ISO /IEC JTC 1/SC 29 /WG 1 N83012
Date: 2019-03-13

ISO/IEC JTC1/SC29/WG1
(ITU-T SG16)

Coding of Still Pictures

JBIG
Joint Bi-level Image Experts Group

JPEG
Joint Photographic Experts Group

TITLE: Contribution to DIS 19566-4: Example with Privacy Rules (v2)

SOURCE: Jaime Delgado, Silvia Llorente, Daniel Naro (DMAG-UPC)
Distributed Multimedia Applications Group (DMAG)
Universitat Politècnica de Catalunya – UPC BarcelonaTECH
Jordi Girona, 1-3. Mòdul D6 Campus Nord
08034 Barcelona, Spain
jaime.delgado@ac.upc.edu

PROJECT: ISO/IEC 19566-4 (JPEG Systems – Privacy, security and IPR features)

STATUS: Proposal

REQUESTED ACTION: Input contribution

DISTRIBUTION: WG1

1. Introduction

This document is the second version of a refinement of our contribution ISO/IEC JTC 1 SC 29/WG1 N83011. In that input document, we have proposed a small modification to Annex A.2.3 describing how access rules can be referenced from the Protection box. In addition, we have provided an example of use.

In this new contribution we re-organize the example in line with other contributions that provide more details. In this way, we try to facilitate integration in the DIS version of ISO/IEC 19566-4.

2. New subclause in the new Annex B

B.x Encryption with Access Rules (informative)

As indicated in A.2.3, the access rules associated to an image, if any, are stored in a referenced JUMBF Box.

Access rules may be expressed using eXtensible Access Control Markup Language (XACML) [1], so in this case the JUMBF Box must be of XML Content type.

Figure B.x shows how the protection box can reference the XML JUMBF Box containing the corresponding access rules. In this example, AES 256 CBC encryption method is used, as indicated in the METHOD field with the code 0000 0100.

