A Fine-grained Digital Rights Transfer Policy and Trusted Distribution and Enforcement

Zhiyong Zhang[1, 2], Qingqi Pei[1], Jianfeng Ma[1], Lin Yang[3] and Kefeng Fan[4]
[1]*Key Lab of Computer Network & Information Security (Ministry of Education), Xidian University, Xi'an, Shannxi 710071, P.R. of China*
[2]*Electron. Inf. Eng. Coll., Henan Univ. of Sci. & Technol., Luoyang, Henan 471003, China*
[3]*The Research Institute, China Electronic Equipment & Systems Engineering Corporation, Beijing 100039, China*
[4]*China Electronics Standardization Institute, Beijing 100007, China*
{zhangzy,qqpei,jfma}@mail.xidian.edu.cn

Abstract

Existing Digital Rights Management (abbr. DRM) approaches lack a flexible fine-grained digital rights transfer policy, also could not guarantee the trustworthy distribution and enforcement of the transferable rights policy by using DRM Agent. First, descriptions of extensible ODRL-based rights transfer policy were presented to complete OMA REL. Then, we introduced a remote attestation mechanism among entities, such as RI and DRM Agent, based on trusted computing platform in order to implement trusted distribution and enforcement of the policy, also mainly proposed trustworthy distribution protocols of the rights object and transfer , as well as a trusted policy enforcement protocol. Final, the approach is made comparison with existing DRM schemes as to several functionality aspects on transferable rights granularity, restricted sharing, temporal limitation, trust of DRM Agent. A conclusion is drawn that the proposed scheme is not restricted within local domain environment, and accomplishes fine-grained rights transfer and contents sharing between users without direct participations of Rights Issuer or Local Domain Manager.

1. Introduction

In decade, illegal copy, free distribution and unauthorized usage of copyrighted digital contents have already been a common phenomenon. Digital Rights Management came into use only in the mid of the 1990s, with a goal to solve the above mentioned problems. Nowadays DRM system aims at effectively implementing the protection and management of digital contents in a whole life cycle from production, distribution, transfers to storage and usage.

In general, a direct binding mode of content-license-device (consumer) is adopted when Rights Issuer (abbr. RI) authorizes for purchaser, but strictly controls the usage of copyrighted contents. In order to make it convenient to share contents among different devices and authenticated users, Digital Video Broadcasting Alliance firstly proposed an Authorized Domain concept [1], which has been adopted in several representative specifications of OMA DRM [2]. The unified managements of RI on the Authorized Domain include the establishment and revocation of the domain, the join and quit of user devices, as well as contents and licenses could be shared among electronics devices in the domain. It is noted that these managing tasks burthen RI, also RI becomes a bottle-neck of the sharing domain. An authorized domain architecture and related security protocols were presented in [3], but the scheme do not support Rights Object (abbr. RO) transfer and contents sharing. In addition, as some approaches available are not involved with fine-grained digital rights transfer, an improvement based on home network DRM was proposed in [4], where Local Domain Manager (abbr. LDM) was introduced in order to substitute RI's functions of license distribution for joined devices in home domain, further rights transfer/delegation was realized by using Delegated RO (abbr. DRO) and Proxy Certificate. However, the approach has a shortcoming that the introduction of LDM increases system cost and becomes a new attacked object; also, the trust relation between RI (LDM) and DRM Agent, which is a essential component having key capability of controlling contents' legal usage in a general DRM

978-0-7695-3508-1/08 $25.00 © 2008 IEEE
DOI 10.1109/CIS.2008.137

system, was mainly dependent on an authentication mechanism by the component's certification issued by a trusted third party, so the scheme does not ensure that DRM Agent has integrity and is not ever tampered with and captured by a malicious adversary. Whether the RO (DRO) embodying digital rights' usage policy could be trustworthily enforced would be still an open issue.

