OPA : Onion Policy Administration Model - Another approach to manage rights in DRM

Thierry Sans, Frédéric Cuppens, and Nora Cuppens-Boulahia

GET/ENST Bretagne,
2 rue de la Châtaigneraie, 35576 Cesson-Sévigné Cedex, France
{thierry.sans,frederic.cuppens,nora.cuppens}@enst-bretagne.fr

Abstract. Digital Rights Management frameworks (DRM) aim at protecting and controlling information contents widely distributed on client devices. Using a license, the content owner specifies which rights can be rendered to end-users. Basically, only the content owner must be able to define this license, but some DRM models go further. In super-distribution scenario, the content owner does not directly manage end-user's rights but rather delegate this task to a third-party called a distributor. Nevertheless, this distribution cannot be done without any control. In existing approaches, the content owner restricts the license issued by the distributors. In this paper, we provide a new approach, called the Onion Policy Administration approach (OPA). Rather than restricting licenses issued by the different distributors, OPA aims at controlling which rights are finally rendered to end-users. The main idea of OPA is to have a traceability of the content distribution. The content must keep track of all third-parties it crossed in the distribution chain. In this case, everyone can distribute the content and define a new license without any restriction. In these licenses, the content owner and distributors specify end-user's rights. Using the content traceability, the DRM controller can gather all licenses involved in the distribution chain and evaluate them. In order to be rendered, a right must be allowed by both the content owner and all distributors involved in the distribution chain.

1 Introduction

Digital Rights Management frameworks (DRM) [12, 1] aim at protecting and controlling information contents which are no longer on a server side but instead distributed on the client side. DRM frameworks provide security mechanisms in order to protect the confidentiality and the integrity of digital contents. Using a license, the content owner specifies which rights end-users can have on the protected content. The license is written according to a specific Rights Expression Languages (REL) [6, 11]. A dedicated rendering application is in charge of evaluating the corresponding license and then render the requested right to the end-user. The set of rights supported by the rendering application is defined in the Rights Data Dictionary (RDD) of the corresponding DRM framework.

Please use the following format when citing this chapter:

Sans, T., Cuppens, F., and Cuppens-Boulahia, N., 2007, in IFIP International Federation for Information Processing, Volume 232, New Approaches for Security, Privacy and Trust in Complex Environments, eds. Venter, H., Eloff, M., Labuschagne, L., Eloff, J., von Solms, R., (Boston: Springer), pp. 349–360.

Fig. 1. Super-Distribution scenario.

For each right corresponds a rendering action enforced in the rendering application. Only the given rendering application [5, 7] is able to deal with the content protection and applies a rendering action on this content. The rendering application is executed on the user's device, so it must be trustworthy and tamper resistant [4, 9].

How to ensure that anyone cannot modify or issue a valid license for a given content? At first glance, only the content owner can issue a valid license for a given content. Only the license provided by the content owner can be evaluated by the rendering application. In existing DRM frameworks, cryptographic and hashing mechanisms are used to guarantee this ownership principle. But, some distribution models go further. Rather than managing rights with end-users, the content owner might want to delegate this task to a third-party called a distributor. The latter would be in charge to manage rights with end-users or even delegate this task again to another distributor. In literature, we talk about multi-tiers distribution or super-distribution scenario [12]. Let us consider a generic example of super-distribution scenario as showed in figure 1. Art'East wants to protect and distribute its own multimedia content on the Web. Rather than dealing with distribution issue, Art'East wants to entrust its content to StoreOnline care. This is a typical B2B business model where a content owner allows a third-party to distribute multimedia contents. The owner does not allow the distributor to read the content but rather allows him or her to manage rights with end-users. This is a typical B2C business model where a distributor allows an end-user to use the content. We can even go further enabling C2C business model where a user can allow another user to use the content.

