years, Digital Rights Management (DRM) has been used in order to control media's resources usage, for instance for the consumers. Several proposals have been made in order to define what kind of controls one could apply and how to apply them. It has also been proposed more recently to use DRM mechanisms in the enterprises (EDRM), not to control media's resources usage but to protect the enterprise resources. However, current studies about EDRM do not take into account an important aspect of the enterprises security information: the environment.

In this article, we will define a framework to express environmental constraints based licences and to evaluate them. Examples will be given to show that the environment can be used to define more expressive licences and to apply better security policies for DRM based solutions.

Keywords: Environmental Constraints, Certifications, Digital Right Management, Licences

Introduction

In the past years, Digital Rights Management (DRM) has been used in order to control media's resources usage with immediate applications in the industry and the well known worldwide controversy it has provoked. In order to apply this notion of rights to contents, several proposals have been made in order to define what kind of controls a provider could apply and how to apply them. These proposals rely on licences, which define *who* is able to do *what* on *what* under *which circumstances*. Among them, MPEG-REL [ISO03] (previously XrML) and OMA-DRM [Ini02] (previously ODRL) are the most famous proposals. More recently, several attempts [AH07, DSG$^+$04] have been made to adapt this notion of content controls for the enterprises, not to control media's resources usage but to control and protect the enterprises' resources.

However, an important aspect of the enterprises security information has not been taken into account: the environment. Many organizations tend to pass certifications with respect to governmental laws such as Sarbanes-Oxley, HIPAA or governmental requirements such as PRIS [ADA05] (France), the Department of Defense Architecture Framework [Gro03] (United States) and the Ministry of Defense Architecture Framework [Par05] (United Kingdom). These documents provide a guide to define a way to manage efficiently sensitive information and to assess that organizational processes are well defined.

* TELECOM Bretagne, SERES team, 2 rue de la Châtaigneraie, CS 17607,35576 Cesson Sévigné Cedex

Based on these certifications, organizations can for instance assess that the DRM-rendering applications are based on a Trusted Computing Base (TCB) [LABW92] or that the user identity is provided by an identity provider compliant with the French security requirements described in PRIS. Such assessments can be used to evaluate the security of the end-user environment. We thus argue that environmental information can be used to define more expressive DRM controls and apply better security policies inside DRM based solutions.

Our article is organized as follows. In section 1, we present basic notions such as certifications and environmental information. We also present existing DRM systems and their lack of expressivity for EDRM purposes. In section 2, we describe our environmental constraints management proposal, with formal models and evaluation principles. We also introduce with an example how our proposal can be introduced into existing DRM standards (XML-based), as an extension of the existing constraints. In section 3, we introduce the implementation of our environmental constraints management process.

1 Related Works

1.1 Environmental Constraints

We claim that environmental constraints describe environment aspects that may impact the security and thus have to be expressed in DRM licences. In this section, we describe several environmental notions mentionned in the literature.

The Analog Hole [MPA02, SC05] has been introduced in 2002 in a federal court of the US by the Motion Picture Association of America [MPA02]. Also named analog reconversion issue, it describes an issue that DRM-based solution still have to face: the operations that are not computer-based and thus not controllable with DRM-based mechanisms. For instance, no DRM-solution can counter the operation that consists in hand-writing the content that one is reading and then mailing it. This issue into account has been an important challenge (e.g. the Digital Transition Content Security Act of 2005 [SC05])

The execution environment is also a point to take into account for security enhancement purposes. For the (E)DRM purposes, we found the following relevant information. This enumeration is an example of relevant information. In section 2, we will propose a generic and thus extensible model.

- the software environment. If the software is certified compliant with a norm N, modification-detectable or deployed on a TCB ([LABW92]), we do not put the same trust in it as when it has no certification at all. Besides, one may not have the same trust in the same software tunning on different platforms.

- the organization environment. If the organization is certified compliant with a norm N, we do not put the same trust in the DRM-based software as when the firm has no certification. For instance, it may be easier to plug the Analog Hole in a governmental organization than at home.

- the user environment. Whether the user identity is well managed (due to the enforcement of some security and quality procedures inside the organization, as described for instance in the French specification PRIS [ADA05] with the security level *** (the highest security level in PRIS)) or not (such as a shared account) makes the user identity trustfully or not.