Generally speaking, Rights Expression Language (abbr. REL) is suitable for specifying and depicting the legitimate rights, usage conditions and constraint rules of purchased digital contents [5]. So far, ODRL[6], XrML[7] and MPEG-21 REL[8] have already specified several permissions with respect to digital rights transfer/delegation, but they do not belong to a fined-grained rights transfer policy, but a coarse-grained one. Moreover, as OMA REL[9] based on ODRL do not depict the relative semantics of rights transfer, large numbers of applications adopting OMA DRM specifications do not implement the effective and controlled fine-grained rights transfer and delegation. The main contributions of the paper are twofold: one is to extend the transferable rights characteristics of OMA REL based on ODRL, make it easy to express a fine-grained rights transfer policy; the other is to realize the trusted distribution and execution of the policy based on Remote Attestation (abbr. RA) mechanism in trusted computing[10].

2. Digital Rights Transfer Policy for DRM

Digital rights transfer policy was presented in extensible ODRL language, as well as RO and TRO were given in the section, respectively.

2.1. Fundamental Characteristics of Rights Transfer

Existing DRM rights transfer policies aim mainly at the whole-permission transfer/delegation for digital contents usage, for instance, Transfer and Loan in XrML, Sell, Lend, Give and Lease in ODRL. But, these above permissions lack of concrete definitions as to transferable rights' granularity, time limitation, depth and cardinality constraints. So, these RELs do not meet an important requirement of fine-grained policy expression, with decreasing the flexibility of contents sharing between purchaser and others. In essence, rights transfer policy should include the following features:

(1) Granularity: denotes the transferable usage permission(s) of digital contents as a whole or part. It is clear that the fine-grained policy is much more advantageous to flexible contents sharing than coarse-grained one.

(2) Constraints: specify the temporal limitation, usage cycle, usage number and devices available of transferred permissions.

(3) Transfer Depth: depicts the cascade number of further transferred permission.

(4) Transfer Cardinality: defines the maximum of user acquiring transferred rights.

2.2. ODRL-based Transfer Policy Depicting of OMA REL

In term of the fundamental characteristics and extensible capability of ODRL, we mainly defined relative semantics in OMA REL as follows: basicPermission denotes such basic privileges as display, play and execute, and permissionConstraint depicts pre-conditions of basicPermission's usage. The transferable permissions are from the basicPermission set, so transferPermission set is a subset of basic privileges one. Besides, transferPermssion has some fundamental constraints denoted by transferConstraint, which include transferTimelimit transferDepth, and transferCardinality. Further, based on these defined tags, we presented a rights transfer policy with light-weight semantics as Fig. 1 and 2, which could be suitable for a general computer network and mobile network environments, and be easier to execute in mobile terminal device with finite computing and storage capabilities. A RO with fine-grained transferable rights is depicted in Fig 1, and the RO executed by DRM Agent would be distributed by RI to consumer. Fig 2 expresses a Transferable Rights Object (abbr. TRO) that includes transferable basic privileges available for a rights sharer. The generation of TRO must meet condtions of transferConstraints depicted in RO, and TRO are also enforced by DRM Agent.

```
<o-ex:rights
  xmlns:delegate="http:/www.mispb.com/rbac/transfer-dd">
  <o-ex:context>
    <o-dd:uid>RightsObjectID</o-dd:uid>
  </o-ex:context>
  <o-ex:agreement>
   <o-ex:asset>
     <o-ex:context>
       <o-dd:uid>ContentID</o-dd:uid>
     </o-ex:context>
   </o-ex:asset>
  <o-ex:permission>
   <o-dd:basicPermission>
     <o-ex:constraint>
       count_constraint OR datatime_constraint OR…
     </o-ex:constraint>
  </o-dd:basicPermission>
  <transfer:transferPermission>
     <o-dd:basicPermission>
       <o-ex:permissionConstraint>
         count_constraint OR datatime_constraint OR…
       </o-ex:permissionConstraint>
       <o-ex:transferConstraint>
         transferDepth OR transferCardinality OR
         transferTimelimit
       </o-ex:transferConstraint>
     </o-dd:basicPermission/>
  </transfer:transferPermission>
 </o-ex:permission>
</o-ex:agreement>
</o-ex:rights>
```