Obviously, this distribution cannot be done without any control from each party of the distribution chain. In particular, the owner wants to control how the content is used even if he or she does not directly issue the license to end-users. Art'East, as a content provider, must be able to control the rights that can be rendered to Alice even if the multimedia content is under control of the distributor StoreOnline. Similarly, a distributor might allow end-users to distribute the content. In addition to issuing the play right, Alice might be able to give the play right to her friend Brice and then become a distributor. Again, StoreOnline and Art'East must be able to control what Brice can really do with the content. In our example, Art'East wants that an end-user can play the content. The distributor StoreOnline is supposed to issue the right to play

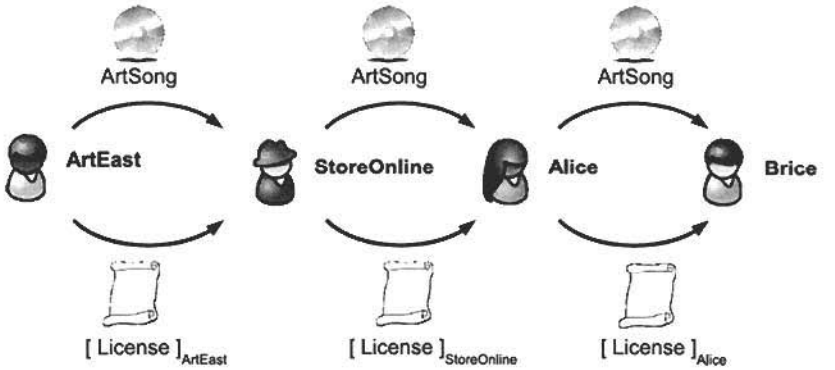

Fig. 2. The issuing approach.

only. Art'East does not restrict who can get this right and leaves StoreOnline free to define it. StoreOnline allows its members to read Art'East multimedia content. Alice, as a StoreOnline member, can read contents from Art'East. A storeOnline member can also allow a friend to play the content but the latter cannot allow someone else to play it. Here, Alice can allow Brice to read a specific content but Brice cannot allow someone else to read it.

Let us focus on how existing DRM frameworks enforce super-distribution mechanisms. We are considering here two open standards: OMA-DRM [10] and MPEG-21 [8]. The OMA-DRM specification talks about a *super-distribution mechanism*. In OMA-DRM, a third-party can redistribute a content to an end-user or to another distributor but cannot issue a new license on it. In consequence the distributor cannot restrict rights initially defined by the content owner license. The super-distribution mechanism provided in OMA-DRM does not satisfy the definition of super-distribution given above. Contrary to OMA-DRM, MPEG-REL enables a super-distribution mechanism as defined here: a third-party distributor can distribute a content and define a new license on it. In MPEG-REL, the content owner can restrict the license issued by the distributor. The content owner defines a license pattern and the license finally issued by the distributor must match this license pattern. Only this latter license is going to be evaluated by the rendering application in order to decide if a right can be exercised or not. We call this mechanism *the issuing approach*. The figure 2 shows how the issuing approach is applied to the super-distribution scenario given above. The content owner Art'East issues a license specifying that the distributor StoreOnline can "issue" a license according to a given pattern. This license pattern specifies that anyone can play the content ArtSong. Using that license, the distributor can now issue a license specifying that Alice can play the content. To be valid, this license must match the license pattern defined

by the content owner license. The problem is more complex for the C2C business model. In that case, Art'East should issue a license specifying that the distributor can issue a license allowing someone else to issue the right to play.

The issuing approach aim at restricting the license issued by the different distributors involved in the distribution chain. The license finally issued is a license allowed by the content owner and by all the distributors involved in the distribution chain. In this paper, we provide a new approach, called the Onion Policy Administration approach (OPA). Rather than restricting licenses issued by the different distributors, OPA aims at controlling which rights are finally rendered to end-users. The main idea of OPA is to have a traceability of the content distribution. The content must keep track of all third-parties it crossed in the distribution chain. In this case, everyone can distribute the content and define a new license without any restriction. In these licenses, the content owner and distributors specify end-user's rights. Using the content traceability, the DRM controller can gather all licenses involved in the distribution chain and evaluate them. In order to be rendered, a right must be allowed by both the content owner and all distributors involved in the distribution chain.

In the following section, we better explain the Onion Policy Administration approach and we show how OPA enables the super-distribution scenario given previously. In section 3, we formalize the content traceability mechanism provided by OPA. We also provide a sketch of Rights Expression Language to express OPA licenses and its corresponding license interpretation algorithm. In section 4, we deal with implementation issues.

