Security in Internet Applications
(XML security)

2024/25 Q2

Jaime Delgado

DAC - UPC

DMAG
DISTRIBUTED MULTIMEDIA APPLICATIONS GROUP

XML security

- XML Encryption
- XML Signature

XML Encryption

- W3C Recommendation *(2013)*
  - XML Encryption Syntax and Processing (version 1.1)
    http://www.w3.org/TR/xmlenc-core/
- Specifies a process to encrypt data:
  - Arbitrary data (including XML documents)
  - XML element
  - Content of a XML element
- The result is represented in XML:
  - Element `EncryptedData` (XML Encryption)
    - Contains the ciphered data (in one of its child elements)
    - References the ciphered data (through an URI reference)

Encrypting process

- If a XML element or its content is encrypted, the `EncryptedData` element will replace the element or content (in the encrypted version)
- If the complete document is encrypted, it will be the root element of a new XML document

XML encryption: Info to encrypt

```
<?xml version=”1.0”?>
<PaymentInfo xmlns=”http://example.org/paymentv2”>
    <Name>John Smith</Name>
    <CreditCard Limit=”5000” Currency=”USD”>
        <Number>4019 2445 0277 5567</Number>
        <Issuer>Example Bank</Issuer>
        <Expiration>05/24</Expiration>
    </CreditCard>
</PaymentInfo>
```

XML encryption: Info to encrypt

```
<?xml version=”1.0”?>
<PaymentInfo xmlns=”http://example.org/paymentv2”>
    <Name>John Smith</Name>
    <CreditCard Limit=”5000” Currency=”USD”>
        <Number>4019 2445 0277 5567</Number>
        <Issuer>Example Bank</Issuer>
        <Expiration>05/24</Expiration>
    </CreditCard>
</PaymentInfo>
```

XML encryption: *Complete card*

```
<?xml version=”1.0”?>
<PaymentInfo xmlns=”http://example.org/paymentv2”>
    <Name>John Smith</Name>
    <EncryptedData
                xmlns=”http://www.w3.org/2001/04/xmlenc#”
                Type=”http://www.w3.org/2001/04/xmlenc#Element”>
        <CipherData>
            <CipherValue>A23B45C56</CipherValue>
        </CipherData>
    </EncryptedData>
</PaymentInfo>
```

XML encryption: Info to encrypt

```
<?xml version=”1.0”?>
<PaymentInfo xmlns=”http://example.org/paymentv2”>
    <Name>John Smith</Name>
    <CreditCard Limit=”5000” Currency=“USD”>
        <Number>4019 2445 0277 5567</Number>
        <Issuer>Example Bank</Issuer>
        <Expiration>05/24</Expiration>
    </CreditCard>
</PaymentInfo>
```

XML encryption: Card number

```
<?xml version="1.0"?>
<PaymentInfo xmlns="http://example.org/paymentv2">
    <Name>John Smith</Name>
    <CreditCard Limit="5000" Currency="USD">
        <Number>
            <EncryptedData
                xmlns="http://www.w3.org/2001/04/xmlenc#"
                Type="http://www.w3.org/2001/04/xmlenc#Content">
                <CipherData>
                    <CipherValue>2B3746D5</CipherValue>
                </CipherData>
            </EncryptedData>
        </Number>
        <Issuer>Example Bank</Issuer>
        <Expiration>05/24</Expiration>
    </CreditCard>
</PaymentInfo>
```

XML encryption: Info to encrypt

```
<?xml version=”1.0”?>
<PaymentInfo xmlns=”http://example.org/paymentv2”>
    <Name>John Smith</Name>
    <CreditCard Limit=“5000” Currency=“USD”>
        <Number>4019 2445 0277 5567</Number>
        <Issuer>Example Bank</Issuer>
        <Expiration>05/24</Expiration>
    </CreditCard>
</PaymentInfo>
```

