draft-zhipeng-pkix-drm-proxy-architecture-01

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on April 24, 2010.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (http://trustee.ietf.org/license-info).
Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Abstract

This document specifies a method and its architecture for proxy DRM application based on the PKI environment with X.509 certificate
[X.509]. It can be a common solution for the proxy DRM application regardless the specific DRM standard which is implemented on the interface between the Client Device and Service Server.

Table of Contents

1. Introduction ......................................................... 3
2. Terminology .......................................................... 4
3. Proxy Architecture for DRM service .............................. 4
4. Trust Architecture for Proxy DRM service ....................... 5
5. Format of Authorization Document ................................. 6
   5.1. Authorization Document based on OMA DRM2.1 ............... 7
6. Initiation and Update of Authorization ........................... 10
7. Application Example .................................................. 12
8. Extended Architecture ............................................... 13
9. Security Considerations .............................................. 14
10. IANA Considerations .................................................. 14
11. Acknowledgements .................................................... 15
12. Normative References ............................................... 15
Authors’ Addresses .................................................... 15
1. Introduction

This document specifies a method and its architecture to fulfill that a device can apply for the DRM (Digital Rights Management) service or content on behalf of another device.

Below list two typical situations when an application MAY use this proxy method:

- In home network, the home server or home gateway can manage the home devices and subscribe services for these home devices and only the home server or home gateway has the right to subscribe services. Thus, the home devices can only access service under the permission of the home server or home gateway.

- In business environment, the device with higher grade can control the access rights of the devices with lower grade. So the lower grade device can only access some special services or some secret content through a server when it holds the authorization of a device with higher grade.

To verify the service subscription or service acquisition, the Service Provider (Service Server) SHOULD verify the trusted relationship between the request device and the proxy device. Currently, most DRM architectures are based on PKI technology. So, it is available to conduct such verification to support such proxy DRM service under the PKI environment with X.509 certification.

Generally, in order to set up the trusted relationship with proxy device, request device SHALL create an authorization document with its digital signature to authorize the proxy device of proxy responsibilities. The authorization document SHALL contain the detail service information that the proxy device can subscribe for the request device. After receiving the authorization document, the proxy device can follow the authorization to take the service operation for the request device.

Meanwhile the authorization between the request device and the proxy device SHALL be delivered to the Service Server. Thus the Service Server will only send or issue the service response which is compliant to the authorization to the proxy device. Finally, the service response will be transferred to the request device by the proxy device.

Under this proxy method and architecture, the proxy device on one hand can manage the service subscription of the request device, and on the other hand can take on most communication tasks with the Service Server to reduce the burden of request device.
As for the implementation, two issues SHOULD depend on the specific application:

The first issue is "types of DRM service". This document does not define the types of DRM service while section 5.1 gives the definition of Authorization Document based on the services defined in OMA DRM specification [OMA DRM2.1 TS] such as "RO Acquisition" and "Join Domain" are referred as an example.

The second issue is "message of DRM service". This document will not define the message of DRM service either. It can depend on the DRM specification. The method defined in this document can be used for any DRM devices if they support PKI and X.509 certificate.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Proxy Architecture for DRM service

The general service architecture for the proxy application is listed as Figure 1:

```
+----------+ DRM Request  +----------+ Proxy DRM Request  +----------+
| Request  | -----------> | Proxy    | ------------------>| Service  |
| Device A |<-----------  | Device B |<------------------ | Server   |
+----------+ DRM Response +----------+ Proxy DRM Response +----------+
```

Figure 1. Proxy Architecture for DRM Service

On the request flow, the Request Device A creates the "DRM Request" and sends it to the Proxy Device B. After verifying over the request successfully, the Proxy Device B creates and sends the "Proxy DRM Request" to Service Server to apply for service for the Request Device A. The "Proxy DRM Request" can contain the whole "DRM Request" as an element or contain relative elements picked up from the "DRM Request".

Similarly, on the response flow, the Service Server sends the "Proxy DRM Response" upon the "Proxy DRM Request" to the Proxy Device B. Then the Proxy Device B creates the "DRM Response" on the basis of the "Proxy DRM Response" and sends the "DRM Response" to the Request Device A. In general third party proxy services, such kind of interactive flow is rather usual. While for DRM application, the key
issue is to build up and to effectively deliver the trust mechanism since the Service Server SHOULD control the risk from all the devices that could access the service or content.