2 OPA: Onion Policy Administration model

Contrary to the issuing approach, in OPA we aim at controlling rights finally rendered to end-users rather than constraining the license issued by the different distributors. Every distributor involved in the distribution chain of a given content can issue a valid license without any restriction. The rendering application must evaluate all licenses provided by both the content owner and all distributors involved in the distribution chain. First, we provide a traceability mechanism in order to identify who is the content owner and who are the different authorized distributors involved in the distribution chain. Secondly, we provide a sketch of Rights Expression Language and its corresponding license interpretation mechanism in order to control the end-users rights. Both content owner and distributors can specify which rights can be finally rendered to end-users according to a specific sub-distribution chain.

2.1 The content traceability

Basically, only the content owner or an authorized distributors can issue a valid license for a given content. We call this principle *the ownership principle.* MPEG-21 and OMA-DRM [8, 10] have the same approach to guarantee the

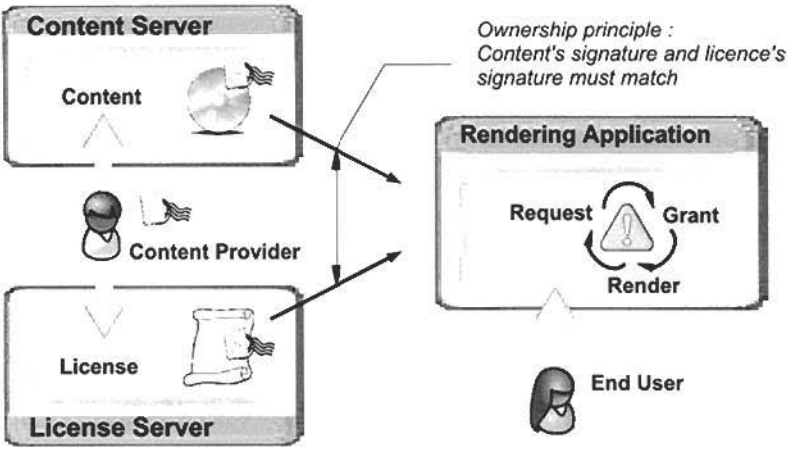

Fig. 3. The ownership principle in DRM.

ownership principle. The content and the license are tight using hashing and cryptographic mechanisms. For example in Windows Media DRM framework (Microsoft Media Player)[1] [3, 2], the content is ciphered using a symmetric key. This key is distributed to the user through the license. Obviously, this key cannot be distributed in clear in the license. The license, or the part of the license with the key, is ciphered using a session key between the license server and the rendering application. Thus, only the license issued by the content provider can be used to decipher the content. In super-distribution scenario, a distributor can issue a new license to end-users or to another distributors. In order to be valid, this new license must match a given license pattern define in a license previously issued by the content owner or the previous distributor in the distribution chain. If this license match this pattern, the cryptographic key is then transferred to the new license. This license is told to be valid because it contains the key to decipher the content.

The way to enforce the ownership principle in OPA is different. The content owner must both sign the content and the license. When the rendering application evaluates a license for a given content, these two identities must match as shown in 3. So how distributors, in super-distribution use-cases, can then issue valid licenses? Obviously, the authorized distributor must appear as the owner of the content, so the solution is to change the owner of the content, i.e. modify the signature of the content owner. To do that, the content must be repackaged in order to change the owner signature. Even if the original owner allows this repackaging operation, the solution is not acceptable as it gives the distributor a mean to issue any rights on the content. In such a configuration, the original owner totally loses control on the content. In OPA, both the owner

[1] Now named Microsoft PlaysForSure. This DRM system is told to enforce the MPEG-21 standard.

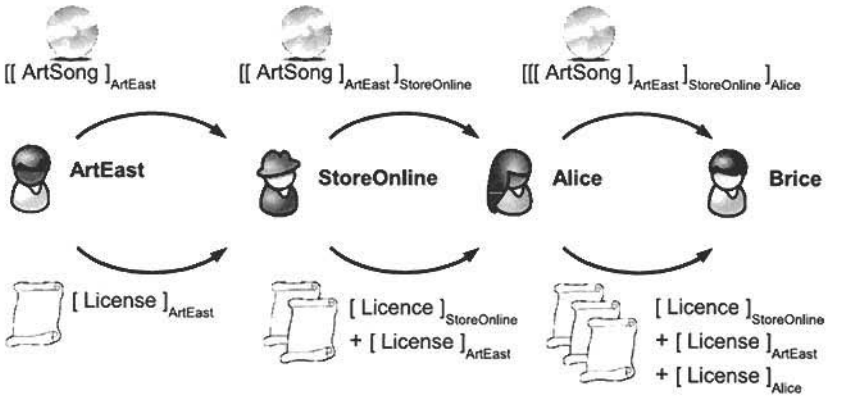

Fig. 4. Content traceability with OPA.

and distributors must be able to specify how users, and distributors if any, can use the content. If one of them does not agree, then the rendering action cannot be performed. Thus, it does not matter which rights are issued by each party but to be valid, both the owner and the distributor must allow it.