Finally, based on the asymmetric encryption and signature mechanisms (see [Den82] for an introduction about encryption), certifications are also important for the evaluation of the organizations environment. Nowadays, certificates have been standardized (e.g. X.509 certificates) and several international organizations are accredited to generate them. The signature principle is the core of the certification mechanism: the use of the chain principle provides a lightweight certification mechanism. For instance, it is widely used on the web (SSL certificates for web-transactions, websites identifications, ...) or for any digital certificates. In the literature, several attempts have been made to study certifications mechanisms and to model them. We can mention the ETSI model ETSI TS 11 456 [ETS02] which describes the requirements certification authorities have to meet in European countries. For instance, the French specification PRIS [ADA05] is an application of the ETSI requirements for the French governmental organizations and the end-users that want to communicate with them.

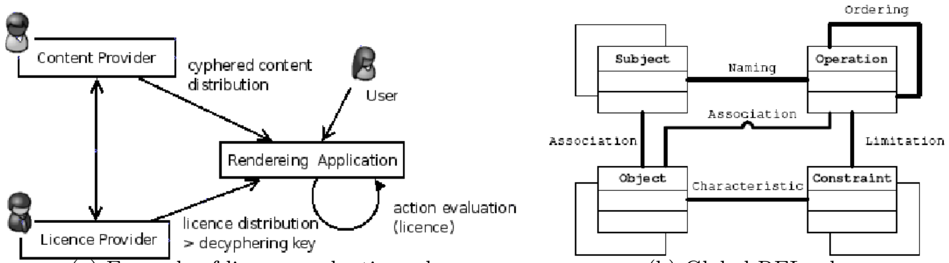
1.2 Digital Rights Management Overview

(E)DRM Principles DRM Solutions are encryption based frameworks which provide a way to control who can decipher the encrypted content and for which purposes. As described in the figure 1a, the global framework can consider (at least) three parties: the content provider, which distributes the protected content, the licence provider and the rendering application with the considered user. DRM solutions provide a way to specify which interactions are possible.

In order to specify the authorizations, the content provider will define a licence, which will specify allowed interactions between the subject (i.e. the considered user) and the object (the content). The figure 1b defined[1] by Parrott [Par01] shows the different components of the DRM-licences. Using our licence language model, we will be able to define specific constraints such as the considered rendering application or the end-user identity affiliation. Many proposals for licences language, called Right Expression Languages (RELs), can be found in the literature and some of them were standardized. We can mention the two most famous ones, MPEG-REL [ISO03] and OMA-DRM [Ini02]. These two RELs are based on XML models and are used by several companies (these two standards are supported by different consortiums). Other proposals, not based on XML, have been proposed. For instance, LicenseScript [CCE$^+$03] is based on Prolog. OpenIPMP [MM06] and MPEG REL SDK [Con08] are two examples of the existing DRM projects.

More recently, it has been proposed ([AH07, DSG$^+$04]) to use DRM mechanisms in the enterprises, no to control media's resources usage but to control the organizations' resources usage and protect sensitive information. These solutions, named EDRM (Enterprise DRM) provide a new way to manage documents security as the documents can still be controlled even if they are not inside the companies access control perimeter anymore.

[1] Note that the DRM structure definition is out of scope. Thus, we do not discuss it here.



(a) Example of licence evaluation scheme

(b) Global REL scheme

Figure 1: DRM Principles

EDRM solutions are most of the time seen as applications of DRM specifications to Enterprise, such as implementation of DRM solution in existing Content Management Systems (CMSs). Besides, enterprises environmental information (such as certifications) are not modelled in current proposals, though they provide relevant information to enhance the evaluation of the environment security.

Lacks Of Expressivity of the Existing RELs In the current right expression languages, models are used to describes the RELs expressivity. Considering models bound to the environmental constraints issues, for instance in ODRL, we found two relevant model:

- The Constraint Model specifies information such as Users, Devices or Aspects, which are obviously bound to our environmental constraints
- The Security Model supports encryption standards, for ODRL licence elements

Though these models express security and environmental information, they do not take into account environmental constraints. Requirements such as «parts of the organization which are certified PRISv2 ***» can not be expressed with current RELs as the Constraints model does not take environmental information into account for the security issues.