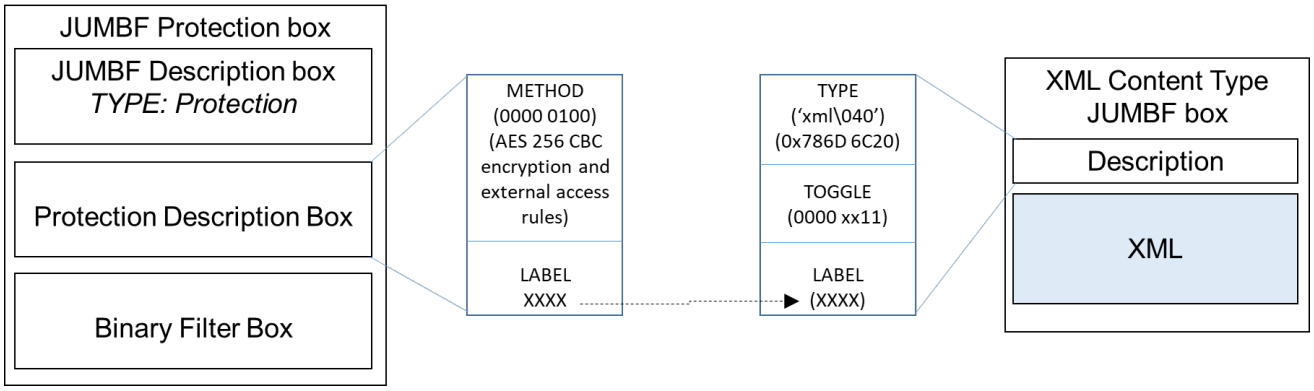

Figure B.x — JUMBF protection Box referring to XML Content Type JUMBF Box

Figure B.y shows a complete XACML policy, containing several access rules, which is the content of the XML JUMBF box. The first rule inside the policy defines the conditions to indicate that any user can **view** the image **urn:mimage:Desert.jpg** before the end of the year (**31/12/2019**). The second one denies any other operation over the image.

```
<Policy xmlns="urn:oasis:names:tc:XACML:3.0:core:schema:wd-17"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    PolicyId="urn:isdcm:policyid:1"
    RuleCombiningAlgId="urn:oasis:names:tc:XACML:1.0:rule-combining-algorithm:first-applicable"
    Version="1.0"
    xsi:schemaLocation="urn:oasis:names:tc:XACML:3.0:core:schema:wd-17
    http://docs.oasis-open.org/XACML/3.0/XACML-core-v3-schema-wd-17.xsd">

    <Description>Desert.jpg</Description>

    <Rule Effect="Permit" RuleId="urn:oasis:names:tc:XACML:2.0:ejemplo:Desert">
      <Description>
        Any user can view urn:mimage:Desert.jpg before the end of the year
      </Description>
      <Target>
         <AnyOf>
            <AllOf>
               <!-- Which resource -->
               <Match
                 MatchId="urn:oasis:names:tc:XACML:1.0:function:regexp-string-match">
                 <AttributeValue
                   DataType="http://www.w3.org/2001/XMLSchema#string">
                     urn:mimage:Desert.jpg
                 </AttributeValue>
                 <AttributeDesignator
                   AttributeId="urn:oasis:names:tc:XACML:1.0:resource:resource-id"
                   Category="urn:oasis:names:tc:XACML:3.0:attribute-category:resource"
                   DataType="http://www.w3.org/2001/XMLSchema#string"
                   MustBePresent="false"/>
               </Match>

               <!-- Which action -->
               <Match MatchId="urn:oasis:names:tc:XACML:1.0:function:string-equal">
                 <AttributeValue
                   DataType="http://www.w3.org/2001/XMLSchema#string">
                     View
                 </AttributeValue>
                 <AttributeDesignator
                   AttributeId="urn:oasis:names:tc:XACML:1.0:action:action-id"
                   Category="urn:oasis:names:tc:XACML:3.0:attribute-category:action"
                   DataType="http://www.w3.org/2001/XMLSchema#string"
                   MustBePresent="false"/>
               </Match>
            </AllOf>
         </AnyOf>
      </Target>
      <Condition>
         <Apply FunctionId="urn:oasis:names:tc:XACML:1.0:function:date-less-than-or-equal">
            <Apply FunctionId="urn:oasis:names:tc:XACML:1.0:function:date-one-and-only">
               <AttributeDesignator AttributeId="accessDate"
                 Category="urn:oasis:names:tc:XACML:3.0:date"
                 DataType="http://www.w3.org/2001/XMLSchema#date" MustBePresent="false"/>
            </Apply>
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#date">
               2020-01-01
            </AttributeValue>
         </Apply>
      </Condition>
    </Rule>
    <Rule RuleId="urn:oasis:names:tc:XACML:2.0:FinalRule" Effect="Deny"/>
</Policy>
```

Figure B.y — XML Content Type JUMBF Box for a XACML Policy containing several access rules

B.x.1 Example of encryption and access rules addition process

Figure B.x.1 shows how an image is encrypted and the access rules governing its usage are also added to the resulting file.

Starting from the original image, metadata is transferred to the resulting image. Then, the image stream is encrypted and the result of the encryption is stored in the Binary Filter Box inside the Protection box. Afterwards, the access rules and encryption information is added to the Protection box. The information

related to access rules is referenced by means of a label. Finally, a new XML Content type JUMBF box is added, which contains the XACML access rules that govern the usage of the protected image file.

Figure B.x.2 shows the new JUMBF boxes resulting from the encryption and access rules addition process.