In OPA, a content provider is an entity able to specify licenses on the content, i.e. an entity allowed to add the signature to the content. Both the content owner and distributors are content providers. As showed in the figure 4, a content can be signed by several content providers depending on where is the content in the super-distribution chain. When the owner wants to allow a distributor to issue some rights, it creates a license allowing the distributor to "wrap" the content. The wrap right enables a distributor to become a content provider of the content. Once the distributor is one of the content providers, he or she can issue licenses on it.

Using this traceability mechanism, the rendering application is now able to extract the complete distribution chain of a given content. The rendering application must then evaluate all the licenses issued by the different content providers involved in this chain. In the following section, we show how content providers can control rights finally rendered to end-users and how the rendering application can decide if a right can be rendered or not.

2.2 Controlling rights with OPA

In OPA, all licenses issued by the different content providers involved in the distribution chain are evaluated. All of them must allow a right to be exercised in order to be rendered to the end-user. In our example, Art'East, as a content owner, provides a content with its signature. Art'East also delivers a license to

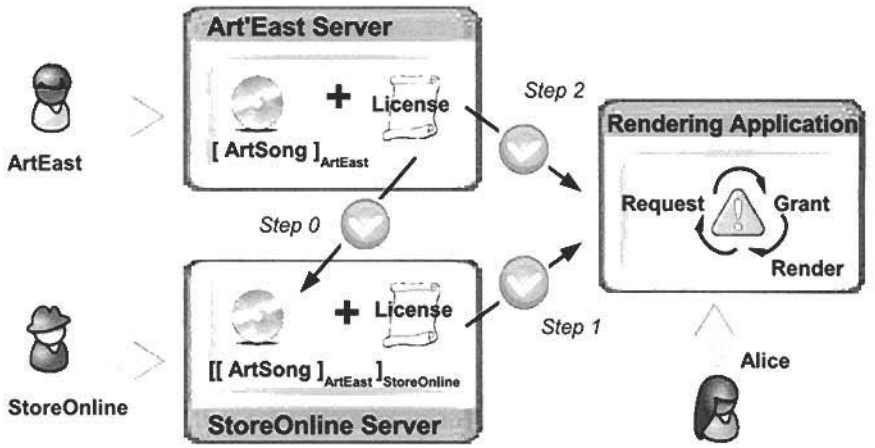

Fig. 5. Authorization mechanism with OPA.

StoreOnline allowing it to wrap the content (step 0) as showed in the figure 5. Using this license, StoreOnline adds its signature to the content and is now able to issue valid license to Alice without any restriction. Let us consider that StoreOnline first allows her to play the content. When Alice wants to read the content, she has first to get the StoreOnline license as StoreOnline is the last content provider (step 1). The license allows her to play it, but with the onion approach this is not sufficient. All content providers, involved in the distribution chain, must allow her to play it. So, she has also to get the Art'East License (step 2). For Art'East, only the play right is allowed but in practice, it does not want to specify who exactly use its content. So, why the owner does not simply specify in its license that anyone can play the content? Because this will introduce a security breach: anyone who got a content only signed by Art'East can play it. Here, Art'East wants to allow everyone to play the content only if it has been distributed by a trusted distributor, the one who got the wrap right on the content. In the onion approach, the owner can specify that everyone can play the content provided by StoreOnline. For this purpose, we introduce a parameter "from" specifying the authorized content provider. In our example, **from** *StoreOnline* is tied up to the target content of the grant.

Now let us consider that StoreOnline issues a license to Alice allowing her to modify the content. Alice can never modify the content because, even though Alice is allowed by StoreOnline, the DRM controller also checks the Art'East license. As there is no grant allowing her to modify the content when the content is provided by StoreOnline, she will fail to do any modification.