2 Environmental Constraints Management

As introduced in the previous section, several concepts can be implemented in (E)DRM models to specify organization contexts. As existing RELs fail to express these contexts (see section 1.2), we present in this section our environmental constraints management model (section 2.2). We first specify in section 2.1 an environment analysis model. Using this evaluation model, which can be seen as an extension of our basic environmental constraints proposal, we obtain a more expressive environmental constraints management model. Finally, contexts evaluation and the identity control issue are presented.

2.1 Environment Analyze (Certification) Model

2.1.1 Information providers

The issue consists in modeling where the information about the environment come from. The end-user based evaluations and provider based evaluations, which respectively refer to performing the environmental constraints evaluations using information from the end-user and using information from the content provider, are two interesting solutions but not relevant in our case. The following two paragraphs describe why, in our case, these models fail. We then present our third-party based proposal.

End-user based evaluations irrelevance As DRM-based software are supposed to be trusted, it may be interesting to perform local evaluations of the environment: the software analyzes the environment and then evaluates the environmental constraints. However, this approach fails for at least two reasons:

- the trust in the software depends on several conditions such as the considered platform or the user privileges. With nowadays DRM-software, we can not assure that the software are totally trustworthy: no TCB, no resilience to modifications, ...

- the evaluation of the environment requires environmental information. Even if the software is trustworthy, we can not assure that the software can obtain the required information or that these information are correct.

Provider based evaluations drawbacks A provider based evaluation is a good way to evaluate environmental information, or at least to trust the information. However, this solution implies requirements that are not required for current DRM solutions: as every constraints evaluation needs the content provider evaluations about the considered environment, the content provider has to be always available. This solution is thus incompatible with superdistribution scenarios.

Third Parties based evaluations Third parties based evaluations combine the benefits of the two previous solutions: as the licence provider and the software trust these third parties, information can be trusted and no control from the licence provider is required.

As described in FORM [TS06], a common way to generalize DRM framework is to use remote evaluations. In FORM, third parties information are used for entities identification, with the identity providers. In our solution, we use third parties information for the evaluation of the environment: accredited authorities can evaluate the environment and assure that the entities respect the specified requirements. The certificates are used by these third parties to assure that they are the issuers of the environment evaluations. Besides, it is compliant with actual applications, in which certificates and environment evaluations are provided by accredited authorities.

2.1.2 Certification Model

As presented in the section 1, PRIS [ADA05] is a proposal made by the French government to define security requirements for governmental organizations and the end-users that want to communicate with them. In this referential, the ETSI [ETS02] proposal is used and refined, to define precise requirements about the certification authorities, such as security levels (PRIS*, ** and ***) about the cryptographic techniques they use. Thus, these two proposals provide information about certification authorities (at least the European ones) and which of their components are considered as relevant for security purposes. Based on these proposals, we define a model to express security requirements for certification authorities. By integrating this certification model in environmental constraints, we are thus able to express the accredited authorities security levels.

This model describes the components of a certification authority, such as the publication service (`Publication service`), and associates to these components

- the security level `SECURITY_LEVEL` of each service they provide.
- the security level `OPTIONAL_LEVEL` for the services that are optional, which consists in defining a security level `SECURITY_LEVEL` or stating that the operation is `REFUSED`.
- the norm `SECURITY_NORM` the services are compliant to, for the services that are not associated to a security level, as the certificates validation of the publication service.

```
Publication service := is certified <SECURITY_LEVEL>
|| modification_access_control is certified <SECURITY_LEVEL>
|| certificate_deliverance is certified <OPTIONAL_LEVEL>
|| certificate_acknowledgement is certified <SECURITY_LEVEL>
|| certificate_validation is certified <SECURITY_NORM>

Modification service := is certified <SECURITY_LEVEL>
|| certificate_update is certified <OPTIONAL_LEVEL>
|| certificate_modification is certified <OPTIONAL_LEVEL>
|| certificate_regeneration is certified <OPTIONAL_LEVEL>

Registration service := is certified <SECURITY_LEVEL>
|| identity_validation is certified <SECURITY_LEVEL> for <ENTITY_CATEGORY>

Generation service :=  is certified <SECURITY_LEVEL>
|| user_key_generation is certified <SECURITY_LEVEL>

Revocation service := is certified <SECURITY_LEVEL>
|| identity_validation is certified <SECURITY_LEVEL>
|| originator is certified <SECURITY_NORM>
|| LCR_management is certified  <SECURITY_LEVEL>

Global service :=  is certified <SECURITY_LEVEL>
|| roles_separation is certified <SECURITY_LEVEL>
|| physical_access is certified <SECURITY_LEVEL>
|| backups are certified <OPTIONAL_LEVEL>
|| CA_key_generation  is certified <OPTIONAL_LEVEL>
|| CA_key_control is certified <OPTIONAL_LEVEL>
|| cryptography is certified <SECURITY_LEVEL>
|| signature is certified <SECURITY_LEVEL>

SECURITY_LEVEL := (AT_LEAST)? <SECURITY_LEVEL_ELEMENT>
OPTIONAL_LEVEL := REFUSED || <SECURITY_LEVEL>
ENTITY_CATEGORY := PARTICULAR || ORGANIZATION ||
ORGANIZATION_W_CertificationMandator || CertificationMandator
```