The solution defined in this document to deliver the trust mechanism is that the Request Device A sets up trusted authorization with the Proxy Device B and authorizes the Proxy Device B with the proxy rights to take service operations with the Service Server for the Request Device A. After the Proxy Device B completes the service session with the Service Server, it will send the response that corresponds to the trusted authorization back to the Request Device A. The mechanism to set up trusted relationship is fully elaborated in the following section 4.

4. Trust Architecture for Proxy DRM service

In bi-entity architecture, based on PKI environment, the trust verification is conducted via verifying the Certificate Chain of each other between the two entities of the service. So, to fulfill the trust verification among the three entities within the proxy service, a method is given that the Request Device creates an authorization to enumerate what right the Proxy Device can hold to subscribe the service for the Request Device A. The Trust Architecture is shown as Figure 2:

```
Certificate A        Certificate A + Certificate B
+--------------------+--------------------+
<p>| | |
|                    |                    |
| Request device A   | Proxy device B     |
| -----------+-----------------+-------------------|</p>
<table>
<thead>
<tr>
<th>Authorization</th>
<th>Service device A</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRM response</td>
<td>Proxy DRM response</td>
</tr>
</tbody>
</table>
```

Figure 2. Trust Architecture for proxy DRM Service

The Authorization can be contained in a document. The Authorization Document is sent to Proxy Device B as well as the Certificate Chain of Request Device A. The Proxy Device B verifies the secure status of Request Device A by checking certificate revocation list (CRL) [RFC5280] for the Certificate A. If successfully, then verifies the Authorization Document using the public Key of Request Device A that is enclosed in the Certificate A, including verifying the signature and the proxy rights enclosed in the Authorization Document. If all the verifications are correct, the Proxy Device B parses each element in the Authorization document to be aware of the proxy rights that are authorized by the Request Device A and confirms that the trusted relationship has been set up successfully.
In the proxy DRM service, the Authorization Document as well as the certificates of both Request Device A and Proxy Device B SHALL also be delivered to Service Server. Upon receiving the two Certificates and the Authorization Document, Service Server SHALL verify the secure statuses of both Request Device A and Proxy Device B by checking CRL for their certificates. If both devices are in secure status, then verifies the Authorization Document using the public Key of Request Device A that is enclosed in the Certificate A. Once the verification of the Authorization Document is successful, the Service Server will catch the rights that are authorized to Proxy Device B by Request Device A via parsing the elements in the Authorization document. For example, if the Authorization Document only contains the rights of 2-pass RO Acquisition, the Proxy Device can subscribe the ROs for the Request Device via the 2-pass RO Acquisition protocol defined in OMA DRM2.1 and can not subscribe for the Request Device for joining a Domain.

After all the operations above have been finished, the trusted relationship has been set up successfully for the proxy service. Thus, the Proxy Device B can only execute the proxy service operation under the authorization by the Request Device A. At the same time the Service Server will only issue the permitted service to the Request Device A via the Proxy Device B according to the Proxy DRM Request (For service flow, please refer to section 3).

5. Format of Authorization Document

The Authorization Document is composed of:
  -- name of Request Device
  -- CertId of Request Device
  -- name of Proxy Device
  -- CertId of Proxy Device
  -- authorized service rights
  -- extensions
  -- signature algorithm
  -- signature

The signature is computed across hash of the Authorization Document except the signature itself.

Since the "authorized service rights" depends on the real application and SHOULD be based on a specific DRM specification, this document just gives the definition of proxy rights based on [OMA DRM2.1 TS] for the OMA DRM service while the definitions of proxy rights can adapt to the real application when needed.
5.1. Authorization Document based on OMA DRM2.1

This sub-section provides a structure of Authorization Document. In this instance, the service types or service authorization are based on the OMA DRM2.1 specification (For details, please refer to [OMA DRM2.1 TS]).

Defined in the specification of OMA DRM2.1, the service operations include "2-pass RO Acquisition", "4-pass Confirmed RO Acquisition", "1-pass RO Acquisition", "3-pass Confirmed RO Acquisition", "2-pass Join Domain", "2-pass Leave Domain", "2-pass Metering Report" and "2-pass RO Upload".