XML encryption: complete XML

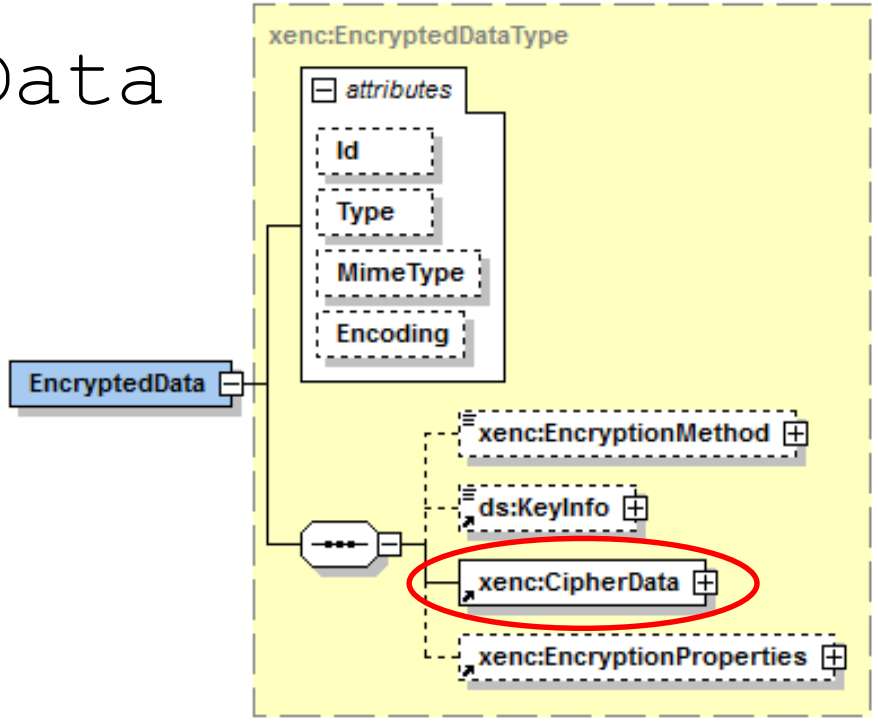
```
<?xml version="1.0"?>
<EncryptedData
        xmlns="http://www.w3.org/2001/04/xmlenc#"
        MimeType="text/xml">
    <CipherData>
        <CipherValue>C834A24D65</CipherValue>
    </CipherData>
</EncryptedData>
```