Based on the PRIS proposal, we define the security levels `PRIS *`, `PRIS **` and `PRIS ***` where the requirements needed to meet the security level `PRIS ***` encompass the requirements of the security level `PRIS **`, which encompass the requirements of the security level `PRIS *`. We thus have `PRIS *` < `PRIS **` < `PRIS ***` and

```
SECURITY_LEVEL_ELEMENT := PRIS* || PRIS ** || PRIS ***
SECURITY_NORM := PRIS
```

2.2 Environmental Constraints Management Model

We can now define the environmental constraints. In the subsection 2.2.1, we present the grammar the environmental constraints rely on. In the other subsection, we illustrate our grammar with examples.

2.2.1 Environmental Constraints Expression Model

The environmental constraints expression model is used to express environmental constraints. It must let users express their requirements with a grain as fine as possible:

- constraints may be combined to express requirements such as «Constraints1 and (Constraints2 or Constraints3)». `CONNECTOR` allows such requirements.

- the environmental constraints model may be defined or refined. The use of `environments_constraints_element` allows users to extend our model.

This model, of type LL(k), is as follows:

```
environments_constraints :=  [environments_constraints_element]
|| (environments_constraints CONNECTOR environments_constraints)
|| NOT (environments_constraints)
environments_constraints_element := <software_environment_constraint>
|| <organization_environment_constraint>
|| <user_environment_constraint>
CONNECTOR := AND || OR
```

Environmental Constraints Taxonomy The taxonomy of the environmental constraints is a way to evaluate the expressivity of the model. We present in this section the environmentals constraints we need to express basic constraints about the entities identity and the software environment. They extend the common constraints of standardized DRM licences about the rendering application (software) and the end-user (organization environment and user environment), the subjects of our environmental constraints.

```
software_environment_constraint := SOFTWARE: <s_e_c>
s_e_c:= "is certified " <SOFTWARE_CERT> " by " <CERT_AUTHORITY>
|| "id is certified " <ID_CERTIFICATION> " by " <CERT_AUTHORITY>
|| "id_management is certified " <ID_MGMT> " by " <CERT_AUTHORITY>

organization_environment_constraint := ORGANIZATION: <o_e_c>
o_e_c:= "is certified " <ORGA_NORM> " by " <CERT_AUTHORITY>
|| "id is certified " <ID_CERTIFICATION> " by " <CERT_AUTHORITY>
|| "id_management is certified " <ID_MGMT> " by " <CERT_AUTHORITY>

user_environment_constraint := USER: <u_e_c>
u_e_c:= "id is certified " <ID_CERTIFICATION> " by " <CERT_AUTHORITY>
|| "id_management is certified " <ID_MGMT> " by " <CERT_AUTHORITY>
```

Based on the information about the environments described in section 1.1, we use the following certification values, where `CERT_AUTHORITY` refers to the certificaton model and the certification value `CONFINED` is introduced in section 2.4.

```
SOFTWARE_CERT := TCB_BASED || UNMODIFIABLE ; ID_MGMT := CONFINED
ID_CERTIFICATION := ANY || UNIQUE ;
ORGA_NORM := PRIS || PRIS* || PRIS** || PRIS***
```

2.2.2 Examples

In the introduction, we have stressed the lack of expressivity of current RELs, to express environmental constraints. In this section, we present examples of new constraints that can be expressed with our model. We first present constraints about the organizations environment then about the entities identity and finally about the software environment.

Organization constraints We can specify constraints about the organization security level, which can be abstracted by the certificates it possesses. For instance, a French organization that meets the security level PRIS *** of the PRIS [ADA05] proposal is certified EAL4+ and uses RSA keys of 2048 bytes. If it behaves as a certification authority, it also satisfies other requirements such as a cryptographic module certified ELA4+ for the generation of the private keys of the certification authority and the recipients certificates.

