

Secure and Reliable eARC Exchange

Mansour Rezaei Mazinani

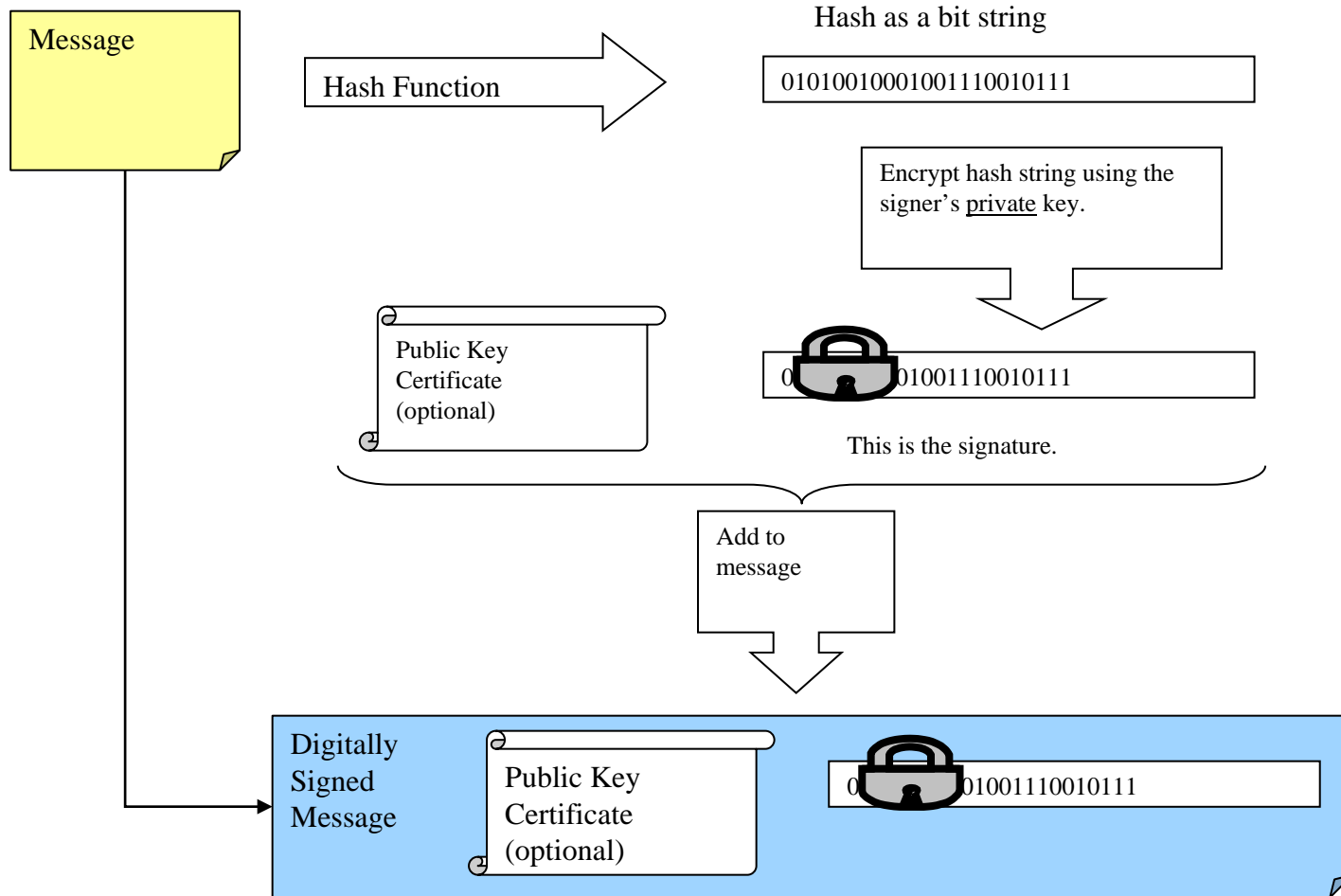
ATA eBusiness Forum
Seattle, May17-19, 2010



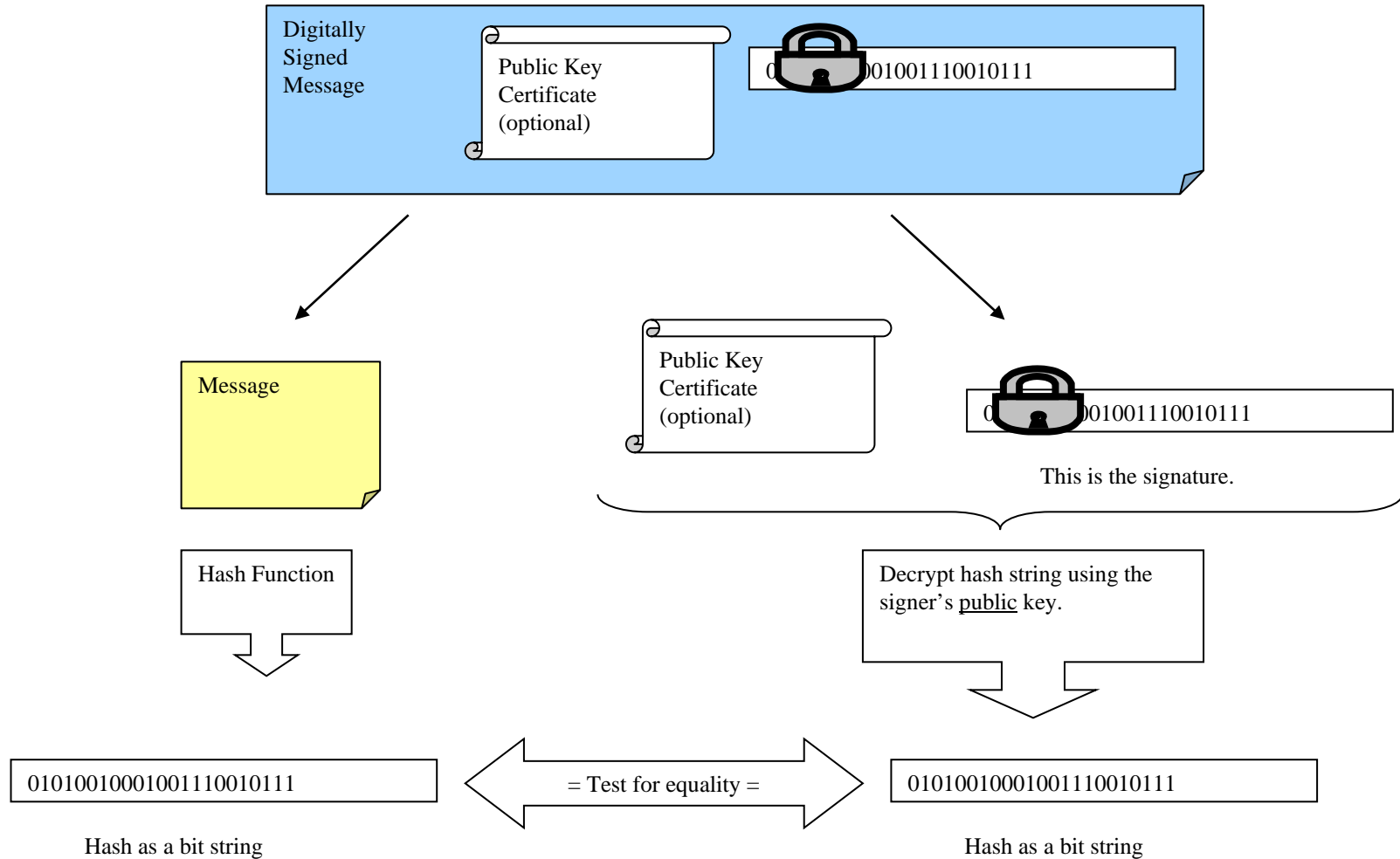
Agenda

- Digital signature & encryption fundamentals
 - What is a digital signature
 - W3C XML digital signature & encryption standards
- Secure and reliable data exchange
 - IATA Type X standard
 - Security extensions
- Composing ATA, IATA and W3C standards to enable reliable and secure eARC exchange
- Conclusions