If we go further in the example, the onion approach is adequate to enable the C2C business model. When Alice wants to allow Brice to play the content, she first adds her signature to the content. She is allowed to do that because both StoreOnline and Art'East allowed her to wrap the content. StoreOnline

directly allowed her to wrap the content. Art'East allowed Alice to do this wrapping because the content is distributed **from** *StoreOnline*. When Brice tries to play the content, the DRM controller first checks the license issued by the last content provider namely Alice. With this license, Alice allows Brice to play the content. Then, the DRM controller checks licenses issued by other content providers of the distribution chain. The second one is the StoreOnline license. With this license, Brice is allowed to play the content because it has been provided **from** *StoreOnline-Member*. Finally, Art'East allowed Brice to play it because the content has been provided both by StoreOnline and by someone else. In that case, Art'East allows a C2C distribution but there is no requirement on the identity.

3 The underlying model

This section formalizes the Onion Policy Administration model. We do not attempt to specify a complete DRM framework, neither to define a new Rights Expression Language covering all the expressiveness of existing ones. We rather aim at specifying the main concepts needed by a DRM framework and its corresponding REL both enabling OPA.

3.1 The content packaging

A content is a digital document wrapped in a secure container and digitally signed by the content owner as required by OPA. When a provider is allowed to redistribute a content ("wrap" right), the provider signature is appended to the previous content. Providers identities involved in the distribution chain can be seen as different onion layers wrapping the original document. The content $[ArtSong]_{Art'East}$ means that Art'East is the owner of the digital document Art'Song. This content traceability mechanism is formalized as follows:

$$\begin{aligned} \text{TYPE}\ & Identity, Right, Document \text{ are nominal types} \\ & Content \triangleq [\ Document\ ]_{Provider} \mid [\ Content\ ]_{Provider} \\ & Provider \triangleq Identity \end{aligned}$$

3.2 The Rights Expression Language

A license is a set of grants where a grant is a triple composed of an identity, a right and a digital document. As required in OPA, the license must contain the provider identity. This Rights Expression Language is defined as follows:

$$\begin{aligned} \text{TYPE}\ & Grant \triangleq Identity \times Right \times Content \\ & License \triangleq [\ \{Grant\}\ ]_{Provider} \\ & Provider \triangleq Identity \end{aligned}$$

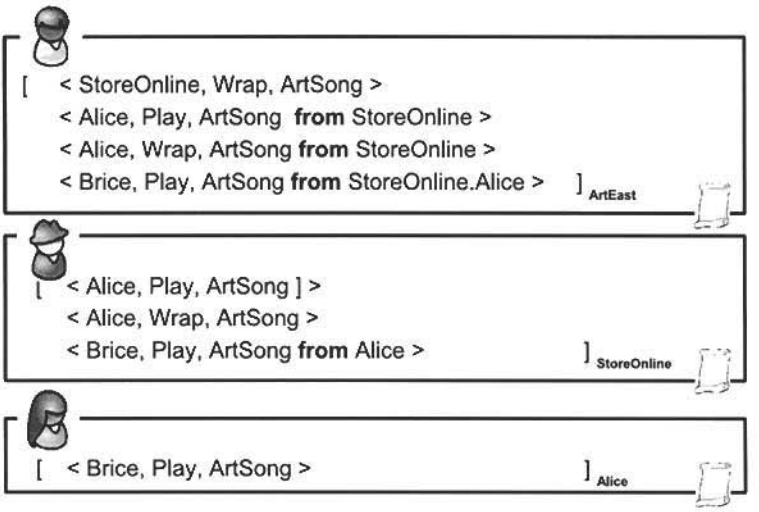

Fig. 6. Super-Distribution scenario with OPA.

Only authorized providers are allowed to append their digital signatures to the content. We define the right $Wrap$[2] enabling someone to become a content provider. If someone is allowed to *wrap* a given content then he or she is allowed to add his or her signature to the content. In our example, Art'East wants to allow StoreOnline to distribute the content. Art'East issues a license $[< StoreOnline, Wrap, ArtSong >]_{Art'East}$ allowing StoreOnline to wrap ArtSong. When StoreOnline uses this right, the rendering application creates the new content $[[ArtSong]_{Art'East}]_{StoreOnline}$. StoreOnline can then issue the valid license $[< Alice, Play, ArtSong >]_{StoreOnline}$ to Alice granting her the right to play. According to the ownership principle, the license is valid as StoreOnline is now one of the ArtSong's providers. But the license issued by StoreOnline is not enough to allow Alice to play the content. In the onion approach Art'East is still able to control which right Alice can get as an end-user of the distribution chain. Art'East must allow Alice to play the content only if this content has been distributed by StoreOnline. To do that, Art'East append a new grant $< Alice, Play, ArtSongs$ **from** $StoreOnline >$.