Thus, a constraint such as «organizations that are certified PRIS ***» allows the usage of the contents in organizations that meet strong requirements. This constraint can be expressed in our model by `[ORGANIZATION: is certified PRIS*** by X]`.

Identity constraints Current models allow the use of identity providers to identify users, software and so one and thus to express constraints such as `<<id is provided by X>>`. However, it is not possible to specify the identity provider security-level, such as `ANY which is certified TRUSTED_LEVEL by X` and the identity security.

We can now express requirements such as the unicity of the user identity with `[USER: id is certified UNIQUE by X]` and the security level of the identity provider with `[USER: id is certified ANY by X]` where we put the requirement `is certified PRIS ** by Y` for the certification authority `X`, as described in the section 2.1.2.

Software constraints The final part of our examples concerns the software constraints. These constraints, checked for a first time before the rendering application obtains the deciphering key, can be used to trust the rendering application. For instance, the following requirements assure that the software will at least not be modified and that the superdistribution process (`SUPERDISTRIB_CERTIFIED`) will be respected:

```
([SOFTWARE:is certified UNMODIFIABLE by X] OR [SOFTWARE:is certified
TCB_BASED by X]) AND [SOFTWARE:is certified SUPERDISTRIB_CERTIFIED by X]
```

2.3 Environmental Constraints Evaluation

The evaluation of our constraints is similar to the evaluation of any existing DRM licence: before allowing an action, the licence is evaluated to check if the action meets its requirements and constraints. In this section, we thus describe the global scheme of our evaluation, an extension of existing evaluation scheme, based on certificates evaluation.

As the environment evaluation is based on third party analyses, the evaluation of the environmental constraints consists in analyzing the certificates issued by these certification authorities. For each certificate, we have to control:

- the entity identity and the certificate recipient
- the certification and the environmental constraint requirements (e.g. `TCB_BASED`)
- the certificate issuer and the certification authority of the licence.

As the validity of the entity identity is checked by current DRM schemes, we do not have to implement it in our evaluation scheme. However, we must assure that the Common Name used for the identity control and the one used for the certificates evaluation are the same. This process is summarized in figure 2. We extended the standard DRM evaluation process (full arrows) with the environmental constraints evaluations. These new evaluations rely on the certificates provided by the accredited third parties and the licence information.

Software Constraints Issue As one can specify software constraints, these constraints have also to be validated before the key is sent to the software. As mentioned previously, the organization identity must be checked too, with a trustfully process (different from the organization identity checking process, as we consider here a distribution scheme). This requirement is illustrated in the figure 2.

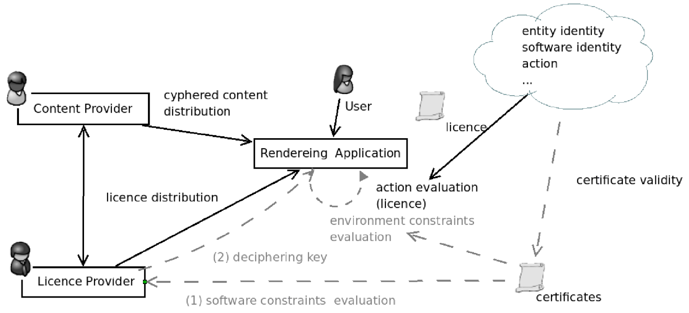


Figure 2: Environmental Constraints Evaluation Principle

XML Transformations for Compliance with XML-Based Licences As presented in the section 1, the current DRM standards, MPEG-REL [ISO03] and OMA-DRM [Ini02], are XML-based. In this paragraph, we thus present how our environmental constraints can be be transformed into a XML format. The evaluation of our constraints «only» requires, then, the implementation of a licence evaluation extension in current DRM tools.

Consider the following example, simplified due to space limitation: the content provider wants to be assured that the rendering application is based on a trusted computing base with the constraints `[SOFTWARE: is certified "TCB_BASED" by [XID, "XKEY"]]` where `[XID, "XKEY"]` identify a certification authority the content provider trusts. Using our generic transformer, this constraint can be transformed into a XML constraint:

```
<software_constraint>
   <certification_value>TCB_BASED</certification_value>
   <certification_authority>
      <certification_ID>XID</certification_ID>
      <certification_key>"XKEY"</certification_key>
   </certification_authority>
</software_constraint>
```

2.4 Identity Control: the main issue

As we will see in this section, the identity Control is a main issue of our model. Users dependent certificates and organization certificate based controls are also discussed in this section, to show why the users dependent certificates solution is better.