The data structure for the Authorization Document is described as follows:

```c
AuthorizationDocument() {
    DeviceId()     // The id of Request Device A
    CertId()       // The key information of certificate A
    DeviceId()     // The id of Proxy Device B
    CertId()       // The key information of certificate B
    AuthorizedRights() // proxy rights authorized to Proxy Device B
    ExpireDate()   // The deadline of validity, UTC time
    SerialNumber   32 uimsbf
    SignatureAlgorithm() // Refer to X.509
    Signature()
}
```

```c
DeviceId() {
    Hash()
}
```

```c
CertId() {
    hashAlgorithmId  8 uimsbf // AlgorithmIdentifier
    issuerNameHash()            // Hash of Issuer’s name
    issuerKeyHash()             // Hash of Issuer’s public key
    serialNumber   32 uimsbf  // CertificateSerialNumber
}
```

The "hashAlgorithmId" is used to indicate the hash Algorithm. Following is the definitions of "hashAlgorithmId" in this document.

<table>
<thead>
<tr>
<th>hash Algorithm</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHA-1</td>
<td>0</td>
</tr>
<tr>
<td>rfu</td>
<td>1-255</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>rfu</td>
<td>1-255</td>
</tr>
</tbody>
</table>
issuerNameHash()
Hash()
}

issuerKeyHash()
Hash()
}

Hash()
// Hash Value, the Hash algorithm refers to [SHA-1]
OctetString8()
}

AuthorizedRights()
// The following service rights in proxy service are defined:
// 0    2-pass RO Acquisition
// 1    4-pass Confirmed RO Acquisition
// 2    1-pass RO Acquisition
// 3    3-pass Confirmed RO Acquisition
// 4    2-pass Join Domain
// 5    2-pass Leave Domain
// 6    2-pass Metering Report
// 7    2-pass RO Upload

// at most contain eight kinds of authorized right as listed above
for( i = 0; i < RightsNub; i++ ){
  ProxyRightsId    8    uimsbf
}

for( i = 0; i < RightsNub; i++ ){
  SuspendedRightsId    8    uimsbf
}

for( i = 0; i < AuthorizationDocumentNub; i++ ){
  // Serial Number of the Authorization Document to be suspended
  SuspendedAuthorizationNumber   32    uimsbf
}

ExpireDate()
// Deadline of Validity
OctetString8()
}

Signature()
// Signature Value
OctetString8()
The fields are defined as follows:

- `.length` - Length of the octet string
- `.octet` - A byte

The Authorization Document contains the essential information for other entities to verify the authorization of the Request Device to the Proxy Device. Following is the detailed description:

**DeviceId:** In OMA DRM 2.1, `<deviceID>` is defined as the hash of the Device’s public key info. Here that definition is followed and the hash algorithm is SHA-1 that is set as default hash algorithm in OMA DRM 2.1.

**CertId:** The essential information of the device’s Public Key Certificate, including `hashAlgorithmId`, `issuerNameHash`, `issuerKeyHash`, and `serialNumber`.

**AuthorizedRights:** It contains three parts: `ProxyRightsId`, `SuspendedRightsId` and `SuspendedAuthorizationNumber`. The `ProxyRightsId` indicates the right that the Request Device has authorized to the Proxy Device to subscribe a specific service. This document supports the services defined in OMA DRM 2.1, including 2-pass RO Acquisition, 4-pass Confirmed RO Acquisition, 1-pass RO Acquisition, 3-pass Confirmed RO Acquisition, 2-pass Join Domain, 2-pass Leave Domain, 2-pass Metering Report and 2-pass RO Upload. To support suspending a proxy right, the `SuspendedRightsId` is used to indicate which right is suspended (refer to the section of "Update of Authorization"). The `SuspendedAuthorizationNumber` indicates the suspended Authorization Document by its serial number.

**ExpireDate:** For security concern, the Authorization Document contains an "ExpireDate" element. Once that date exceeds the ExpireDate, the authorization turns invalid immediately.

**SerialNumber:** the serial number of the Authorization Document. The Request Device SHALL assign each Authorization Document a unique serial number.

**Signature and SignatureAlgorithm:** The Authorization Document contains the digital signature over the other part of the Authorization...
Document except the Signature itself under the request Device’s private device key. The SignatureAlgorithm indicates the Algorithm for creating the digital signature.