Model

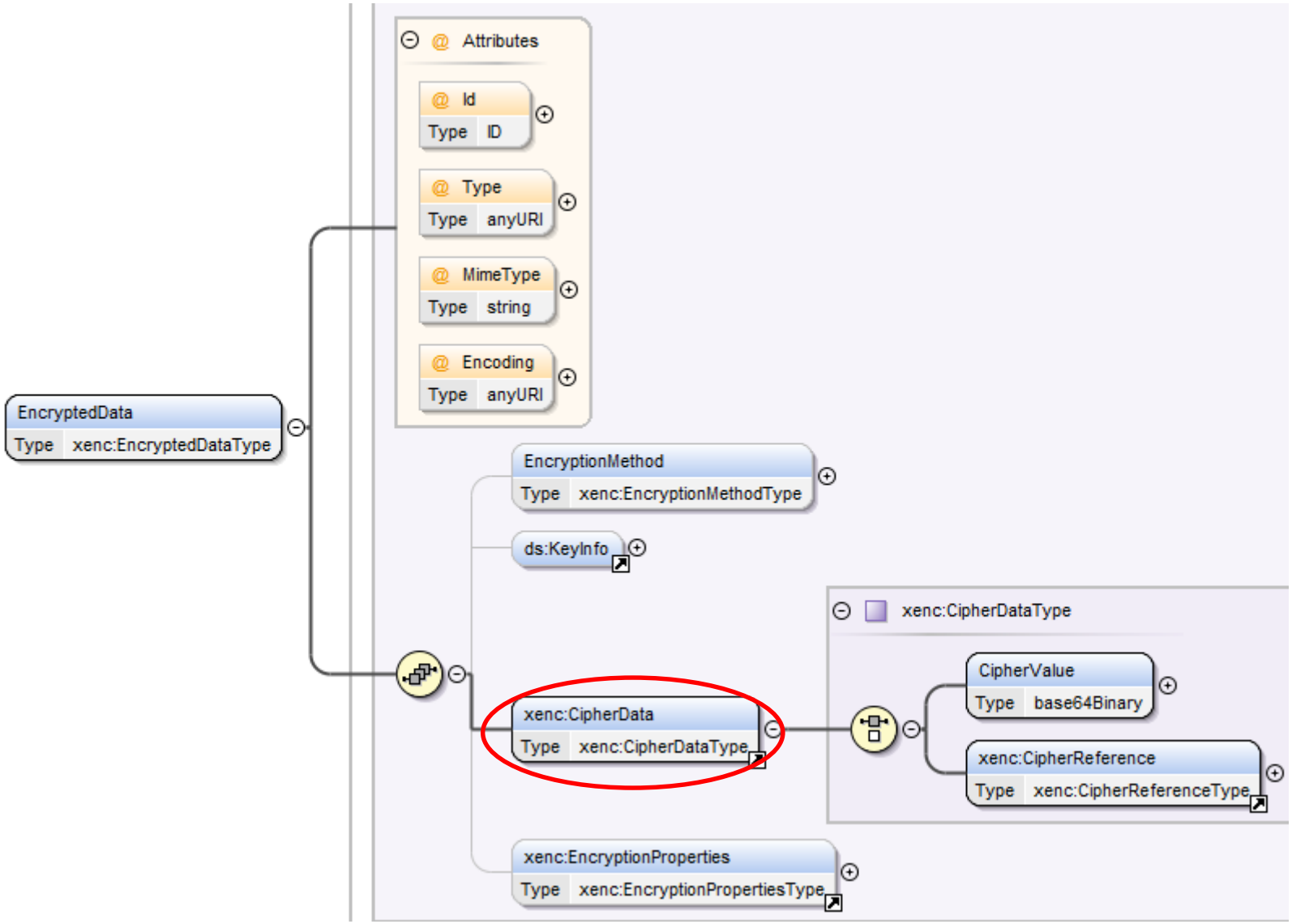
- The specification uses XML Schemas

Prefix	Namespace
xenc	http://www.w3.org/2001/04/xmlenc#

- `EncryptedData`



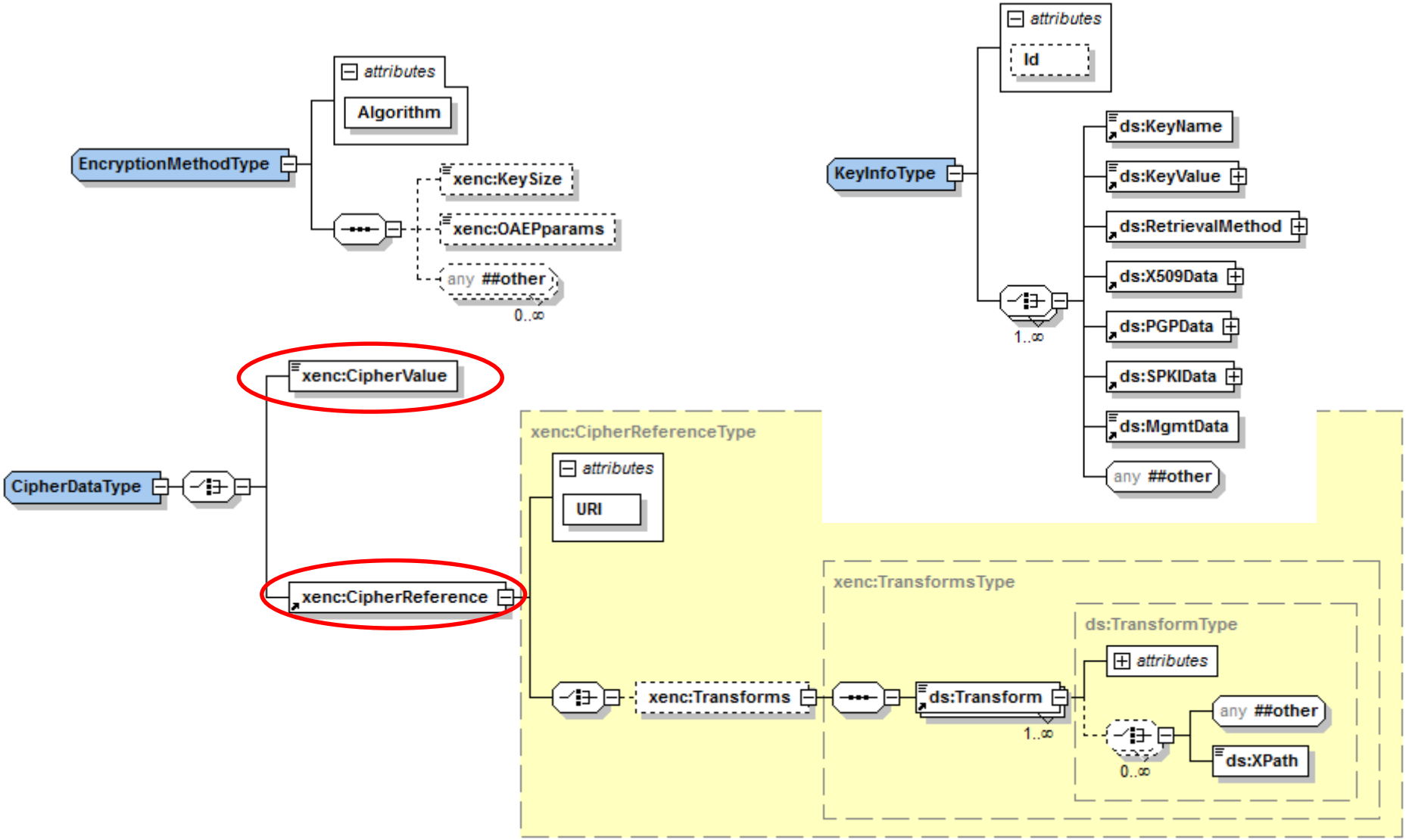
EncryptedData element structure



Encrypting process

1. Select algorithm (and parameters) to use.
   Element `xenc:encryptionMethod`
2. Obtain the key. Element `ds:keyInfo`
3. Encrypt the data (octets result of serializing data in UTF-8)
4. Build the structure `xenc:EncryptedData`
   (for the key `xenc:EncryptedKey`)
   i. If ciphered data included: element `xenc:CipherValue`
   with the sequence of encrypted octets coded in base 64
   ii. If ciphered data kept externally: element `xenc:CipherReference`
   with the URI and tranforms (if any)

Element types – EncryptedData



XML security

- XML Encryption

- **XML Signature**

Digital signature (reminder)

- **Integrity**: If the signed information is modified, signature validation will fail
- **No repudiation**: The entity who signed the information cannot claim it did not sign
- **Authentication**: We know who signed, since we can validate it with its public key

XML Signature

- W3C Recommendation *(2013)*
  - XML Signature Syntax and Processing (Version 1.1)
    http://www.w3.org/TR/xmldsig-core/

- Defines a XML syntax for the digital signature that provides integrity, and messages and signatories authentication

XML signatures

- *Over data external to the signature element:*
  - **Detached**: Used to sign a resource out of the XML document that contains the signature. Related through an URI.
- *Over data within the same XML file as the signature:*
  - **Enveloped**: Used to sign a part of the document that contains the signature. The signature is an element of the XML content signed (signature not to include their own value in the calculation of the `SignatureValue`).
  - **Enveloping**: The XML signature contains the signed data inside itself. Related through fragment identifiers.

