文章
姚 鑫 · 六月 24, 2021 阅读大约需 6 分钟

第十七章 加密XML文档

第十七章 加密XML文档

本章介绍如何加密XML文档。

提示：发现在此命名空间中启用SOAP日志记录非常有用，这样就可以收到有关任何错误的更多信息。

关于加密的XML文档

加密的XML文档包括以下元素：

- <EncryptedData>元素，其中包含由随机生成的对称密钥加密的加密数据。(使用对称密钥加密比使用公钥加密更有效。)
- 至少有一个<EncryptedKey>元素。每个<EncryptedKey>元素携带用于加密数据的对称密钥的加密副本;它还包含一个带有公钥的X.509证书。拥有匹配私钥的接收方可以解密对称密钥，然后解密<EncryptedData>元素。
- (可选)其他明文元素。

```
<?xml version="1.0" encoding="utf-8"?>

<Container xmlns="http://www.w3.org/2001/04/xmlenc#">
  <EncryptedKey>
    <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p">
      <DigestMethod xmlns="http://www.w3.org/2000/09/xmldsig#" Algorithm="http://www.
w3.org/2000/09/xmldsig#sha1"></DigestMethod>
    </EncryptionMethod>
    <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
      <X509Data>
        <X509Certificate>MIICnDCCAYQCAWUwDQYJKo... content omitted</X509Certificate>
      </X509Data>
    </KeyInfo>
    <CipherData>
      <CipherValue>J2DjVgcB8vQx3UCy5uejMB ... content omitted</CipherValue>
    </CipherData>
    <ReferenceList>
      <DataReference URI="#Enc-E0624AEA-9598-4436-A154-F746B07A2C55"/>
    </ReferenceList>
  </EncryptedKey>
  <EncryptedData Id="Enc-E0624AEA-9598-4436-A154-F746B07A2C55" Type="http://www.w3.or
g/2001/04/xmlenc#Content">
    <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"></Encry
ptionMethod>
    <CipherData>
      <CipherValue>LmoBK7+nDelTOsC3 ... content omitted</CipherValue>
    </CipherData>
  </EncryptedData>
</Container>
```

要创建加密文档，请使用类%XML.Security.EncryptedData和%XML.Security.EncryptedKey。这些启用XML的类投影到适当名称空间中的有效<EncryptedData>和<EncryptedKey>元素。

创建加密的XML文档

创建加密的XML文档的最简单方法如下：

1. 定义并使用可以直接投影到所需XML文档的通用容器类。
2. 创建包含要加密的XML的流。
3. 加密该流，并将其与相应的加密密钥一起写入容器类的相应属性。
4. 为容器类生成XML输出。

加密的前提条件

在加密文档之前，必须创建包含要将加密文档发送到的实体的证书的IRIS凭据集。在这种情况下，不需要(也不应该拥有)关联的私钥。

容器类的要求

一个通用容器类必须包括以下内容:

- 类型为%XML.Security的属性。
  被投影为<EncryptedData>元素的EncryptedData。

这个属性将携带加密的数据。

- 至少一个类型为%XML.Security的属性。被投影为<EncryptedKey>元素的EncryptedKey。

这些属性将携带相应的密钥信息。

示例如下:

```
Class XMLEncryption.Container Extends (%RegisteredObject, %XML.Adaptor)
{

Property Data As %XML.Security.EncryptedData(XMLNAME = "EncryptedData");

Property Key As %XML.Security.EncryptedKey(XMLNAME = "EncryptedKey");

Parameter NAMESPACE = "http://www.w3.org/2001/04/xmlenc#";

}
```

生成加密的XML文档

要生成并编写加密文档，请执行以下操作:

1. 创建包含XML文档的流。

为此，通常使用%XML.Writer将启用XML的对象的输出写入流。

2. 创建%SYS.X509Credentials的至少一个实例,将访问要向其提供加密文档的实体的InterSystems IRIS凭据集。为此，请调用此类的GetByAlias()类方法。例如：

```
 set credset=##class(%SYS.X509Credentials).GetByAlias("recipient")
```

若要运行此方法，必须以该凭据集的OwnerList中包含的用户身份登录，否则OwnerList必须为空。

3. 至少创建%XML.Security.EncryptedKey实例。若要创建此类的实例，请使用此类的CreateX509()类方法。例如：

```
 set enckey=##class(%XML.Security.EncryptedKey).Createx509(credset,encryptionOptions,
referenceOption)
```

- credset是%SYS的实例。
  x509credentials在刚刚创建的新窗口中打开。
- encryptionOptions是$$$SOAPWSIncludeNone(还有其他选项，但它们不适用于此场景)。

此宏在%soap.inc包含文件中定义。

- referenceOption指定了对加密元素的引用的性质。

这里使用的宏在%soap.inc包含文件中定义。

4. 在创建%Library.ListOfObjects实例，并使用其Insert()方法在刚创建插入%XML.Security.EncryptedKey实例。
5. 使用%New()方法创建%XML.Security.EncryptedData实例。例如：

```
 set encdata=##class(%XML.Security.EncryptedData).%New()
```