Fundamentals: digital signature - signing



Fundamentals: digital signature - validation



W3C XML Encryption

W3C XML Encryption specifies syntax and processing for encrypting data and representing the result in XML

The data to be encrypted may be an XML document, an XML element or the content of an XML element

The data to be encrypted is replaced by cipher data after encryption

W3C XML Encryption

An important use is to specify whether the data encrypted is an *element* or *content*.

Cryptographic algorithm used for encryption (optional here since it may be specified elsewhere globally for all *EncryptedData* elements).

Optional keying information URL of key provider or info like order/invoice.

`enc:EncryptedDataType`

`attributes`

`EncryptedData`

`xenc:EncryptionMethod`

`ds:KeyInfo`

`xenc:CipherData`

`xenc:EncryptionProperties`

Element which replaces the encrypted XML.

The encrypted data

W3C XML Encryption: an example

Original Data

```
<?xml version='1.0'?>  
<PaymentInfo xmlns='http://example.org/paymentv2'>  
  <Name>John Smith</Name>
```

```
  <CreditCard Limit='5,000' Currency='USD'>  
    <Number>4019 2445 0277 5567</Number>  
    <Issuer>Example Bank</Issuer>  
    <Expiration>04/02</Expiration>  
  </CreditCard>
```

```
</PaymentInfo>
```

Encrypted Data

```
<?xml version='1.0'?>  
<PaymentInfo xmlns='http://example.org/paymentv2'>  
  <Name>John Smith</Name>
```

```
  <EncryptedData  
    Type='http://www.w3.org/2001/04/xmlenc#Element'  
    xmlns='http://www.w3.org/2001/04/xmlenc#'>  
    <CipherData>  
      <CipherValue>A23B45C56... </CipherValue>  
    </CipherData>  
  </EncryptedData>
```

```
</PaymentInfo>
```

W3C XML Encryption: processing

Encryption - for each data item to be encrypted:

1. Select the encryption algorithm (this may be done once only for all data)
2. Obtain the key for the encryption
3. Encrypt the data
4. Populate the *EncryptedData* element
5. Process the XML document by replacing the encrypted data item with the *EncryptedData* element.

Decryption - for each *EncryptedData* element:

1. Determine the encryption algorithm (this may be done once only for all elements)
2. Decrypt the data
3. Replace the *EncryptedData* with the decrypted data

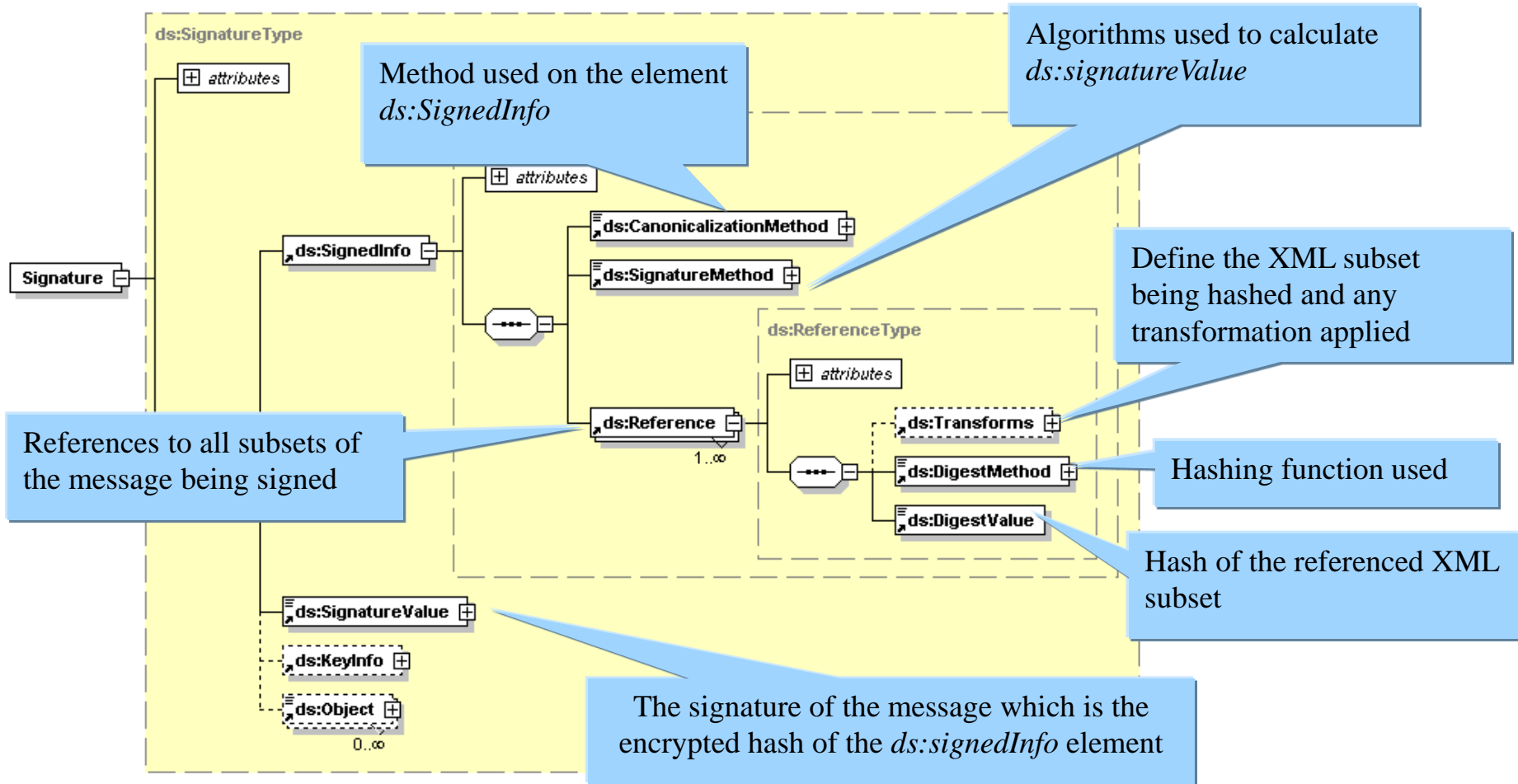
W3C XML Digital Signature

W3C XML Signature defines XML syntax and processing rules for creating and representing digital signatures of XML messages