The Identity Control Issue In the previous section, we defined a scheme to evaluate the end-user environment, before authorizing the use of DRM-based contents and the deciphering key distribution. However, the certifications used in this scheme are linked to the organization where the environment evaluation is performed. Thus, the evaluation of environmental constraints depends on sensitive information: the organization identity. For instance, deployment of software are performed in a company and thus certificates about the environment of the software depend on the company identity.

The organization identity is thus essential in the process: if it is not well protected, one can use at home the identity he was given by his company, pretending being at work. The best way to enforce this requirement is to define a certification rule such as `[ORGANIZATION: id_management is certified CONFINED by X]`, where `X` is an entity we trust for the confinement evaluation.

Identity Checking Model Checking the organization identity can be performed using two different approaches. The first approach relies on a single and well protected global certificate, attesting the organization identity. However, in this case, the security of the whole organization relies on the non-compromission of a single private key, the one associated to the certificate.

In order to enhance the protocol robustness, we chose the second solution. It is a decentralized identity checking process, based on the users' organization certificates. This process can express the same functionalities as the centralized one but mitigates the impact of the lost/diffusion of a private key: in such a case, only a user certificate would have to be revoked and regenerated, not the organization one. The process, summarized in the figure 3, consists of the three following steps.

- the user logs in the DRM system and uses his/her confidential access key, which is also his/her certificate deciphering key
- the software is able to decipher the protected certificates

- the software can then use the identity of the user and its organization

The use of ciphering techniques for the storage of the certificates protects these sensitives information from compromission by physical access.

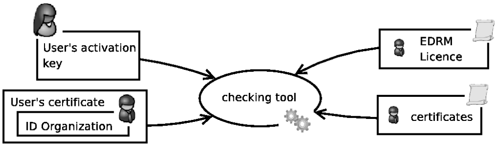


Figure 3: Identity Evaluation Model

3 Implementation of the Environmental Constraints Management Process

As described in section 2.4, the organization identity control is the core issue of the environmental constraints management. In this section, we thus choose to describe the implementation of such a control process. We first present the functional aspects of our tool and then the confinement module, to assure that nobody except the identity control process can access to the sensitive information. This second part describes a way to assure the control of the organization identity and be certified `[ORGANIZATION: id_management is certified CONFINED by X]`.

3.1 Identity Checker

For the Identity Checking Process, two actions have to be performed: the certificate deciphering action and the certificate analysis action. We can consider these operations as two actions of the same process and develop a single module but we considered two modules for several reasons described below. The figure 4 summarizes our Identity Checking Process, tested with an implementation developed in C and based on the CryptLib [Gut08] library.

- Several encryption standards already exist and using a deciphering module does not imply a specific encryption technique for the model.
- Certificates analysis is independent of the certificates ciphering technique. One can thus develop a personal certification platform or use an existing standard.
- From a security point of view, the two modules do not require the same authorizations (see subsection 3.2.1). We can thus define a more fine grained policy.

3.2 Identity Certificates Confinement

In the previous section, we have presented the functional aspects of our confinement module. However, the identity checking process (processes and data) must be confined in order to enhance the global security. We will then be able to asset that each part of our process can access only the information it needs (for instance, the deciphering tool must not be

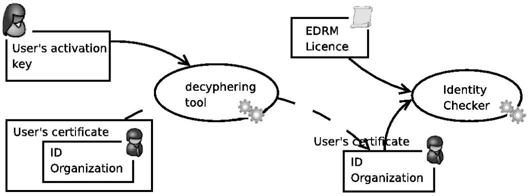


Figure 4: Identity Checking Model

able to leak information about the certificates) and that other parts of the system are not able to access to sensitive information, which is required to get the desired certification.

In this section, we thus describe the implementation of the identity checking process confinement module. The implementation of such a confinment requires the use of a flow control mechanism, at the operating system layer. SELinux [LS01] is an example of such implementation in Linux. In a first subsection, we present the SELinux security model. This let us define the minimal trust we need in each module of the process. In the next subsection, we briefly present the different parts of our module, in order to show how we have implemented the confinement module.