6. Initiation and Update of Authorization

The procedure of initiating or updating the Authorization can be an independent service flow. The Request Device will get a response for its request of Authorization Initiation or Authorization Update.

Once the authorization is expired, the trusted relationship between the Request Device and the Proxy Device will not exist any more. The Request Device needs to initiate a new Authorization. And during the valid duration, the authorization can also be updated. For example, the Request Device wants to authorize the Proxy Device some more rights, it can either suspend the original Authorization Document and then create a new Authorization Document or just create another Authorization Document containing the additional authorized rights. Hence the Proxy Device can conduct all the proxy rights up to date.

If the Request Device wishes to suspend the whole or part of the existing authorization, it can create an Authorization Document to the Proxy Device to indicate which Authorization Document or which proxy right SHOULD be suspended on the base that the Proxy Device is fully trusted by the Request Device. If to guarantee a higher security, the Request Device can take a measure to inform the Service Server that the trusted relationship to the Proxy Device or some a proxy right is suspended.

On the other side, in the real application, the Proxy Device can also suspend part or whole of the trusted relationship. Thus the Proxy Device will not apply for the suspended service for the Request Device any more.

The procedure of Authorization Initiation & Update is shown in Figure 3 below.
The possible status value of AuthorizationResponse and ProxyAuthorizationResponse are defined as below:

<table>
<thead>
<tr>
<th>status</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUCCESS</td>
<td>0</td>
</tr>
<tr>
<td>Certificate A invalid</td>
<td>1</td>
</tr>
<tr>
<td>Certificate B invalid</td>
<td>2</td>
</tr>
<tr>
<td>Certificate A&amp;B invalid</td>
<td>3</td>
</tr>
<tr>
<td>Authorization invalid</td>
<td>4</td>
</tr>
<tr>
<td>rfu</td>
<td>5-255</td>
</tr>
</tbody>
</table>

In the Authorization Request, the certificate of Request Device
A(certificate A) and the Authorization Document are contained. If the certificate A and Authorization Document are verified successfully by the Proxy Device B, and the Proxy Device B has accepted the Authorization request, then the Proxy Device B will send the Proxy Authorization Request to the Service Server, else return the Authorization Response with the status value indicating the failure reason. The Proxy Authorization Request contains the certificates of both Request Device A and Proxy Device B(certificate A and certificate B) and the Authorization Document. If the information in the Proxy Authorization Request is verified successfully and Service Server has accepted the Proxy Authorization Request, the Proxy Authorization Response will contain the status value of ‘SUCCESS’, else contain the status value according to the failure reason. Once the Proxy Device B receives the Proxy Authorization Response with the status value of ‘SUCCESS’, it will return the Request Device A the Authorization Response with the status value of ‘SUCCESS’, else if the failure reason in the Proxy Authorization Response is only on the side of Proxy Device B(such as the ‘Certificate B invalid’), it will correct the errors and send the Proxy Authorization Request again to the Service Server, else will return the Request Device A the Authorization Response with the status value of the failure reason.

7. Application Example

Take the home application as an example, the Home Gateway( or the Home Server) acts as the Proxy Device. The benefit is the User can just register one Device(the Home Server) to the Service Provider and apply for the services for all his personal devices as his desire very conveniently.

The procedure is very simple that the User operates a device to create and send an Authorization Document to the Home Server. Then he can use that device to apply for DRM service via the Home Server’s proxy at once after the Home Serve accepts the authorization. On this architecture the service charging will also be very convenient to the service since the charge can be bounded on the Home Server only.

On this architecture, another advantage is that the Home Server can actually control the other devices for applying the service. Somehow this architecture acts similarly as the "Parent Mechanism" to constrain the children’s content applications via controlling the devices of the children since the Home Server can judge and refuse the service applications of the Request Device, or refuse the "Authorized Rights" from the Request Device.
On some cases, the Request Device SHOULD ask the Home Server to suspend some service applications. For instance, the Request Device will be given to the User’s friend, then the Proxy relationship can be suspended once the Request Device sends the Authorization Document containing the suspending rights.