Figure1. Extensible ODRL–based RO description

```
<o-ex:rights
    xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
    xmlns:o-dd="http://odrl.net/1.1/ODRL-DD"
    xmlns:delegate="http:/www.mispb.com/rbac/transfer-dd">
<o-ex:context>
    <o-dd:uid>TranferableRightsObjectID</o-dd:uid>
</o-ex:context>
<o-ex:agreement>
   <o-ex:asset>
     <o-ex:context>
        <o-dd:uid>ContentID</o-dd:uid>
     </o-ex:context>
  </o-ex:asset>
<o-ex:permission>
  <o-dd:basicPermission_1>
     <o-ex:constraint>
       <o-dd:count>count_constraint_1</o-dd:count>
       <o-dd:datetime>datatime_constraint _1</o-dd:datetime>
          …
     </o-ex:constraint>
  </o-dd:basicPermission_1>
  <o-dd:basicPermission_2>
       basicPermisson_2 constraint
  </o-dd:basicPermission_2>
  <transfer:transferPermission>
       transferPermisson_1
  </transfer:transferPermission>
</o-ex:permission>
</o-ex:agreement>
</o-ex:rights>
```

Figure2. Extensible ODRL–based TRO description

3. Trusted Distribution and Enforcement of Rights Transfer Policy

Based on the digital rights transfer policy presentation in the above section, three protocols involved with the trusted distribution and enforcement of the policy were proposed.

3.1. DRM System Architecture with Rights Transfer based on Trusted Computing Technology

OMA DRM is a representative reference specification of digital contents protection and rights management. In the specification, a corresponding license of purchased contents is expressed by using OMA REL, that is to say that RO is generated, and then RI distributes the RO to DRM Agent located on user terminal device based on the ROAP protocol. Here RI needs to check the Agent's certificate via PKI system before the distribution of RO. When it is validated to be legal or belong to a trusted vendor, the Agent is trusted to interpret and execute the permissions included in RO, thus the trust relation between both being established. But, it is clearly seen that the trust relation is in essence entirely dependent on a static certificate, so the mechanism could not guarantee the dynamic run-time integrity.

Of key technologies in trusted computing, Remote Attestation is responsible for an attestation of platform integrity based on Trusted Platform Module and Trusted Software Stack located on upper layer, and validates whether the terminal is ever tampered with and attacked by malicious codes, as well as is consistent with pre-defined security policies or not, and then trustworthily reports the integrity status to the outside entities so that the remote entities could make effective decisions on network admission, resource distribution and usage. According to OMA DRM architecture and related protocol specifications [11], we introduced remote attestation mechanism for DRM Agent and terminal platform in the procedure of digital rights object distribution and transfer, as is shown in Fig 3.

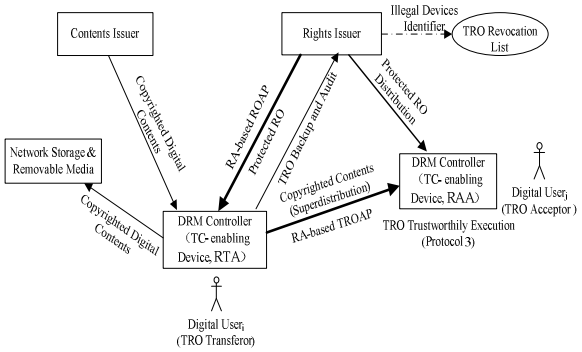

Figure3. DRM architecture with digital rights transfer based on remote attestation and trusted computing environment

With regard to the improved ROAP in the paper, RI and Agent together accomplish 4-pass register and RA-based RO distribution, and implement trusted distribution of RO with rights transfer extension. When it has received RO, Rights Transferor Agent (abbr. RTA) interprets extensible ODRL-based RO in Fig 1, and generates a TRO from several transferable permissions in RO by using TRO description in Fig 2. Meanwhile, TRO would be copied and returned to RI, with a goal to audit and track transferable rights. In addition, as to a device illegally generating, transferring and usage of TRO, it would be issued by RI in TRO Revocation List, so the device would not acquire RO, TRO and digital contents once again. In term of the following Protocol 2 in Section 3.2, RTA makes decision on the distribution of TRO after validation of the integrity of Rights Acceptor Agent (abbr. RAA); then RAA accesses to digital contents by the acquired TRO. It should be pointed out that user terminal platform needs to meet trusted computing related specification, for example TCG specifications [9]. The distribution of digital contents between CI and RTA, together with the superdistribution between RTA and RAA could refer to OMA DRM, also user authentication and key management are also out of scope of the paper.