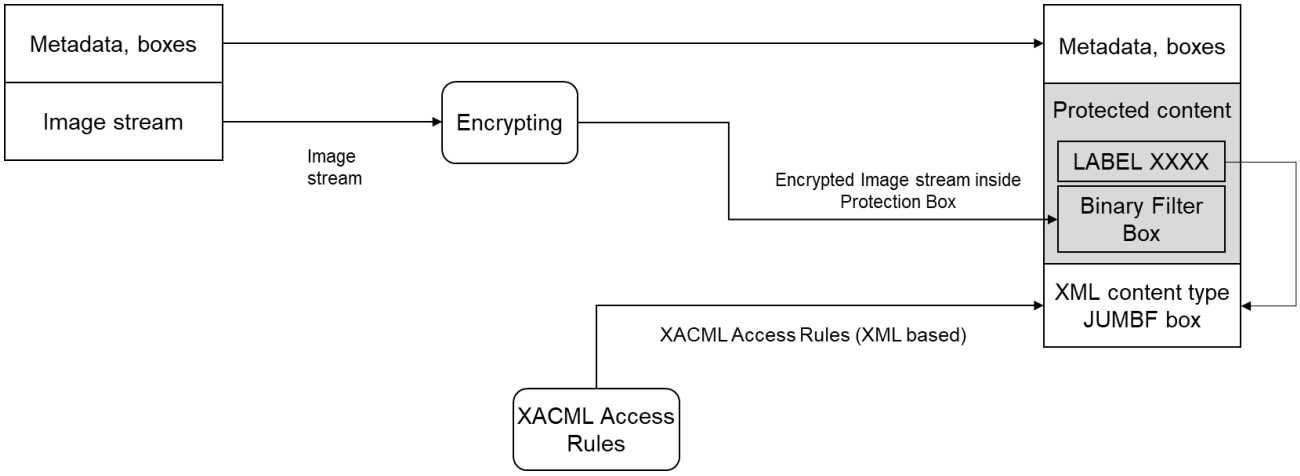

Figure B.x.1 — Encrypting and adding XACML access rules to an image

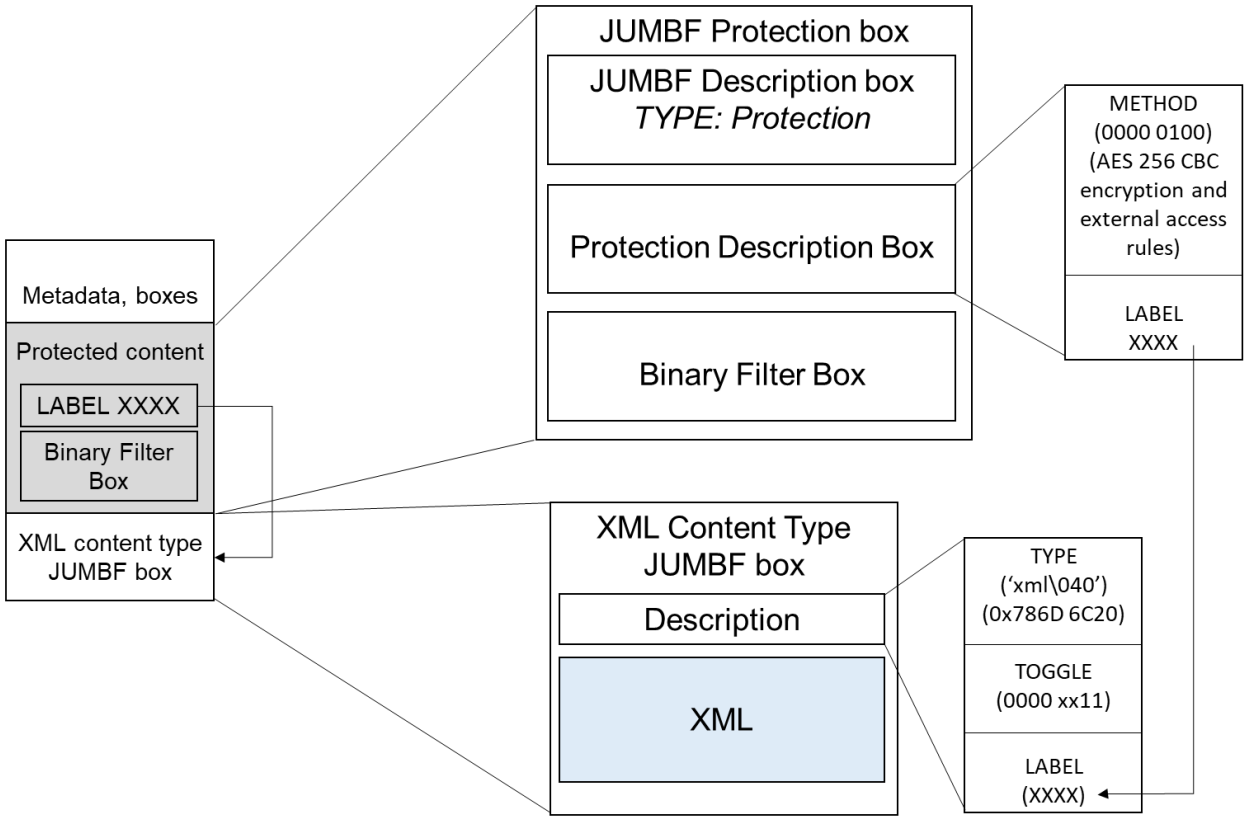

Figure B.x.2 — Generated JUMBF boxes by encryption and access rules process

B.x.2 Example of authorization and decryption process

Figure B.x.3 shows the authorization and decryption (if any) process when access rules are present. In this case, first, the label referencing the XML content type JUMBF box containing the access rules (LABEL

XXXX inside the JUMBF protection box) must be obtained from the JUMBF protection box. Then, Access Rules must be extracted from the XML content type JUMBF box. The access rules are given to the Authorization service, which also needs an Authorization Request (as specified in the