XML signature: Element to add

<Signature>

https://www.w3.org/TR/2002/REC-xmldsig-core-20020212/xmldsig-core-schema.xsd

<SignedInfo>

<CanonicalizationMethod />

<SignatureMethod />

<Reference>

<Transforms>

<DigestMethod>

<DigestValue>

</Reference>

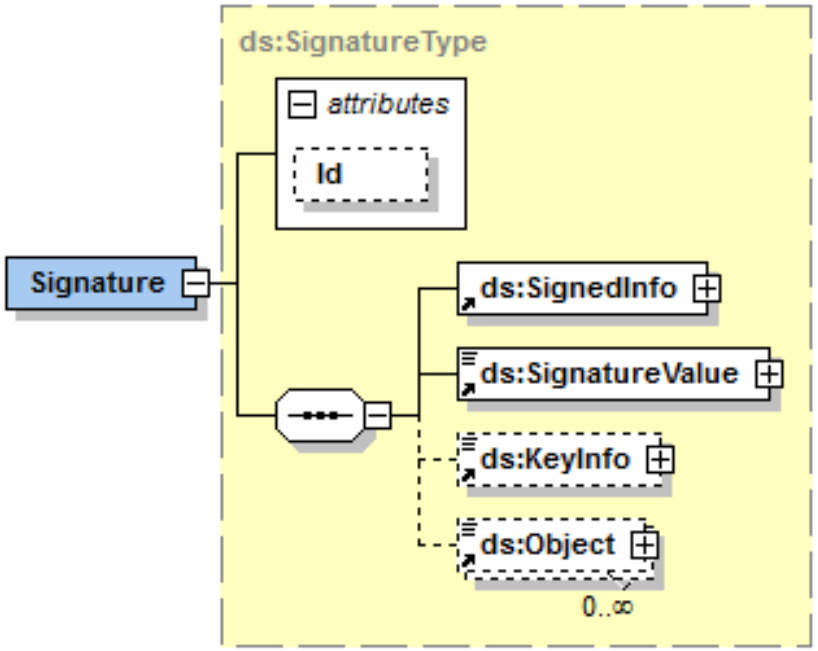
</SignedInfo>

<SignatureValue />

<KeyInfo />

<Object />

</Signature>

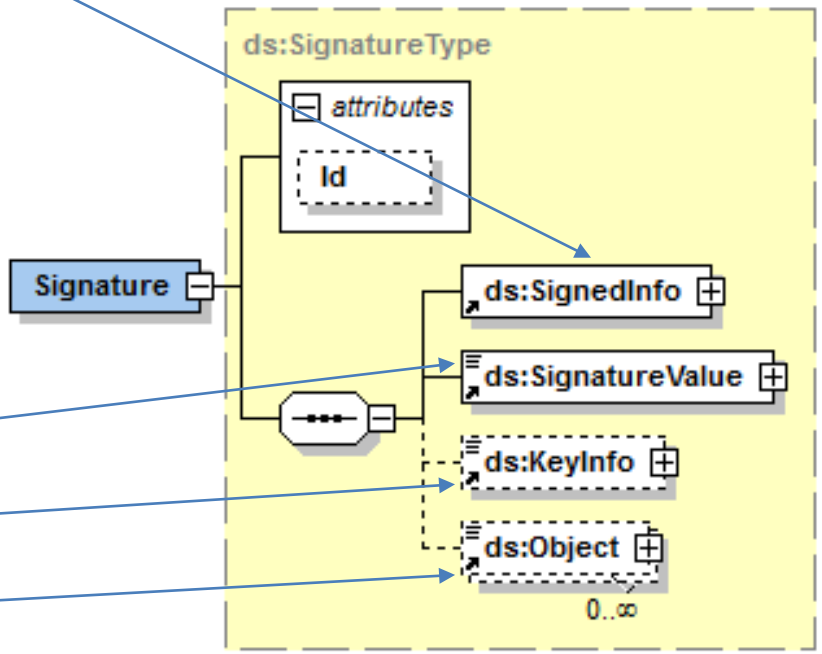


XML signature: Element to add

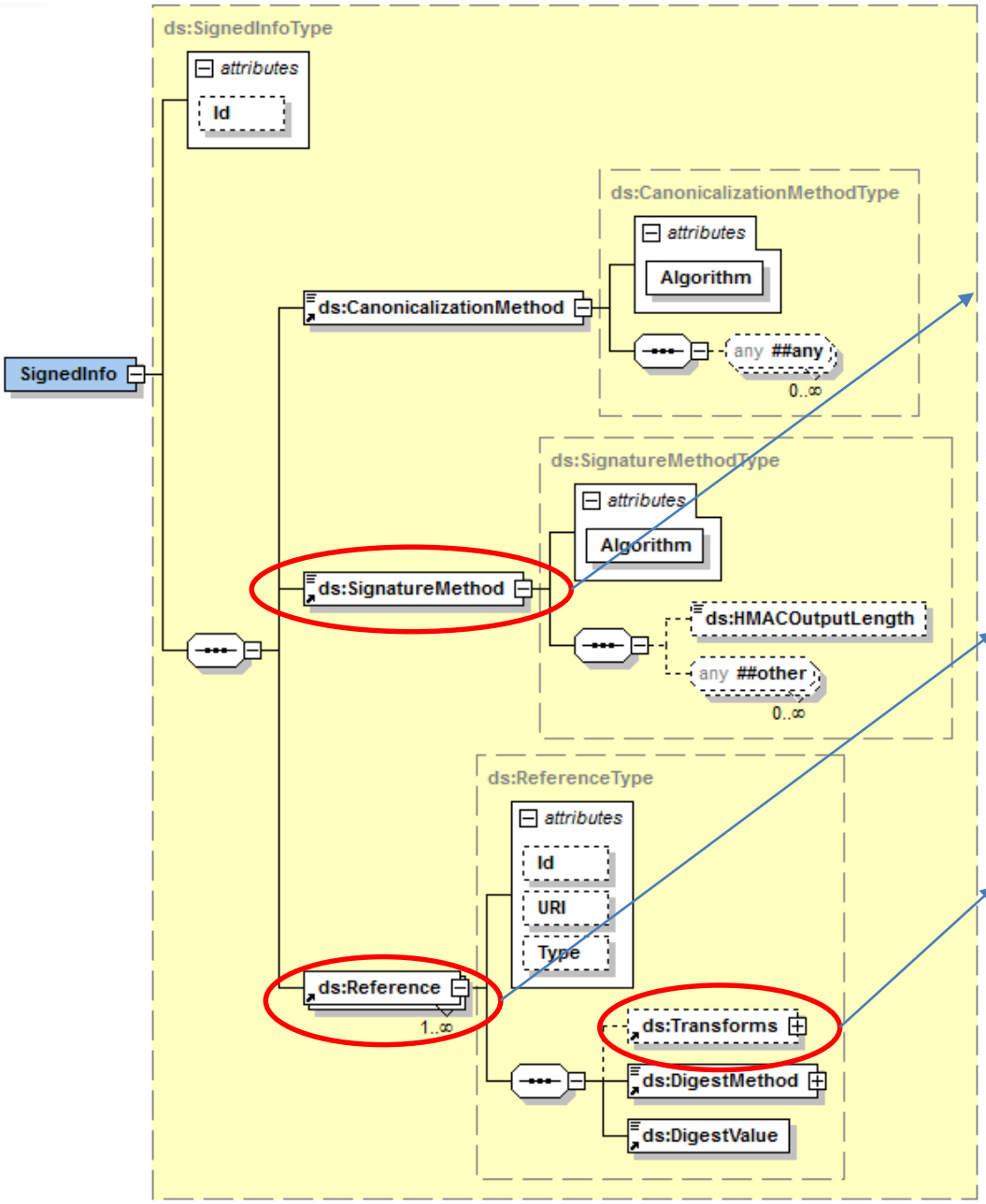
<Signature>

https://www.w3.org/TR/2002/REC-xmldsig-core-20020212/xmldsig-core-schema.xsd

<SignedInfo>
<CanonicalizationMethod />