Due to the nature of XML, the message usually needs to be canonicalized which transforms the source XML into a standardized XML form

Canonicalization addresses the syntactical changes to XML caused by parsing and processing techniques which will invalidate the digital signature

W3C XML Digital Signature



W3C XML Digital Signature: an 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/2000/09/xmldsig#rsa-sha1"/>
    <Reference URI="http://www.w3.org/TR/2000/REC-xhtml1-2000126/">
      <Transforms>
        <Transform Algorithm="http://www.w3.org/2006/12/xml-c14n11"/>
      </Transforms>
      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <DigestValue>dGhpcyBpcyBub3QgYSBzaWduYXR1cmUK.../DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue>Ed54KL...</SignatureValue>
</Signature>
```

Method used on the element
ds:SignedInfo

Signing algorithm on
the element
ds:SignedInfo

Document to be signed
(URI or Xpath or..)

Canonicalization of
document

Hashing algorithm

Digest of the document to
be signed

Signature of the document

W3C XML Digital Signature: processing

Signing procedure

1. Select a portion of the message (termed a *ds:Reference*) to sign
2. Standardize (i.e. canonicalize) the selection using a W3C canonicalization transform
3. Hash the output of the canonicalization
4. Place the hash of step 3 in the W3C *ds:SignedInfo/ds:Reference/ds:DigestValue* element
5. Specify the algorithm (*ds:DigestMethod*) and transforms (*ds:Transforms*)
6. Repeat steps 1 to 5, for each portion of the message to sign

Final steps to create the signature

7. Specify the signature algorithm (*ds:SignatureMethod*) and canonicalization method (*ds:CanonicalizationMethod*) in the *ds:SignedInfo* element
8. Canonicalize the *ds:SignedInfo* element
9. Apply the signature algorithm to the output of the canonicalization method (step 8)
10. Place the result of the signature algorithm (step 9) in the W3C *ds:SignatureValue* element (the signature of the message)

W3C XML Digital Signature: processing

Validation procedure

1. Standardize (i.e. canonicalize) the *ds:SignedInfo*.

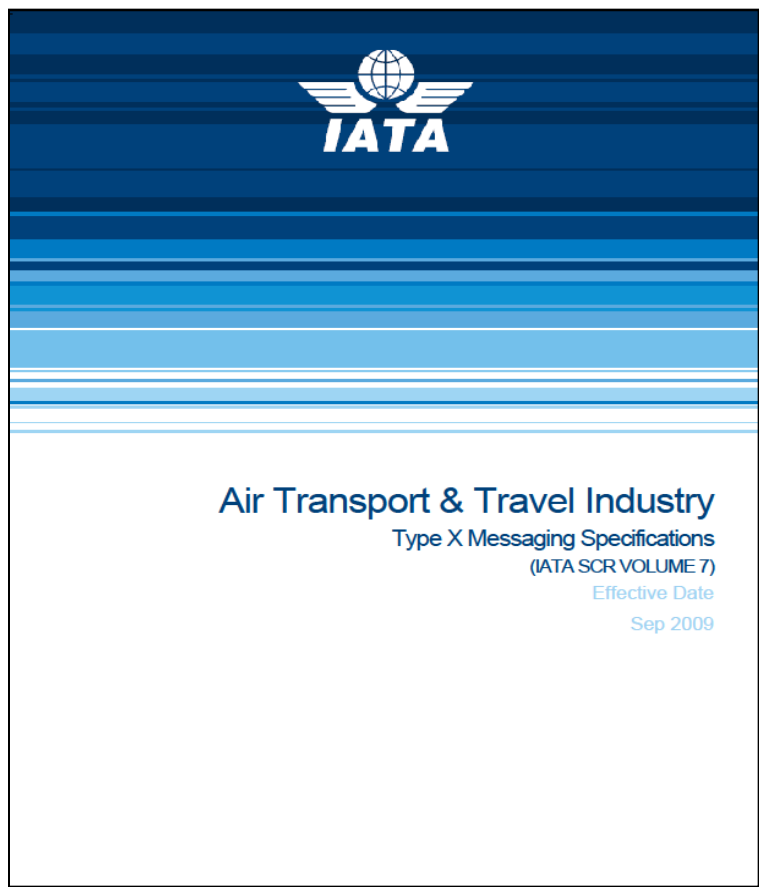
For each *ds:SignedInfo* /*ds:Reference* element::

1. Obtain the data using *ds:Transforms*
2. Hash the data using *ds:DigestMethod*
3. Compare the resulting digest with *ds:DigestValue*; if they are not identical, validation fails

Final steps validate the signature:

1. Decrypt the *ds:SignatureValue*, obtaining the digest
2. Hash the canonicalized *ds:SignedInfo* element
3. If the digests are not identical, validation fails

Available Documentation – Type X Specification



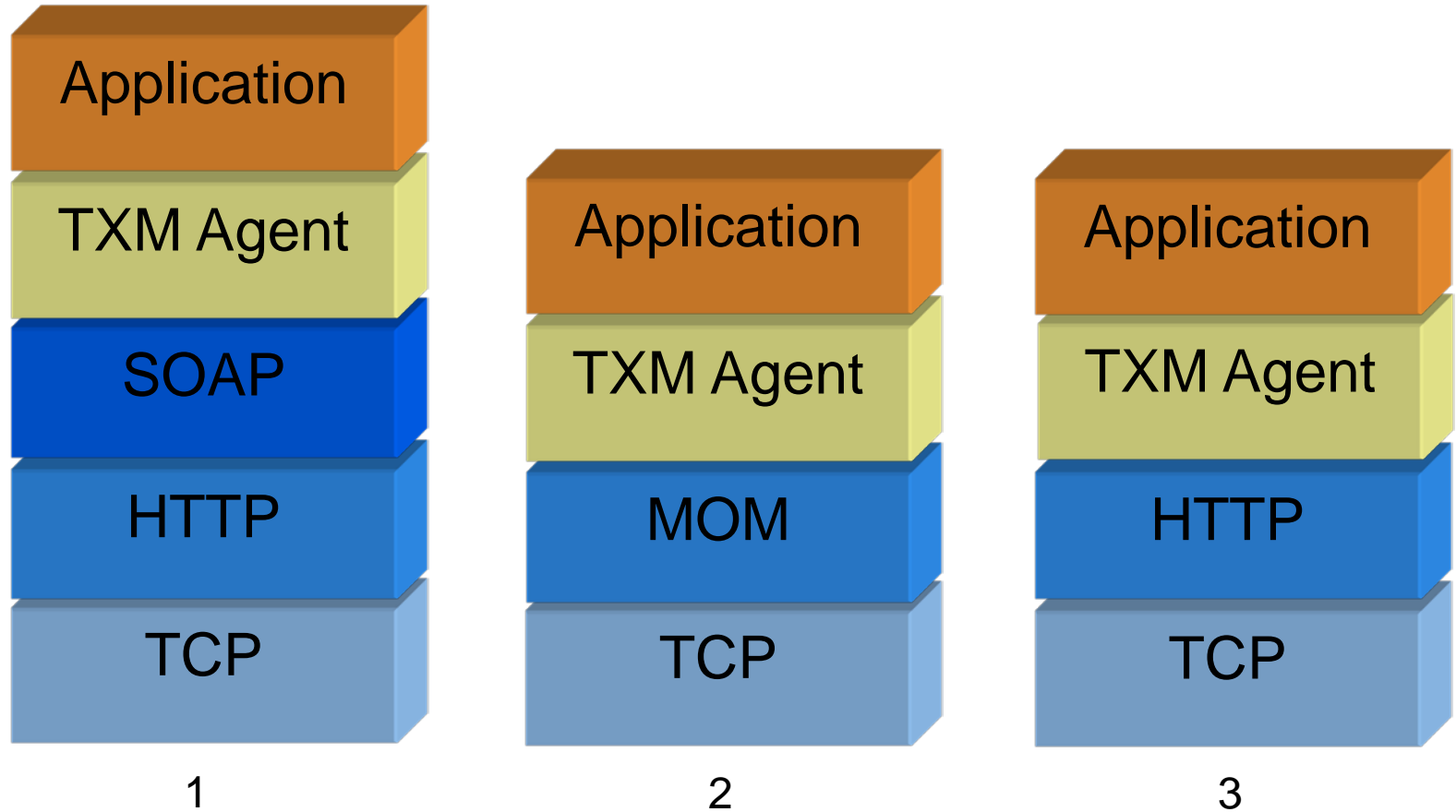
- ✓ Specifies Type X Message (TXM) structure and necessary headers
- ✓ Specifies TXM – transport mapping
- ✓ Specifies security extensions
- ✓ Specifies reliability protocol