3.2. RO Trusted Distribution Protocol

On the basis of 2-pass ROAP protocol, a remote attestation on DRM Agent and terminal platform is introduced, and the interactions of concrete messages are shown in Fig 4. Assume that user terminal has already acquired AIK and other certificates from Privacy CA.

Protocol 1 (RA-based RO Trusted Distribution) after accomplishing 4-pass register protocol, RI firstly validates the integrity of DRM Agent and terminal platform. If they are compliant to integrity references, RI further distributes protected RO with rights transfer extension to RTA.

(0) Preliminary procedure: RI and RTA implement 4-pass register protocol all together, realizing bidirectional authentication based on PKI; then RI Context is sent to RTA' s platform, meanwhile a secure channel is successfully established for message interactions between both.

(1) DRM RTA sends $Message_{RTA}$, which includes device identifier, RI identifier, RO request time, RO request message and a $Nonce_{RTA}$ produced by terminal device, and its RSA signature by SK_{RTA} to RI;

(2) After the above messages are accepted by RI, a 160-bit $Nonce_{RI}$ that is consistent with the bit number of PCR (Platform Configure Register) is produced, and a RA challenge for RTA is sponsored. Here $Nonce_{RI}$ is used to prevent a replay attack from adversary.

(3) In the bootstrap procedure of RTA platform, the integrity measurement merits by Hash operation of OS pre-load and post-load, together with the hash value of RTA are stored in several PCRs, as accomplishes the local measurement of components; in general hash operation adopts SHA-1 or MD5 algorithm. RTA returns the response messages of RA challenge to RI, which include the RSA signature of PCR values along with $Nonce_{RI}$ by $SK_{RTA\cdot TPM.AIK}$, as well as RTA's platform certificate cert (RTA.TPM.AIK).

(4) RI receives the response messages and realizes integrity attestation: firstly, RI validates not only the cert (RTA.TPM.AIK) issued by CA, but also the signature by using public key of the certificate $PK_{RTA.TPM.AIK}$, so that the origin of PCRs is trusted, meanwhile PCRs and $Nonce_{RI}$ are also acquired; secondly, the integrity of RTA's platform is checked through a comparison between PCRs and integrity references. It is noted that the provision, issue, storage and query of these references could refer to TCG-IMM model; finally, when the attestation is accomplished, RI produces $Message_{RI}$ that includes the status of protocol execution (e. g. success or fail), device identifier, RI identifier, $Nonce_{RTA}$ and protected RO; RI sends $Message_{RI}$ and its signature by using SK_{RI} to RTA via a pre-established secure channel.

(5) RTA validates the signature of $Message_{RI}$ after receiving the above messages, and acquires RO and an encryption key of digital contents, which is called CEK.