Let us now focus on how to enforce the C2C business model using the onion policy administration. In that business model, Alice can distribute the content to Brice. This latter can only play it but cannot distribute it. First of all, Alice must be able to wrap the ArtSong content provided by Art'East and StoreOnline. It means that both of them must allow Alice to *wrap* the content $[[ArtSong]_{Art'East}]_{StoreOnline}$. Art'East must issue the license $[< Alice, wrap, ArtSong$ **from** $StoreOnline >]_{Art'East}$ and StoreOnline must issue

[2] The Wrap right is one of the rights defined by the Right Data Dictionary of our DRM framework.

the license $[< Alice, wrap, ArtSong >]_{StoreOnline}$. Using those licenses, Alice can wrap the content and then distribute $[[[ArtSong]_{Art'East}]_{StoreOnline}]_{Alice}$ to Brice. If Brice wants to play this content then Alice, StoreOnline and Art'East must allow him to play it. Each of them must include a grant in their license allowing Brice to play the content according to who distributes it. The figure 6 shows all licenses issued by different parties enforcing all business models discussed previously in the example.

3.3 The license interpretation mechanism

In order to formalize the license interpretation mechanism, we first define the *isPermitted* predicate which is true, for a given set of license Γ, if there is a grant matching the given identity, the given right and the given document. The isPermitted predicate is formalized as follows :

$$\text{PREDICATE } isPermitted \triangleq Identity \times Right \times Content \times listOf(Provider) \rightarrow Boolean$$

$$\text{AXIOM } \Gamma \cup [\langle i, r, d \textbf{ from } plist\rangle]_{p_0} \vdash isPermitted(i, r, [d]_{p_0}, plist)$$

$$\Gamma \vdash isPermitted(i, r, [[[d]_{p_0}]\ldots]_{p_n}, (p_{n+1}|plist)) \rightarrow \Gamma \cup [\langle i, r, d \textbf{ from } plist\rangle]_{p_{n+1}} \vdash isPermitted(i, r, [[[[d]_{p_0}]\ldots]_{p_n}]_{p_{n+1}}, plist)$$

Then, we define a set of authorization predicates. These predicates are used to decide if a requested right (*request* predicate) can be allowed (*allow* predicate) or not (*deny* predicate). Such a request comes from the environment of the information system Σ. With OPA, such a request can be allowed if and only if every license in Γ, from the different content providers involved in the distribution chain, allows the request. If one content provider does not allow the request, the right cannot be rendered. These authorization predicates (*allow* and *deny*) are defined as follows:

TYPE λ is the empty list

$$\begin{aligned}\text{PREDICATE } request &\triangleq Identity \times Right \times Content \rightarrow Boolean\\ allow &\triangleq Identity \times Right \times Document \rightarrow Boolean\\ deny &\triangleq Identity \times Right \times Document \rightarrow Boolean\end{aligned}$$

$$\begin{aligned}\text{AXIOM } &\Sigma \vdash request(i, r, [[[d]_{p_0}]\ldots]_{p_n})\\ &\wedge \Gamma \vdash isPermitted(i, r, [[[d]_{p_0}]\ldots]_{p_n}, \lambda)\\ &\rightarrow \Sigma, \Gamma \vdash allow(i, r, d)\end{aligned}$$

$$\begin{aligned}&\Sigma \vdash request(i, r, [[[d]_{p_0}]\ldots]_{p_n})\\ &\wedge \Gamma \vdash \neg isPermitted(i, r, [[[d]_{p_0}]\ldots]_{p_n}, \lambda)\\ &\rightarrow \Sigma, \Gamma \vdash deny(i, r, d)\end{aligned}$$

4 Implementation

We developed a prototype as a proof of concept of a DRM framework enabling OPA. In this framework, we developed a content packager to protect a digital document and sign it. This packager enforces the wrapping mechanism defined in OPA in order to enable the content traceability. XML envelopes are used to wrap the content and XML digital signatures [13] are used to sign XML envelopes. Secondly, we defined our own XML-based REL enabling OPA. XML Signatures are also used to sign the licenses. Finally, we developed the rendering application in charge of interacting with a physical user (Graphical User Interface) or an external application (service). Through this interaction layer, a request can be made and the corresponding rendering can be delivered. The DRM controller embedded in the rendering application is able to interpret the licenses and give a decision if the right can be granted or not. The DRM controller algorithm is compliant with the OPA decision mechanism as shown in figure 5.