<SignatureMethod />
<Reference>
<Transforms>
<DigestMethod>
<DigestValue>
</Reference>
</SignedInfo>
<SignatureValue />
<KeyInfo />
<Object />
</Signature>



XML Signature Elements: *SignedInfo*



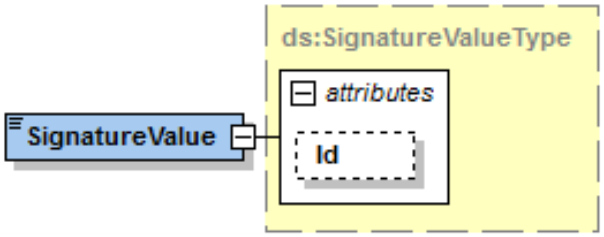
`SignedInfo`:
Information that is really signed

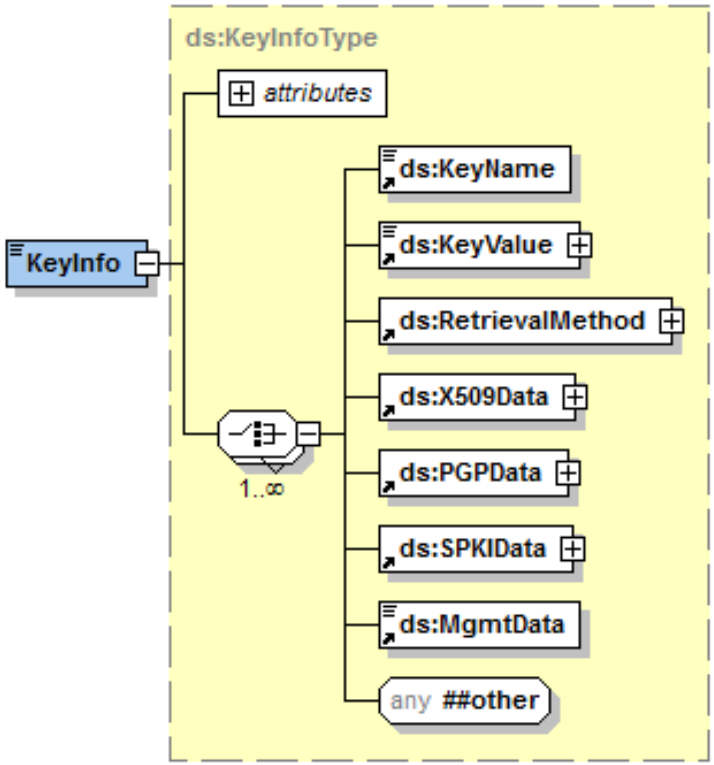
`SignatureMethod`: Algorithm used to obtain the value of the signature. Combination of a digest algorithm and a key algorithm, and possibly others, such as padding.

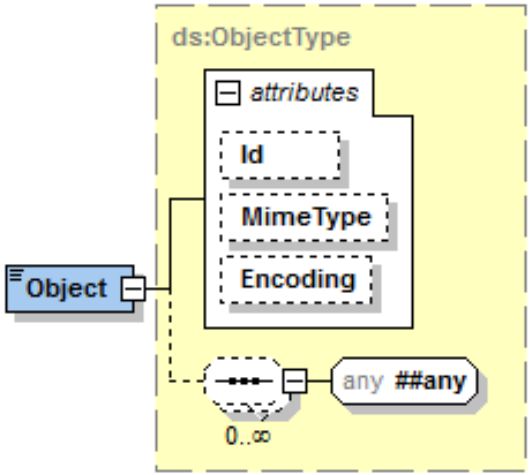
`Reference`: Includes the digest method and calculated value over the data object (URI attribute).