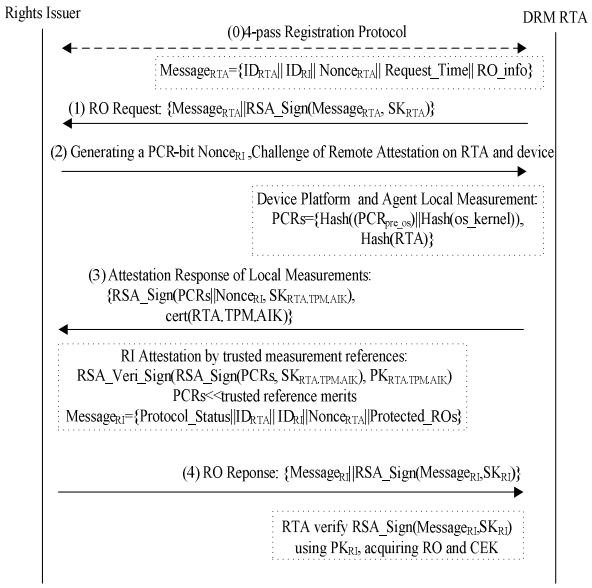

Figure4. RA-based RO trusted distribution protocol

3.3. TRO Trusted Distribution and Enforcement Protocols

RTA shares digital contents with RAA based on superdistribution. When it needs to share contents, RAA must acquire TRO transferred by RTA in advance. Protocol 2, which is shown in Fig 5, aims at resolving the issue. Supposed that RAA firstly sends the TRO request after RAA has received contents.

Protocol 2 (RA-based TRO Trusted Distribution) the TRO distribution between DRM Agents is on basis of bidirectional authentication each other by PKI, with ensuring the validity of RTA and RAA; then, RTA validates the integrity of RAA platform by using remote attestation mechanism, further decides whether to send TRO or not. In term of the shared permissions depicted in TRO, RAA and RTA share the corresponding contents. The concrete message interactions of the protocol are similar to Protocol 1.

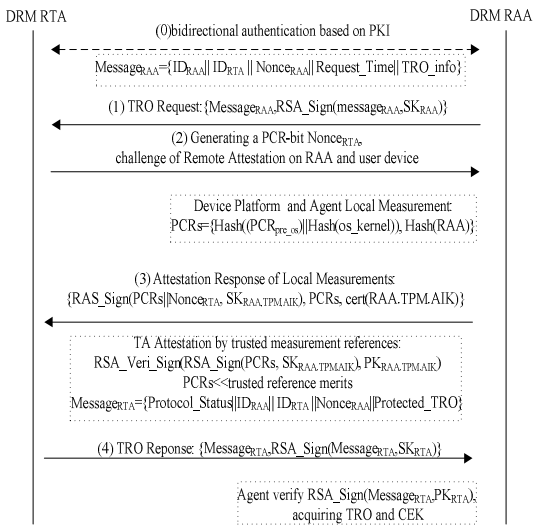

Figure5. RA-based TRO trusted distribution protocol

Protocol 3 (TRO Trusted Execution) before contents are called by a DRM application App, RAA validates the integrity of the App, and then generates a session key K_s that is used to encrypt called contents. App accepts the encrypted contents and the ciphered session key, then decrypts the contents by using K_s and presents them on user terminal device. The practical procedure is shown in Fig 6.

(0) User activates a session of usage on contents by App;

(1) RAA generates a random number Nonce'$_{RAA}$, and sponsors a RA challenge for App;

(2) Several key components of App, for instance DLL files and the security attribute-related files, are measured by OS kernel based on binary codes' hash operation, and these hash values are stored in PCR_{App}; PCR_{App} along with Nonce'$_{RAA}$ are executed RSA-signature operation by the private key of OS kernel, and the signature values together with the public key certificate of App cert (App) are sent to RAA;

(3) RAA validates the signature RSA_Sign (PCR_{App}, SK_{OS_kernel}) by using PK_{OS_kernel}, and compares PCR_{App} with the corresponding references; if the check is successful, a session key K_s is generated to protect contents in the session. K_s encrypted by the public key of App and ciphered contents $\{content\}_{Ks}$ are sent to App;

(4) App decrypts K_s by using its own private key SK_{App}, and then deciphered contents by the acquired K_s. When RAA are executing the usage control policies expressed in TRO, the related attributes of the permissions would change simultaneously, such as the decrease of display and usage time. There is an assumption that App and platform have no capabilities of copy, store and tamper with contents.