- ✓ Specifies Session Management
- ✓ Specifies all Type X related Schema
- ✓ Initial implementations and testing completed

Type X is published as IATA Standard SCR Volume 7 in September 2009

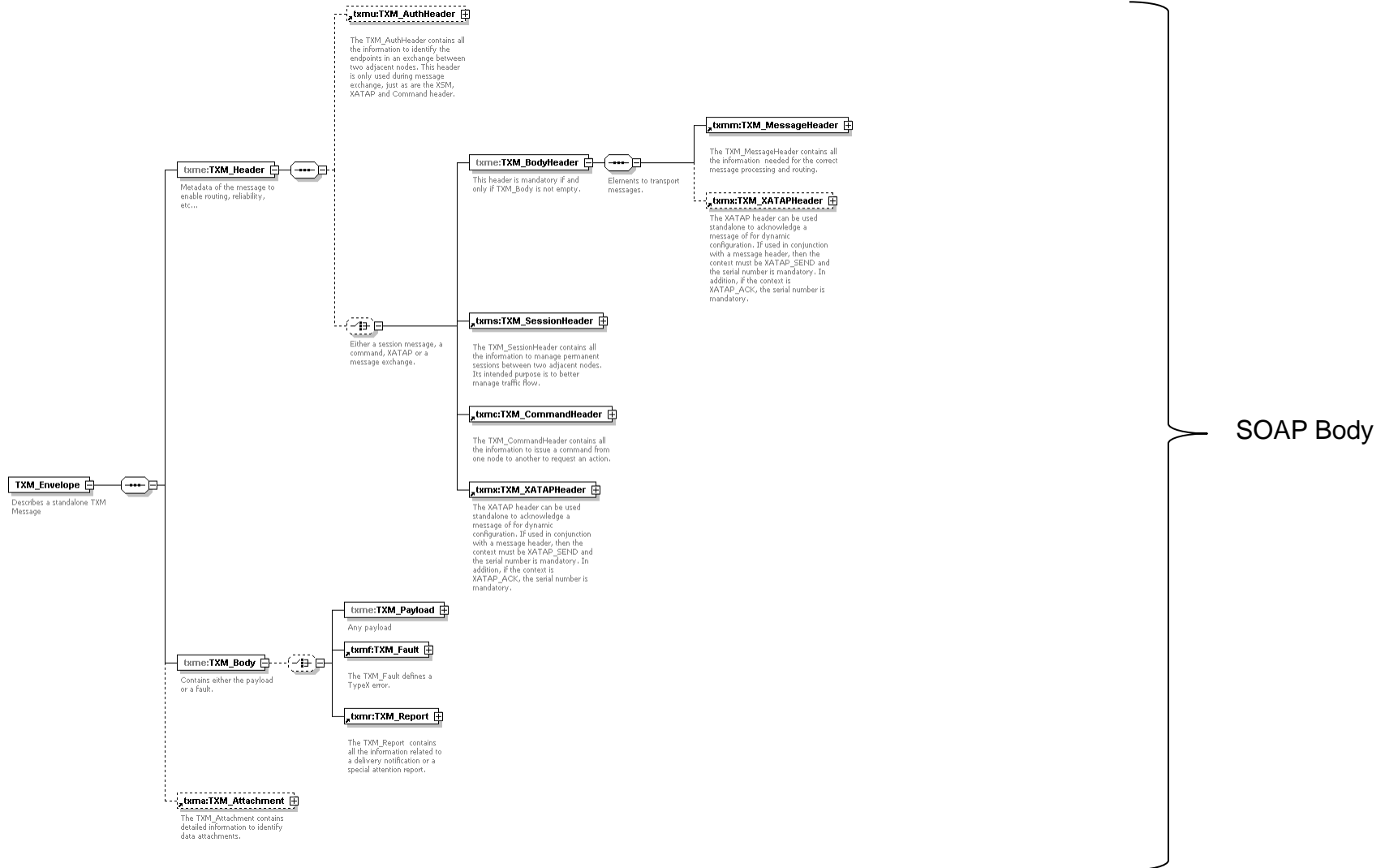
Type X - Properties

- Compatible with air transport business practices
- Support for all standard ATN message exchange patterns
- Full message assurance support for application to application delivery
- Permits detection of duplicate messages
- Permits messaging ordering
- Session management
- SOAP, JMS and HTTP bindings
- Security options
- Messaging priority
- Grouping of messages
- Multicast (one message to many recipients)
- End-to-End addressing (originator to recipient)
- End-to-End delivery notification (ultimate receiver or recipient to originator)
- Service command messages
- Openness by providing free fields
- Support for attachments

Recommended Type X Stacks



Type X Message to SOAP Mapping



Type X- security

- TypeX security capabilities include:
 - ✓ Content Integrity
 - ✓ Confidentiality
 - ✓ Authentication
 - ✓ Non repudiation
- Functionality enabled by the use of PKI for encryption and digital signature mainly by the end users
- Uses standard OASIS Web Services Security framework defined as SOAP extensions i.e. WS-Security with W3C XML Encryption & XML Digital Signature, WS-Trust, WS-Federation, WS-SecureConversation, WS-SecurityPolicy, SAML
- Security extension using W3C encryption and digital signature
- Implementation guidelines for TypeX is part of the work group documents

Type X Security: SOAP Message Level Security

OASIS WS-* stack of security standards. The important standards are:

- **WS-Security:**

- A protocol for securing web service message exchanges by providing security token transport, message integrity and message confidentiality
- Core specification for service security, relying on two other W3C specifications: XML Encryption and XML Digital Signatures

- **WS-Trust:**

- Defines mechanism to assess the validity of the security credentials of the exchanging parties
- Defines a means of negotiating security credentials among partners within different trust domains

Type X Security: SOAP Message Level Security

- **WS-Federation:**

- Builds on WS-Trust to provide brokerage and federation of trust claims

- **WS-SecureConversation:**

- Builds on WS-Trust to establish security context valid for the life of an exchange session

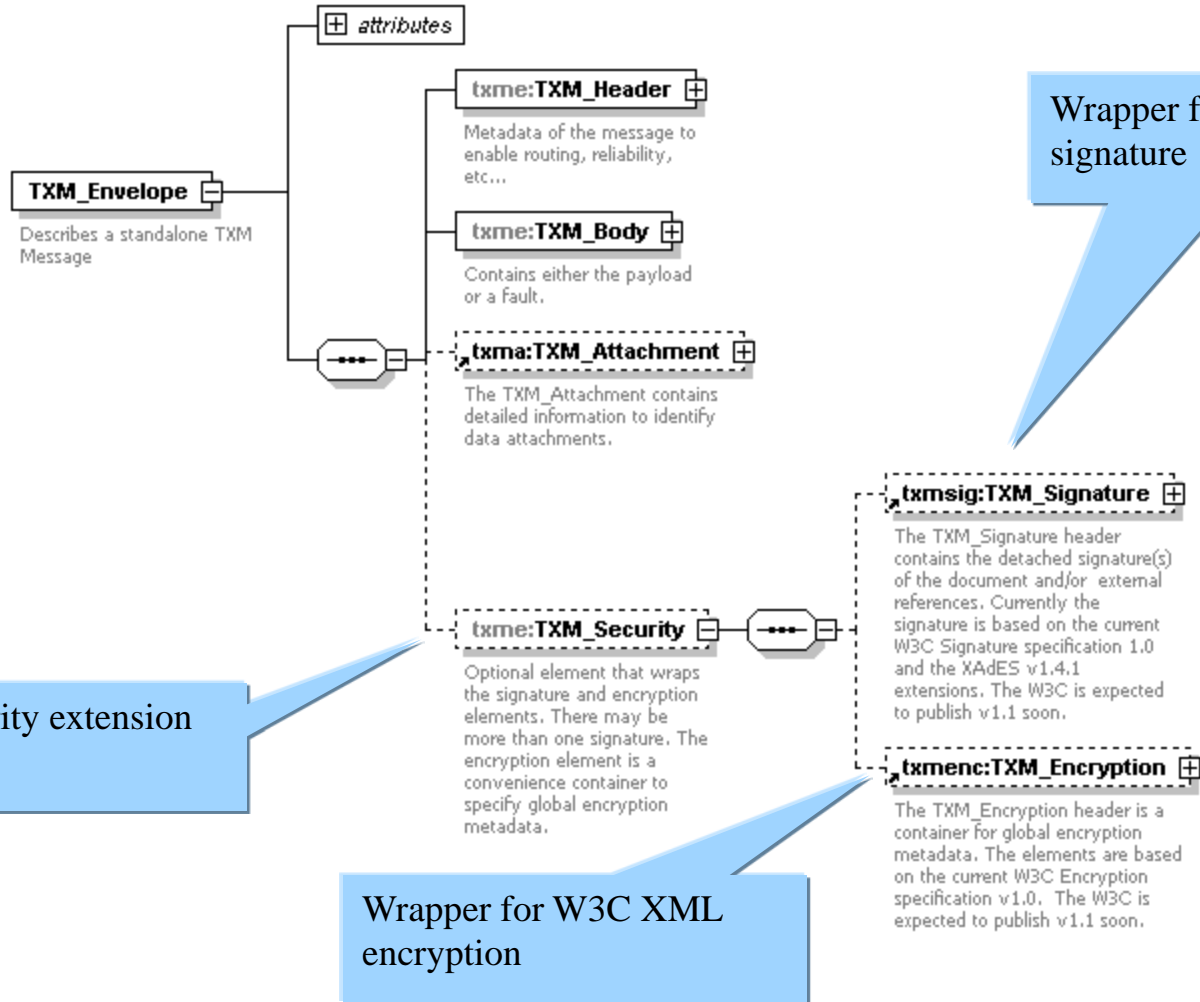
- **WS-SecurePolicy:**

- Leverages the WS-Policy framework that enables a web service to express as well as communicate constraints, requirements and properties, as policy assertions (XML sentences)
- In a security context, these assertions are *security policy assertions* that express how a message is secured

Type X Security Extension: definition & purpose