`Transforms`: Ordered list with the processing steps applied to the content before applying the digest algorithm (Ex.: canonicalization, XSLT, XPath, etc.).

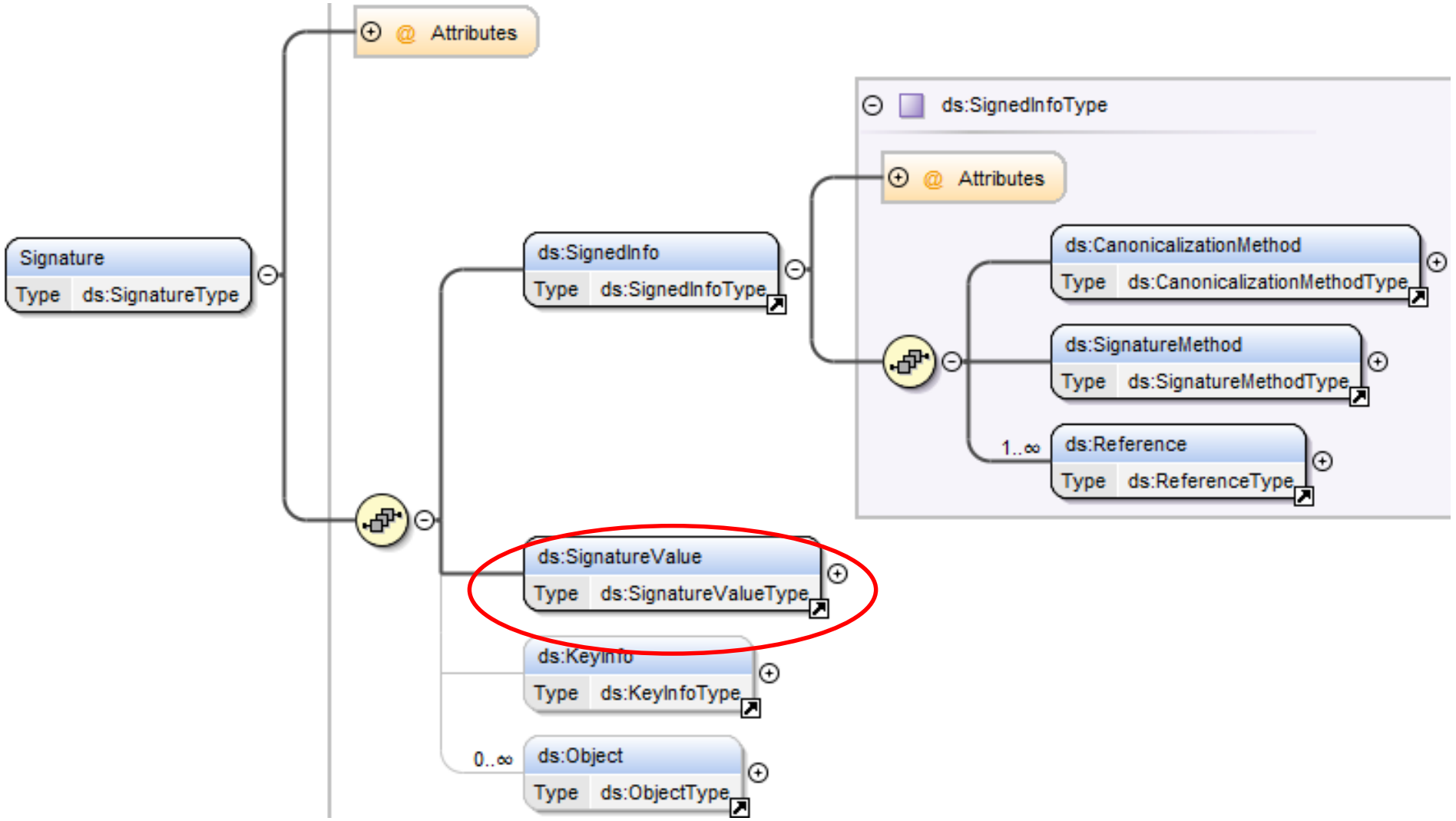
XML Signature Elements: Others







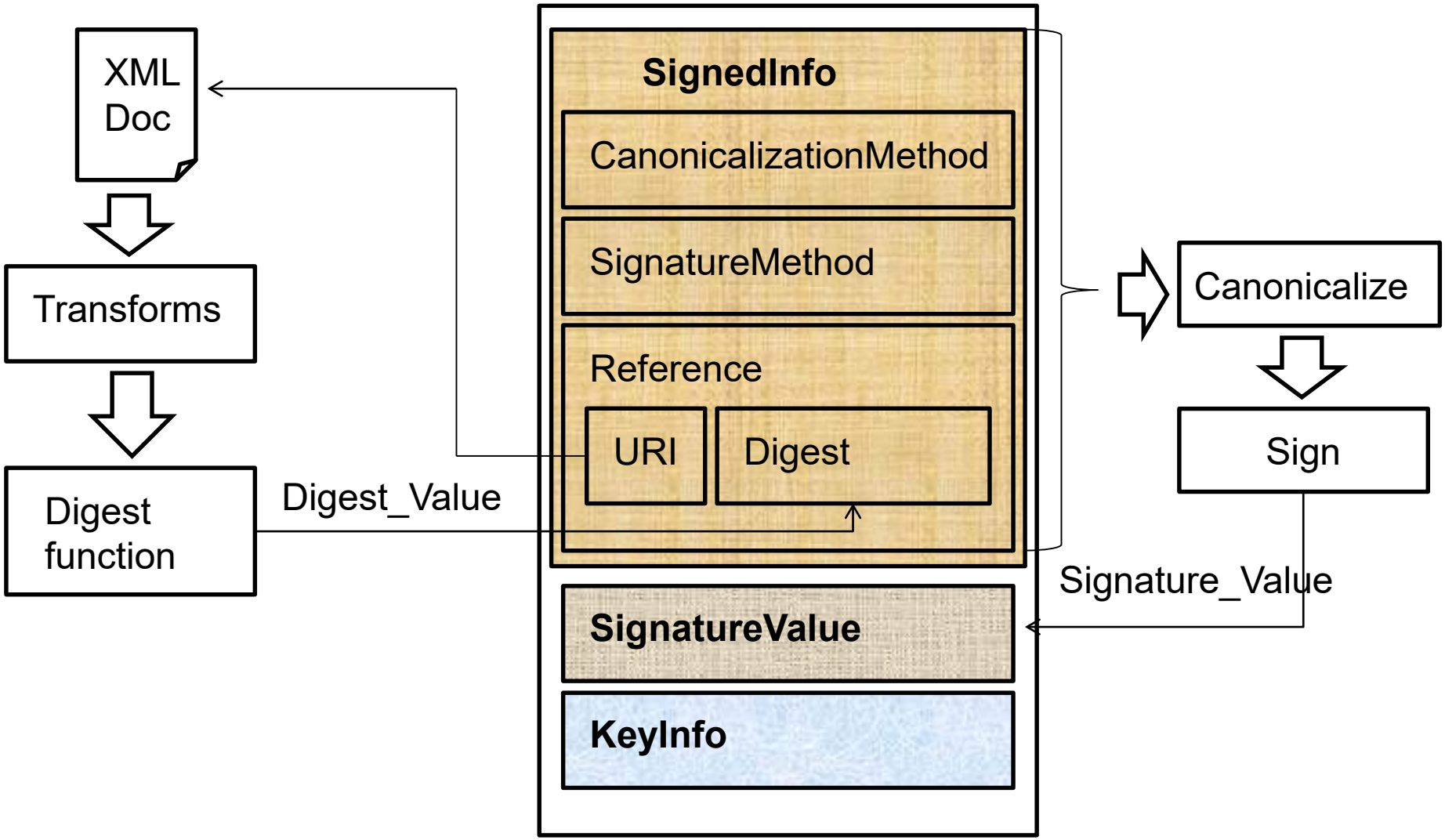
Signature Element structure



Example

```
<Signature Id="MyFirstSignature" xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
    <CanonicalizationMethod Algorithm="http://www.w3.org/2006/12/xml-c14n11"/>
    <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
    <Reference URI="http://www.w3.org/TR/2000/REC-xhtml1-20000126/">
      <Transforms>
        <Transform Algorithm="http://www.w3.org/2006/12/xml-c14n11"/>
      </Transforms>
      <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
      <DigestValue>dGhpcyBpcyBub3QgYSBzaWduYXR1cmUK.../DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue>...</SignatureValue>
  <KeyInfo>
    <KeyValue>
      <DSAKeyValue>
        <P>...</P><Q>...</Q><G>...</G><Y>...</Y>
      </DSAKeyValue>
    </KeyValue>
  </KeyInfo>
</Signature>
```

Model


XML Doc
Transforms
Digest function
Digest_Value
SignedInfo
CanonicalizationMethod
SignatureMethod
Reference
URI
Digest
SignatureValue
KeyInfo
Canonicalize
Sign
Signature_Value