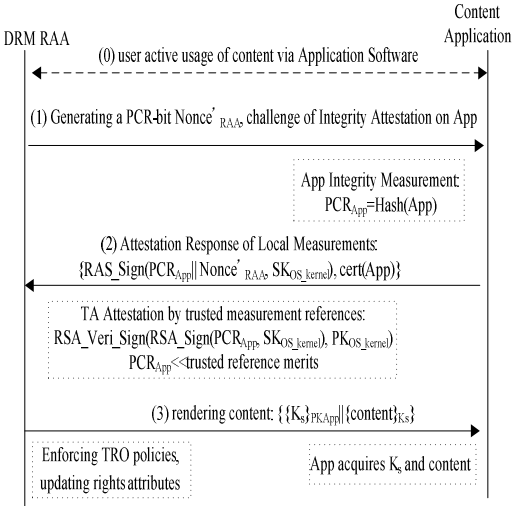

Figure6. TRO trusted enforcement protocol

4. Performances Analyses and Comparisons

We made a comparison among the proposed approach and existing representative DRM schemes, such as OMA DRM, Ref [3] and [4], as is shown in Table 1, where such symbols as "○", "×" and "-" show covering, lacking and not referring to the corresponding characteristics or functionalities, respectively.

Table 1.Comparisons of Existing Schemes

Performances	OMA	Ref [3]'s	Ref[4]'s	Ours
Contents Sharing	○	○	○	○
Local Domain	○	○	Limit	No Limit
License Enforce	×	×	LDM	No need LDM
RO & TRO Distribution	Domain-based RO	×	Proxy Certificate-DRO	Extensible ODRL-TRO
Transfer Granularity	Coarse-grained	-	Fine-grained	Fine-grained
Sharing Constraints	×	×	○	○
Time Limitation	○	×	○	○
Agent Trust	×	×	×	○
Rights Revocation	RI Control	LRL & GDRL	PC & PCRL	TRO RL
Cipher	PKI	Symm.	PKI	PKI
System Cost	Medium	Small	Large	Large

5. Conclusive Remarks

The paper proposed a fine-grained rights transfer policy and extensible ODRL-based transferable rights expression, further presented relative protocols, with a goal to implement the trusted distribution and enforcement of RO and TRO. Finally, we compare existing representative schemes with ours in several key performances. The future work is to formalize syntaxes and semantics of the fine-grained policy in order to guarantee the correctness and un-ambiguity of rights expression.

Acknowledgments

We would like to show gratitude to anonymous reviewers for their helpful comments and suggestions. The work is supported by National Natural Science Foundation of China Grant No. 60803150 and No. 60672112, as well as '111' Project Grant No.B08038.

References

[1] C. Hibbert, "A copy protection and content management system from The DVB," The DVB Consortium. http://www.dvb.org/ documents/newsletters/DVB-SCENE-05-Copy Protection Article. pdf.

[2]DRM Architecture Candidate Version 2.1.Open Mobile Alliance. Jul., 2007.

[3]B. Popescu, B. Crisop, A. Tanenbaum and F. Kamperman, "A DRM security architecture for home networks," In: Proceedings of 4th ACM Workshop on Digital Rights Management, Oct. 2004.

[4]H. KIM, Y. Lee and B. Chung, H. Yoon, J. Lee and K. Jung, "Digital Rights Management with right delegation for home networks," In: Proceedings of 9th International Conference on Information Security and Cryptology, M.S. Rhee and B. Lee (Eds.): 2006, LNCS 4296, pp. 233–245.

[5]C. Barlas, "Digital Rights Expression Languages," *JISC Technology and Standards Watch*, Jul. 2006.

[6]Open Digital Rights Language (ODRL) version 1.1, http://www.w3.org/TR/odrl, 2002.

[7]eXtensible rights Markup Language (XrML) 2.0 Specification, ContentGuard, Inc. Nov. 2001.

[8]Information technology—Multimedia framework Part 5: Rights Expression Language, .ISO/IEC 21000-5, 2004.

[9]DRM Rights Expression Language Candidate Version 2.1.Open Mobile Alliance. Jul 2007.

[10]TCG Specification Architecture Overview Revision 1.3.https://www.trustedcomputinggroup.org/Trusted Computing Group, Mar. 2007.

[11]DRM Specification Candidate Version 2.1.Open Mobile Alliance.http://www.openmobilealliance.org/Technical/PublicMaterial.aspx.Jul.2006.