- Provide a standard framework to secure Type X messaging at message level
- Provide an interoperable mechanism to enable secure TypeX message exchanges regardless of the underlying transport protocol
- Binding to W3C standards for digital signature and encryption
- Binding defined in terms of application processing rules
- Guiding principles:
 - Adherence to existing standards: use of the W3C recommendations for signature and encryption for interoperability
 - Simplicity: allow what is necessary
 - Independence: TypeX Security binding maintains the standalone aspect of TypeX
 - Evolutivity: loose coupling principle for fast and simple evolution

Type X Security Extension

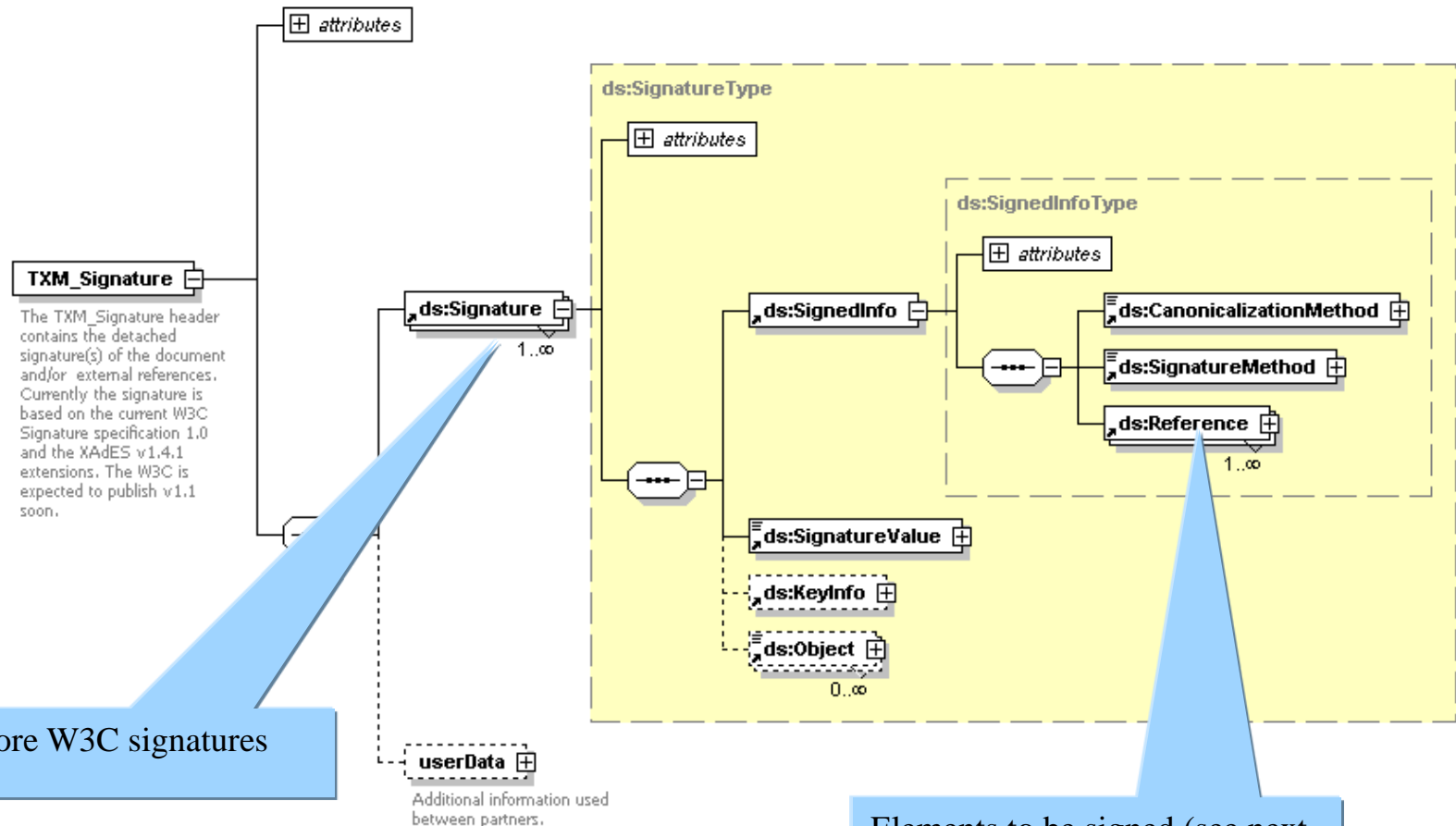


Type X Security Extension: signature characteristics

Proposed *TXM_Signature* binding has the following characteristics:

- Use the W3C XML Signature standard for interoperability
- Employs a minimal set of recommended transforms: XPATH Filter 2.0 and Canonicalization
- Selection of nodes to be signed is achieved with XPATH Filter 2.0 so that an application can simply determine what is signed
- Obeys best practices
- Achieves simplicity and usability by imposing restrictions
- Facilitates interoperability and adoption

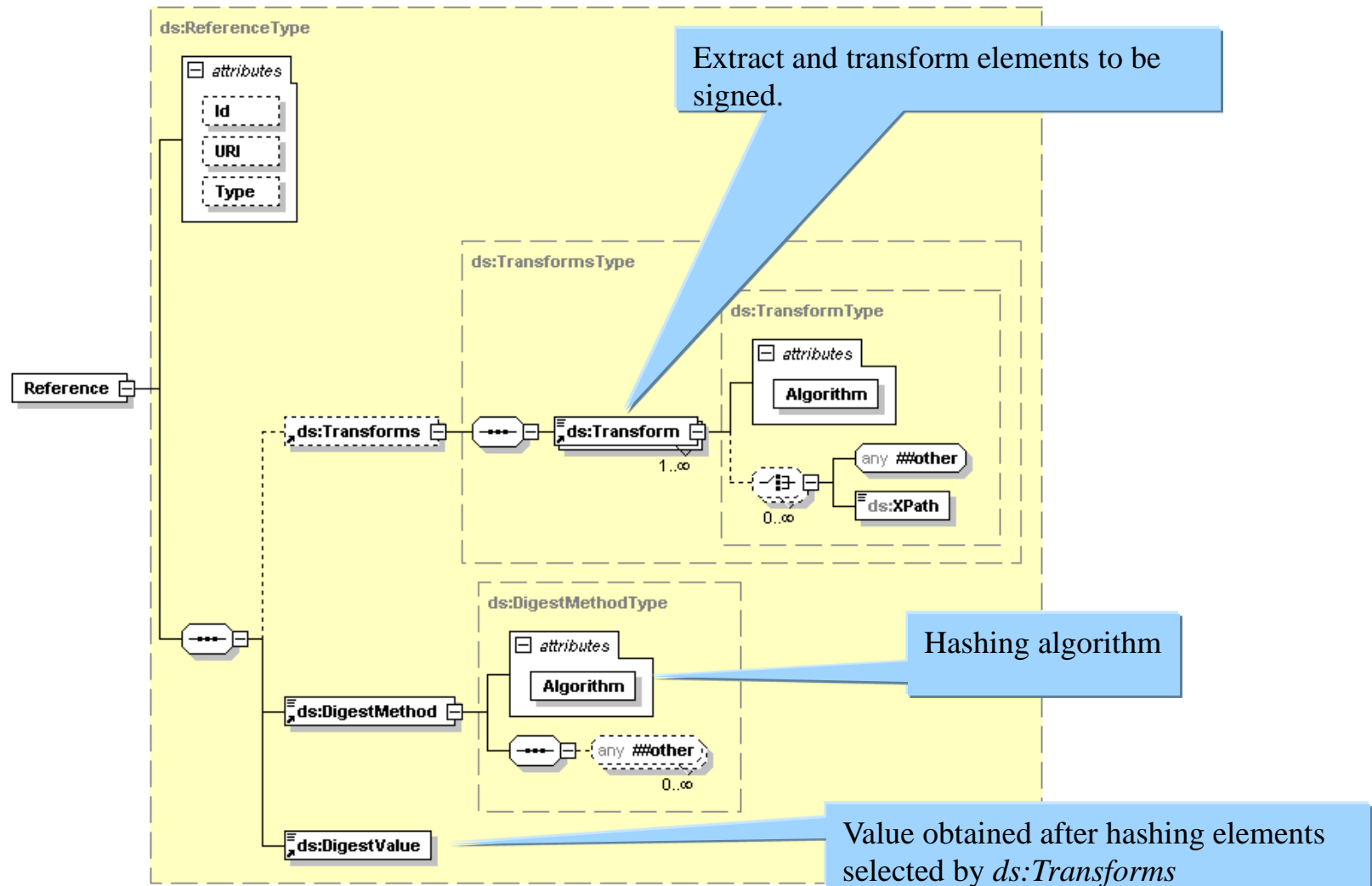
Type X Security Extension: XML digital signature



One or more W3C signatures

Elements to be signed (see next slide for details)

Type X Security Extension: XML digital signature



Type X Security Extension: digital signature design

The elements of TypeX envelope to be signed are as follows:

- */txme:TXM_Envelope/txme:TXM_Header/txme:TXM_BodyHeader/txmm:TXM_MessageHeader*
- */txme:TXM_Envelope/txma:TXM_Body/txmr:TXM_Report*
- */txme:TXM_Envelope/txma:TXM_Body/txmf:TXM_Fault*
- */txme:TXM_Envelope/txme:TXM_Body/txme:TXM_Payload*
- */txme:TXM_Envelope/txma:TXM_Attachment*

All other elements are not signed

- Thus, there is always at least two internally signed references (i.e. the *txme:TXM_Body* elements and the *txmm:TXM_MessageHeader*); if the optional *txma:TXM_Attachment* is present, then it must also be signed

Type X Security Extension: digital signature processing

- The generation of a signature is the same as W3C digital signature processing. The only additional step is to place the resulting *dsig:Signature* element in the *txme:TXM_Signature* element

- The validation of a signature is the same as W3C validation processing

Type X Security Extension

Digital Signature Processing

1. **Create the dsig:SignedInfo element containing all referenced node sets (i.e. portions of the document), calculating the digest for each reference**
2. **Calculate the dsig:SignatureValue by :**
 - 2.1 **apply the canonicalization transform to dsig:SignedInfo**
 - 2.2 **calculate the digest of the result of 2.1**