3.2.1 SELinux Security Model

The Security model is an extension of the identity checking model (figure 4). In our module, we first (figure 5) define domains and types labels, in order to formalize the environment. We have deployed the identity checking module in `/etc/drm`, labelled `etc_t`, and the deciphering tools decipher certificates in `deciphered_certs/`, labelled `certs_DRM_dir_t`. The SELinux environment is important for the module deployment as labels have to be correctly instantiated to guarantee that the module behaves securely: domains have to be well defined in order to assure that only the desired domain can access the specified information and the security policy defined the authorized translations between the domains. The modules authorizations are described in the table 1.

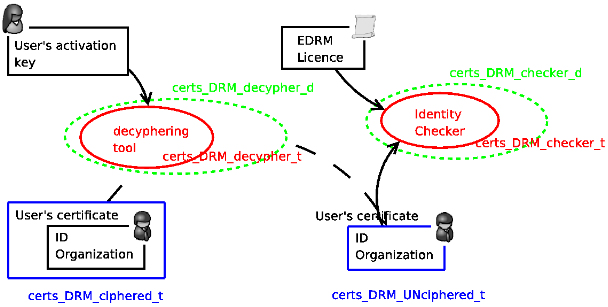


Figure 5: Identity Checking Model in SELinux Environment

This model makes it possible to have a fine grained security policy:

- the deciphering tool must not generate false deciphered contents or false answers must be at least detectable, for instance in case of an incorrect behavior. As any output is labelled `certificates_DRM_deciphered_t` and any unspecified flow is forbidden, the deciphering tool cannot leak intentionaly any information.

- the checking tool must be trusted, as it reads deciphered certificates. and thus may disclose confidential information.

If these two requirements about the process modules are satisfied, the identity checking confinement certification (associated to the constraint `[ORGANIZATION: id_management is certified CONFINED by X]`) is also satisfied.

Table 1: Identity Checking Model Authorizations

Domain	Data	Access
staff_t	certificates_DRM_decipher_t	execute
staff_t	certificates_DRM_OrgaChecker_t	execute
certs_DRM_decipher_d	certificates_DRM_ciphered_t	read
certs_DRM_decipher_d	certificates_DRM_deciphered_t	write
certificates_DRM_checker_d	certificates_DRM_deciphered_t	read

3.2.2 SELinux Model Environment

The identity checking confinement module has been implemented in Gentoo (see [Fou]) with the LSM (Linux Security Module) SELinux [LS01]. In order to simplify the example, we only describe the deciphering tool rules, as they are similar to the identity checking tool rules, except the authorizations ones.

Types and domains of the SELinux Model The first part of the SELinux policy specify our SELinux environment presented in the section 3.2.1.

```
type certs_DRM_decipher_d;
typeattribute certs_DRM_decipher_d domain;
role staff_r types certs_DRM_decipher_d;
type certs_DRM_decipher_t; type certs_DRM_ciphered_t;
type certs_DRM_deciphered_t; type certs_DRM_dir_t;
```

Flows control Another part of the SELinux policy defines the transitions between the domains. For instance, we must specify the following rules:

- who (user's domain) can execute the deciphering tool
- which user and which program can enter the `certs_DRM_decipher_d` domain

This is defined by the following SELinux statements:

```
type_transition staff_t certs_DRM_decipher_t:process certs_DRM_decipher_d;
allow certs_DRM_decipher_d certs_DRM_decipher_t:file entrypoint;
allow staff_t certs_DRM_decipher_t:file {execute getattr read ioctl};
allow staff_t certs_DRM_decipher_d:process
        {transition noatsecure rlimitinh siginh signal sigchld};
type_transition certs_DRM_decipher_d certs_DRM_dir_t:file
        certs_DRM_deciphered_t;
```

Authorization Rules Finally, we specify the authorization rules defined in table 1 about the identity certificates: who can read (or more generally obtain information) what? Who can write (or more generally modify information) on what?

```
allow certs_DRM_decipher_d certs_DRM_ciphered_t:file {read getattr};
allow certs_DRM_checker_d certs_DRM_deciphered_t:file {read getattr};
allow certs_DRM_decipher_t etc_t:dir { getattr search read lock ioctl};
allow certs_DRM_decipher_t etc_t:file { getattr read lock ioctl };
allow certs_DRM_decipher_d certs_DRM_deciphered_t:file  {create ioctl
        getattr lock write setattr append link unlink rename};
allow certs_DRM_decipher_d certs_DRM_dir_t:dir {getattr search read
        lock ioctl write setattr append add_name};
allow certs_DRM_deciphered_t fs_t:filesystem {associate};
```