6. 使用%XML.Security.EncryptedData的EncryptStream()实例方法加密在步骤2中创建的流。例如：

```
 set status=encdata.EncryptStream(stream,encryptedKeys)
```

- stream 流是在步骤1中创建的流。
- encryptedKeys是在步骤4中创建的密钥列表。

7. 创建并更新容器类的实例。

- 将键列表写入此类的相应属性。
- 将 %XML.Security.EncryptedData的实例写入此类的相应属性。
8. 使用%XML.Writer为容器类生成输出。

例如，前面显示的CONTAINER类还包括以下方法：

```
/// w ##class(XMLEncryption.Container).Demo("E:\temp\SecurityXml.txt")
ClassMethod Demo(filename = "", obj = "")
```

```
{
#Include %soap

    if (obj = "") {
        s obj = ##class(MyApp.Person).%OpenId(1)
   }

    //????XML??????
    set writer = ##class(%XML.Writer).%New()
    set stream = ##class(%GlobalCharacterStream).%New()
    set status = writer.OutputToStream(stream)
    if $$$ISERR(status) {do $System.Status.DisplayError(status) quit }
    set status = writer.RootObject(obj)

    if $$$ISERR(status) {do $System.Status.DisplayError(status) quit }
    do stream.Rewind()

    set container = ..%New()             ; ??????????
    set cred = ##class(%SYS.X509Credentials).GetByAlias("servercred")
    set parts  =$$$SOAPWSIncludeNone
    set ref = $$$KeyInfoX509Certificate


    set key = ##class(%XML.Security.EncryptedKey).CreateX509(cred, parts, ref)

    set container.Key = key        ; ????????

    //?????????(???????)
    set keys = ##class(%Collection.ListOfObj).%New()
    do keys.Insert(key)

    set encdata = ##class(%XML.Security.EncryptedData).%New()

    set status = encdata.EncryptStream(stream, keys)
    set container.Data = encdata   ; ????????

    // ??????
    set writer = ##class(%XML.Writer).%New()
    set writer.Indent = 1
    if (filename'="") {
        set status = writer.OutputToFile(filename)
        if $$$ISERR(status) {do $system.OBJ.DisplayError(status) quit}
    }
    set status = writer.RootObject(container)
    if $$$ISERR(status) {do $system.OBJ.DisplayError(status) quit}
}
```

此方法可以接受任何启用XML的类的OREF；如果没有提供，则使用默认值。

解密加密的XML文件

解密的前提条件

在解密加密的XML文档之前，必须同时提供以下两项：

- IRIS要使用的受信任证书。
- IRIS凭据集，其私钥与加密中使用的公钥匹配。

解密文档

要解密加密的XML文档，请执行以下操作：

1. 创建%XML.Reader实例打开并使用它打开文档。
2. 获取Document属性，%XML.Reader实例。
   其中包含作为DOM的XML文档。
3. 使用阅读器的correlation()方法将<EncryptedKey>元素或元素与类%XML.Security.EncryptedKey关联起来。
   例如:

```
 do reader.Correlate("EncryptedKey","%XML.Security.EncryptedKey")
```

4. 遍历文档以读取<EncryptedKey>元素或多个元素。
   为此，可以使用阅读器的Next()方法，该方法通过引用返回一个导入的对象(如果有的话)。
   例如:

```
if 'reader.Next(.ikey,.status) {
    write !,"Unable to import key",!
    do $system.OBJ.DisplayError(status)
    quit
    }
```

导入的对象是%XML.Security.EncryptedKey的实例。

5. 创建%Library.ListOfObjects的实例。
   并使用它的Insert()方法插入%XML.Security.EncryptedKey的实例。
   刚从文档中获得的。

6. 调用类%XML.Security.EncryptedData的ValidateDocument()方法

```
 set status=##class(%XML.Security.EncryptedData).ValidateDocument(.doc,keys)
```

第一个参数(通过引用返回)是在第2步中检索到的DOM的修改版本。
第二个参数是上一步中的键列表。

7. 可以选择使用%XML.Writer为修改后的DOM生成输出。

例如，前面显示的CONTAINER类包含以下类方法：

```
ClassMethod DecryptDoc(filename As %String)
{
#Include %soap
    set reader = ##class(%XML.Reader).%New()
    set status = reader.OpenFile(filename)
    if $$$ISERR(status) {do $System.Status.DisplayError(status) quit }
```

```
    set doc = reader.Document
    //??<Signature>??
    do reader.Correlate("EncryptedKey","%XML.Security.EncryptedKey")
    if 'reader.Next(.ikey,.status) {
        write !,"??????",!
        do $system.OBJ.DisplayError(status)
        quit
    }

    set keys = ##class(%Collection.ListOfObj).%New()
    do keys.Insert(ikey)
    // ??????????
    set status = ##class(%XML.Security.EncryptedData).ValidateDocument(.doc,keys)

    set writer = ##class(%XML.Writer).%New()
    set writer.Indent = 1
    do writer.Document(doc)
    quit $$$OK
}
```

#Caché

源URL:
https://cn.community.intersystems.com/post/%E7%AC%AC%E5%8D%81%E4%B8%83%E7%AB%A0-%E5%8A%A0%E5%AF%86xml%E6%96%87%E6%A1%A3