 - 2.3 **encrypt the result of 2.2**
3. **Construct the dsig:Signature element**
4. **Place the dsig:Signature in the txme:TXM_Signature element**

Validation Processing

1. **Validate the key of the signer, i.e. authenticate and establish trust.**
2. **Validate the dsig:SignedInfo**
3. **Validate the dsig:SignatureValue**

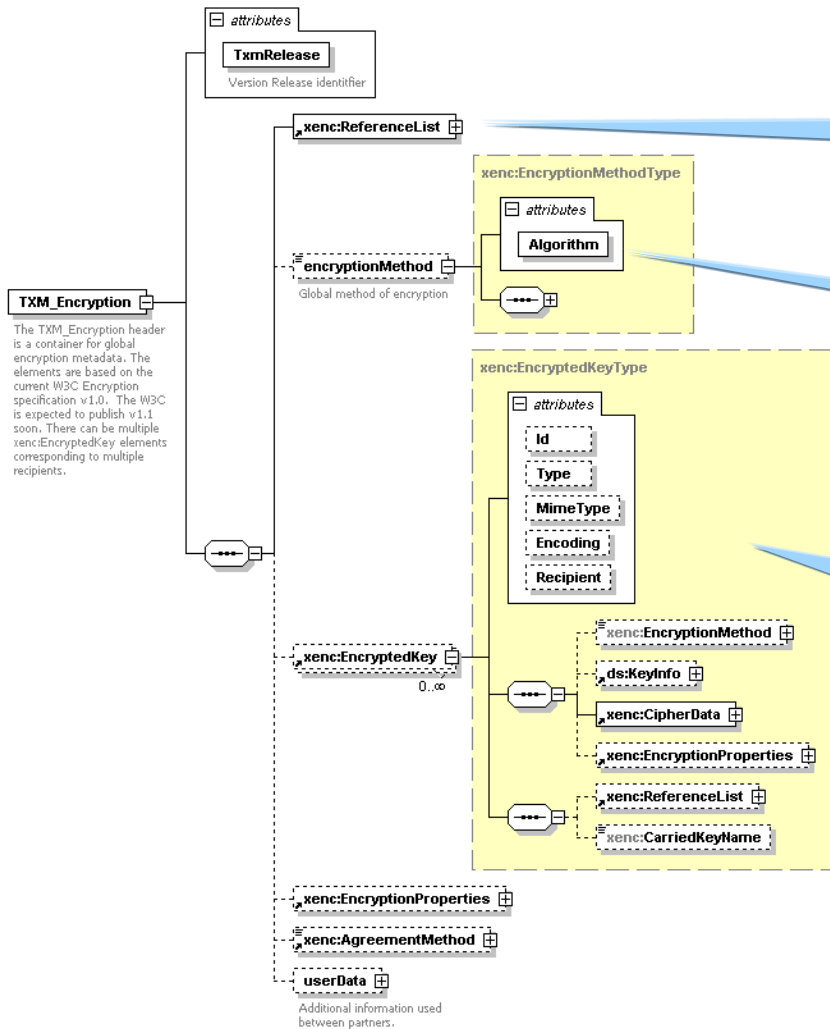
Type X Security Extension: encryption characteristics

- The encryption processing is the same as in W3C XML encryption specification

The characteristics of TypeX encryption:

- Composes with the W3C XML Encryption standard
- Achieves simplicity and usability by imposing restrictions on how the envelope is encrypted
- Facilitates interoperability and adoption

Type X Security Extension: XML encryption



List of Encrypted data elements

Data Encryption algorithm

Encrypted key data : one element per recipient if using each recipient's public key; otherwise one element if encrypting with a shared symmetric key.

Type X Security Extension: encryption design

➤ The elements of a TypeX envelope that may be encrypted are listed here; all other elements are not encrypted.

- */txme:TXM_Envelope/txme:TXM_Header/txme:TXM_BodyHeader/txmm:TXM_MessageHeader*
- */txme:TXM_Envelope/txme:TXM_Body/txme:TXM_Payload*
- */txme:TXM_Envelope/txme:TXM_Body/txmr:TXM_Report*
- */txme:TXM_Envelope/txme:TXM_Body/txmf:TXM_Fault*
- */txme:TXM_Envelope/txma:TXM_Attachment*
- */txme:TXM_Envelope/txme:TXM_Security/txmsig:TXM_signature*

eARC – FAA Form 8130-3 from electronic form to XML schema

1. Approving National Aviation Authority/Country: FAA/United States		2. AUTHORIZED RELEASE CERTIFICATE FAA Form 8130-3, AIRWORTHINESS APPROVAL TAG				3. Form Tracking Number: AP54321	
4. Organization Name and Address: Anyone's Aviation, 1104 Wing Avenue, Anyplace, TX 72212 (PC 234)					5. Work Order/Contract/Invoice Number: WO 99987		
6. Item:	7. Description:	8. Part Number:	9. Eligibility: *	10. Quantity:	11. Serial Batch Number:	12. Status/Work:	
1	Flap Track	B9876-1	N/A	8	N/A	PROTOTYPE	
13. Remarks: Detail part conformity for FAA Project AP54321, dated Feb 10 2008, Drawing No. 12345-001, Revision G1, dated Oct 1 2007 requested. 1. Request for Conformity FAA 8120.10, #06-09222, dated Feb 19 2008 reviewed. 2. FAA 8130-9, Statement of Conformity, dated May 3 2007 provided, reviewed, and attached. 3. Part No. B9876-1 Flap Track (8 ea.), inspected to engineering to include Drawing No. 12345-001, Revision G1, dated Oct 1 2007. DEVIATION: 8 ea. Flap Tracks, Part No. B9876-1 holes should be ".250 +/- .005." Holes are oversized by ".020." DER Disposition: Oversized holes does not affect static testing and parts can be used as is per DER-888002-SW, A. Engineer, dated Feb 11 2008.							