5 Conclusion

In super-distribution scenario, the content owner does not directly manage end-user's rights but rather delegate this task to a third-party: a distributor. Existing DRM approach, enforcing super-distribution, are based on *the issuing approach* where a content owner or a distributor restricts the license issued by the next distributor in the distribution chain. This paper provides a new approach called OPA (Onion Policy Administration) to manage rights in super-distribution models. OPA aims at controlling which rights are finally rendered to end-users rather than restricting the licenses issued by the different distributors in the distribution chain. With OPA, distributors are free to issue a license without any restriction. In these license, the content owner and the distributors specify which rights can be rendered to the end-users according to how this content was distributed, i.e. the content owner and the distributors specify that a given content must have been distributed according to a specific sub-distribution chain. The rendering application, in charge of deciding if a right can be rendered or not, must evaluate all licenses involved in the distribution chain of a given content. All of them must allow the requested right in order to be rendered.

Compared with the issuing approach, we assume that OPA has two main advantages. First, OPA provides a content traceability mechanism in order to identify the distribution chain of a given content. Each time a content is redistributed by a third party, the distributor signature is stored in the content. This traceability mechanism can be required by critical DRM applications. Indeed, we believe that DRM techniques can be used in critical information systems such as medical, administrative or military applications and not only in commercial application. The second main advantage is that OPA simplifies rights

management in super-distribution. With the issuing approach, the distribution chain is defining using overlapping grants, so there are as many overlapping grants as there are distributors in the distribution chain. The more third parties there are in the chain, the more difficult is to write license. In the onion approach, there is only two grants : one specifying if someone can be a content provider (using the *Wrap* right) and another one specifying what the end-user can do with the content according to a valid sub-distribution chain. Thus, OPA is more adequate for DRM application involving many third-parties in content super-distribution.

Acknowledgement: This work was supported by funding from the French ministry for research under "ACI Sécurité Informatique: CASC Project"

References

1. Eberhard Becker, Willems Buhse, Dirk Gnnewig, and Niels Rump, editors. *Digital Rights Management: Technological, Economic, Legal and Political Aspects.* Lecture Notes in Computer Science - Springer Berlin / Heidelberg, 2003.
2. Microsoft Corporation. Using windows media encoder to protect content. Technical report, March 2003.
3. Microsoft Corporation. Architecture of windows media rights manager. Technical report, May 2004.
4. John S. Erickson. Fair Use, DRM and Trusted Computing. *Communication of the ACM*, 46(4), April 2003.
5. Richard Gooch. *Requirements for DRM Systems Introduction - The Requirement for DRM*, volume 2770. January 2003.
6. Susanne Guth. *Rights Expression Languages*, volume 2770. January 2003.
7. Susanne Guth. *A Sample DRM System*, volume 2770. January 2003.
8. International Organization for Standardization (ISO). *ISO/IEC 21000:2004 Information technology – Multimedia framework (MPEG-21)* , 2004.
9. Dirk Kuhlmann and Robert A. Gehring. *Trusted Platforms, DRM, and Beyond*, volume 2770. January 2003.
10. Open Mobile Alliance (OMA). *OMA Digital Rights Management V2.0*, 2006. http://www.openmobilealliance.org/release_program/drm_v2_0.html.
11. David Parott. Requirements for a Rights Data Dictonary and Rights Expression Language. Technical report, Reuters, June 2001.
12. Bill Rosenblatt, Bill Trippe, and Stephen Mooney. *Digital Rights Management: Business and Technology.* Wiley, Decembre 2001.
13. World Wide Web Consortium (W3C). *XML-Signature Syntax and Processing*, 2002. www.w3.org/TR/xmldsig-core/.