Conclusion and future works

In this paper, we have proposed an extension of current (E)DRM model in order to be able to express environmental constraints. These constraints, such as «DRM-rendering applications are based on a trusted Computing Base (TCB)», can be used to enhance the security of DRM based solutions. We have also presented how this proposal can be included in existing XML-based licences, as an extension of the existing constraints.

We have then stressed the fact that the organization identity is a central problem for environmental constraints, as the DRM-based tools must be able to know with certainty in which organization it is. We have thus proposed a global process for the control of the organization identity, with an example of implementation, in SELinux, to assure the validity of the organization identity.

In future works, we will study the implementation of our solution with concrete licences and use our proposal to enhance existing content management proposals.

Acknowledgements

The study is funded by the DGA under a DGA/CNRS grant.

References

[ADA05] SGDN DCSSI ADAE. Politique de référencement intersectorielle de sécurité - service de signature - politique de certification type, Juin 2005. version 2.0.

[AH07] Alapan Arnab and Andrew Hutchison. Persistent access control: a formal model for drm. In *DRM '07: Proceedings of the 2007 ACM workshop on Digital Rights Management*, pages 41–53, New York, NY, USA, 2007. ACM.

[CCE+03] C.N. Chong, R. Corin, S. Etalle, P. Hartel, W. Jonker, and Y.W. Law. Licensescript: a novel digital rights language and its semantics. *Web Delivering of Music, 2003. 2003 WEDELMUSIC. Proceedings. Third International Conference on*, pages 122–129, 15-17 Sept. 2003.

[Con08] ContentGuard. MPEG REL SDK 1.0 for Java, 2008.

[Den82] Dorothy Elizabeth Robling Denning. *Cryptography and data security.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1982.

[DSG$^+$04] D. Doest, A. Salden, G. Goedvolk, R. Kersemakers, D. Slijp, and C. Prins. Digital rights management in information publishing. *Information & Communications Technology Law*, 13(2):99+, 2004.

[ETS02] European Telecommunications Standards Institute ETSI. *ETSI TS 101 456 - Policy requirements for certification authorities issuing qualified certificates*, April 2002. v1.2.1.

[Fou] Gentoo Foundation. *Gentoo SELinux Handbook.*

[Gro03] DoD Architecture Framework Working Group. *DoD Architecture Framework*, August 2003. v 1.0.

[Gut08] Peter Gutmann. The CryptLib project, 2008.

[Ini02] ODRL Initiative. *Open Digital Rights Language (ODRL)*, August 2002.

[ISO03] ISO/IEC 2004. *Information Technology - Multimedia Framework (MPEG REL) - Part 5: Rights Expression Language*, August 2003.

[LABW92] Butler Lampson, Martín Abadi, Michael Burrows, and Edward Wobber. Authentication in distributed systems: Theory and practice. *ACM Transactions on Computer Systems*, 10(4):265–310, 1992.

[LS01] Peter Loscocco and Stephen Smalley. Integrating flexible support for security policies into the linux operating system. In *Proceedings of the FREENIX Track: 2001 USENIX Annual Technical Conference*, pages 29–42, Berkeley, CA, USA, 2001. USENIX Association.

[MM06] Chris Mollis and Craig Miller. OpenIPMP project, 2006.

[MPA02] Motion Picture Association of America MPAA. Content Protection Status Report I, II & III, April to November 2002.

[Par01] David Parrott. Requirements for a rights data dictionary and rights expression language version 1.0. Technical report, Reuters Ltd, June 2001.

[Par05] MODAF Partners. *MOD Architectural Framework. Technical Handbook*, August 2005. v 1.0.

[SC05] James Sensenbrenner and John Conyers. Digital Transition Content Security Act of 2005, December 2005.

[TS06] Nora Cuppens-Boulahia Thierry Sans, Fédéric Cuppens. Form: A federated rights expression model for open drm frameworks. In *11th Annual Asian Computing Science Conference "Focusing on Secure Software and Related Issues" (ASIAN'06)*, pages 45–59, December 2006.