14. Certifies the items identified above were manufactured in conformity to: <input type="checkbox"/> Approved design data and are in a condition for safe operation. <input checked="" type="checkbox"/> Non-approved design data specified in Block 13.				19. <input type="checkbox"/> 14 CFR 43.9 Return to Service <input type="checkbox"/> Other regulation specified in Block 13 Certifies that unless otherwise specified in Block 13, the work identified in Block 12 and described in Block 13 was accomplished in accordance with Title 14, Code of Federal Regulations, part 43 and in respect to that work, the items are approved for return to service.			
15. Authorized Signature: <i>A. Inspector</i>		16. Approval/Authorization No.: DARF-1234567-SW		20. Authorized Signature:		21. Approval Certificate No.:	
17. Name (Typed or Printed): A. Inspector		18. Date (m d y): Mar 3 2008		22. Name (Typed or Printed):		23. Date (m d y):	
User/Installer Responsibilities							
It is important to understand that the existence of this document alone does not automatically constitute authority to install the part/component/assembly. Where the user/installer performs work in accordance with the national regulations of an airworthiness authority different than the airworthiness authority of the country specified in Block 1, it is essential that the user/installer ensures that his/her airworthiness authority accepts parts/components/assemblies from the airworthiness authority of the country specified in Block 1. Statements in Blocks 14 and 19 do not constitute installation certification. In all cases, aircraft maintenance records must contain an installation certification issued in accordance with the national regulations by the user/installer before the aircraft may be flown.							

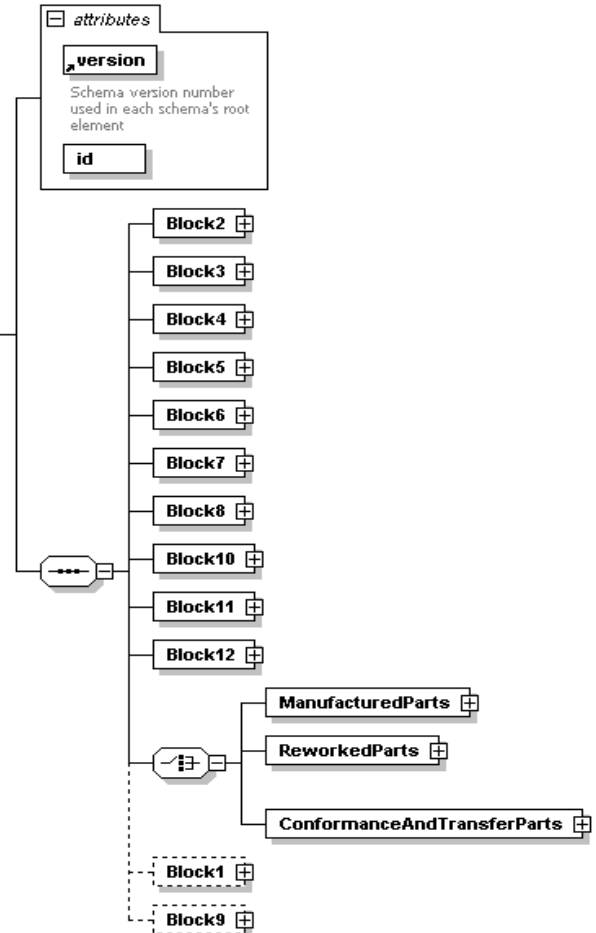
FAA Form 8130-3 (6-01)

*Installer must cross-check eligibility with applicable technical data.

NSN: 0052-00-012-9905

ATA_PartCertificationForm

Corresponds to FAA Form 8130-3, EASA Form 1, TCCA 24-0078, C of C, Transfer Document, etc.



Composing ATA, IATA and W3C standards to enable reliable and secure eARC exchange

- ATA Spec 2000 Chapter 16
 - Provides specific schema as standard format for the exchange of electronic form 8130-3 for products, parts, and appliances
 - Provides the minimum requirements for digital security and data exchange
- Use of data encryption is desired
- Choice of communication protocol is left up to trading partners to decide. However, users should select a protocol that provides features for reliable messaging and non-repudiation including:
 - Unique message ID
 - Acknowledgement of message receipt
 - Time of transmission and receipt of messages

Enabling reliable and secure eARC exchange

eARC as payload & digital signature - example

Since XML is verbose, we present the payload and TXM_Signature element for TypeX envelope in snippets.