8. Extended Architecture

In the section 3, one Proxy Device is deployed in the Proxy Architecture for DRM Service. In fact, by the proxy mechanism defined above, a Multiple-Proxy Architecture can be implemented. It is shown in Figure 4 below:

```
DRM                  Proxy DRM                  Proxy DRM
+--------+ Request +---------+ Request-1 +---------+ Request-n +-------+
|Request | ------->|Proxy    | .........>|Proxy    |---------->|Service|
|Device A|<------- |Device-1 |<..........|Device-n |<----------|Server |
+--------+ DRM     +---------+ Proxy DRM +---------+ Proxy DRM +-------+
Response                     Response-1                       Response-n
```

Figure 4. Multiple-Proxy Architecture for DRM Service

In the Multiple-Proxy DRM Architecture, if the Proxy Device can not apply for service to the Service Server directly either, it can forward the Request to the next Proxy Device. On the return way, the Response will be delivered back one by one to the Request Device.

The Trust Architecture for Multiple-Proxy DRM Service is shown in Figure 5 below.

```
+---------------------------+---------------------------+
|Authorization             |Authorization             |
|Request                   |Proxy                     |
|Device A                  |Device-1                  |
+---------------------------+---------------------------+
|DRM Response              |Proxy                     |
|                           |DRM Response-1            |
|                           |Certificate-A             |
|                           |+  .......                |
|                           |  + Certificate-n         |
|                           |  + Authorization         |
|                           |  + Service               |
|                           |  + Server                |
|                           |  + Proxy DRM Response-n  |
```

Figure 5. Trust Architecture for Multiple-Proxy DRM Service
In the Trust Architecture for the Multiple-Proxy DRM Service, the certificates and authorization SHOULD be relayed until to the Service Server. If in one step, the verification of certificates and Authorization is not successful, the trusted relationship chain between the Request Device and the Service Server can not be built up. Hence the proxy service will not be available for the Request Device.

9. Security Considerations

To guarantee the Authorization Document is valid in limited period, it SHALL contain the expiration date which SHALL be earlier than the expiration date ("notAfter" element defined in X.509) of the certificates of both Request Device and Proxy Device. During the valid period, the Authorization Document does not need to be delivered in every service operation if the target entity (the Proxy Device or Service Server) has already held it.

If during the valid period, either the Request Device or the Proxy Device has updated its certificate, this Authorization Document will expire automatically. It means either the Proxy Device or the Service Server will reject the DRM Request or Proxy DRM Request if the Authorization Document can not be verified successfully. Therefore the Request Device needs to create a new one if it like to continue the proxy service.

Regarding both the Request Device and Proxy Device, they are completely DRM devices and SHALL obey the DRM specification strictly and SHALL own the security capability requested for DRM device in the DRM specification. Therefore all the related security measures defined in DRM specification SHALL be completely implemented in these Devices. The Proxy Device MAY need to register with Service Device and only the registered Proxy Device can conduct the Proxy Service. But the Proxy Device will be able to take proxy service for any Request Device although it SHOULD send the certificate of the Request Device to the Service Server to ensure that the Service Server can verify the Request Device’s security status.

Regarding the Service Server, it SHALL obey the DRM specification too. For instance, it acts as the RI (Rights Issuer) that is defined in OMA DRM2.1 [OMA DRM2.1 TS].

10. IANA Considerations

This document requires no actions from IANA.
11. Acknowledgements

Many thanks to all the members of Huawei DRM project team. Countless discussions and ceaseless effort have surely helped us transcend each tough process one after another. Great and Cheer!

Also many thanks to all the people I know in the DRM-related Chinese and international standard organizations. Interesting debates and discussions help me much and will remain in my mind forever. Wish all of us big success in the future.

12. Normative References


[OMA DRM2.1 TS] OMA-TS-DRM_DRM-V2_1-20081106-A, "DRM Specification".


Authors’ Addresses

Zhipeng Zhou (editor)
Huawei Technologies, Inc.
Floor 2, Building A, NO.48, Ning Nan AV.,
Nanjing, Jiangsu Province  210001
P.R.China

Phone: +86-25-82276771
Email: zhouzp@huawei.com
Ge Chen  
China Telecom  
109 West Zhongshan Ave, Tianhe District  
Guangzhou, Guangdong Province  510630  
P.R.China

Phone: +86-20-38639766  
Email: cheng@gsta.com