XACML standard [1]). Then, if the authorization is granted, the decryption process obtains the encryption information and the encrypted image from the JUMBF protection box. After that, the Decryption process performs the decryption of the image, generating a result file with the corresponding metadata and the decrypted image stream. If the operation is not authorized according to the access rules, the resulting image contains only the image information which was not encrypted in the original image.

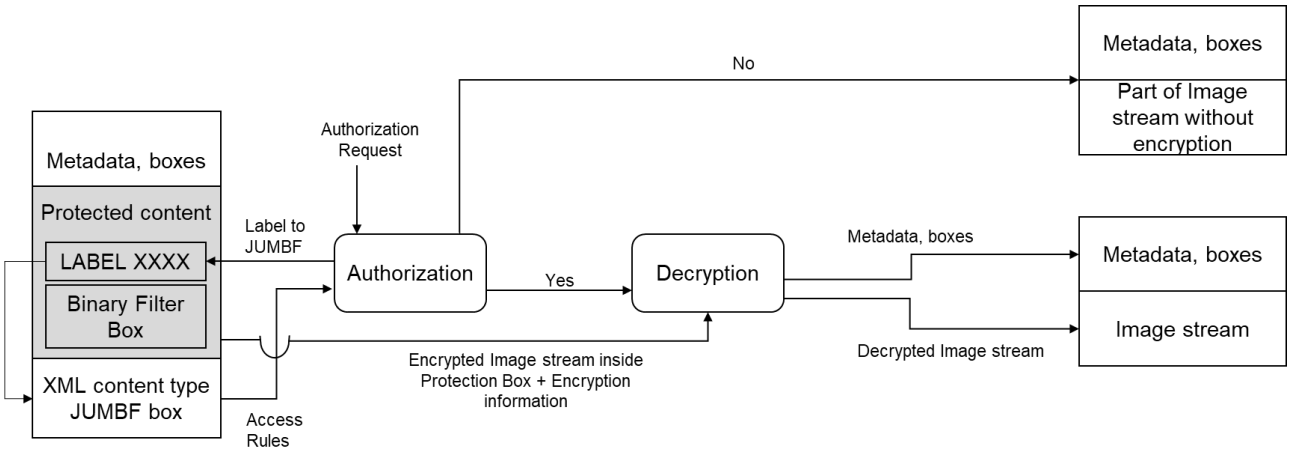

Figure B.x.3 — Authorizing access and decrypting an image

Bibliography

[1] OASIS, eXtensible Access Control Markup Language (XACML) Version 3.0, 2013, Available: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-cs-01-en.pdf.