```
<txmsig:TXM_Signature TxmRelease="TXM2009A0"
  xmlns:txmsig="http://www.iata.org/txm/sig">
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2006/12/xml-c14n11"/>
      <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
      <ds:Reference URI="">
        <ds:Transforms xmlns:ds-xpath="http://www.w3.org/2002/06/xmlds-filter2">
          <ds:Transform Algorithm="http://www.w3.org/2002/06/xmlds-filter2">
            <ds-xpath:XPath Filter="intersect">
              /txme:TXM_Envelope/txme:TXM_Header/txme:TXM_BodyHeader/txmm:TXM_MessageHeader
            </ds-xpath:XPath>
```

Select (intersect) the
entire MessageHeader

Enabling reliable and secure eARC exchange

eARC as payload & digital signature - example continued

```
<ds-xpath:XPath Filter="subtract">
/txme:TXM_Envelope/txme:TXM_Header/txme:TXM_BodyHeader/txmm:TXM_MessageHeader/Destination/RecipientInformation/ResponsibilityFlag
</ds-xpath:XPath>
<ds-xpath:XPath Filter="subtract">
/txme:TXM_Envelope/txme:TXM_Header/txme:TXM_BodyHeader/txmm:TXM_MessageHeader/Destination/NodeTrace
</ds-xpath:XPath>
<ds-xpath:XPath Filter="subtract">
/txme:TXM_Envelope/txme:TXM_Header/txme:TXM_BodyHeader/txmm:TXM_MessageHeader/Information/PossibleDuplicateMessage
</ds-xpath:XPath>
<ds-xpath:XPath Filter="subtract">
/txme:TXM_Envelope/txme:TXM_Header/txme:TXM_BodyHeader/txmm:TXM_MessageHeader/Information/MessageId
</ds-xpath:XPath>
</ds:Transform>
```

Discard (subtract) all XML tags that must not be signed

Enabling reliable and secure eARC exchange

eARC as payload & digital signature - example continued

```
<ds:Transform Algorithm="http://www.w3.org/2002/06/xmlds-filter2">
  <ds-xpath:XPath Filter="intersect">
    /txme:TXM_Envelope/txme:TXM_Body/txme:TXM_Payload</ds-xpath:XPath>
  </ds:Transform>
  <ds:Transform Algorithm="http://www.w3.org/2006/12/xml-c14n11"/>
</ds:Transforms>
  <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
  <ds:DigestValue>UjBsR09EbGhGUX... </ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
  <ds:SignatureValue>UjBsR09EbG ...</ds:SignatureValue>
</ds:Signature>
</txmsig:TXM_Signature>
```



Select (intersect) the payload.

Conclusion

- Spec2000 specification recommends use of reliable messaging for related data exchanges, including the use of W3C security
- IATA Type X provides a rich set of features and transport independence. It composes with W3C XML Signature and Encryption standards for security extensions and simplifies greatly digital signing of Type X envelopes by proposing rules on how signing is done
- Composing IATA Type X, ATA CH16 and W3C DS specifications meet all requirements for reliability and security, and enable digitally signed eARC interoperability among business partners

Thank You

Mansour.rezaei-mazinani@sita.aero

