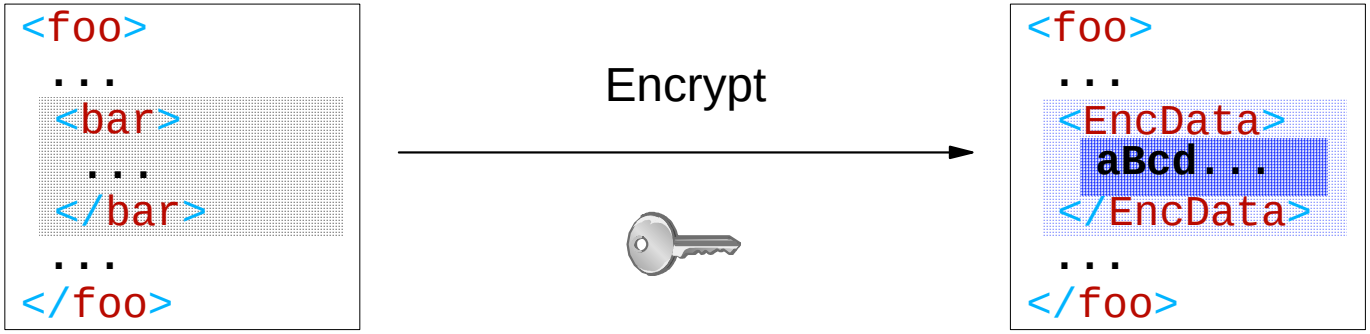
XML Encryption Implementation Experience

Takeshi Imamura
Tokyo Research Laboratory
IBM Research
imamu@jp.ibm.com

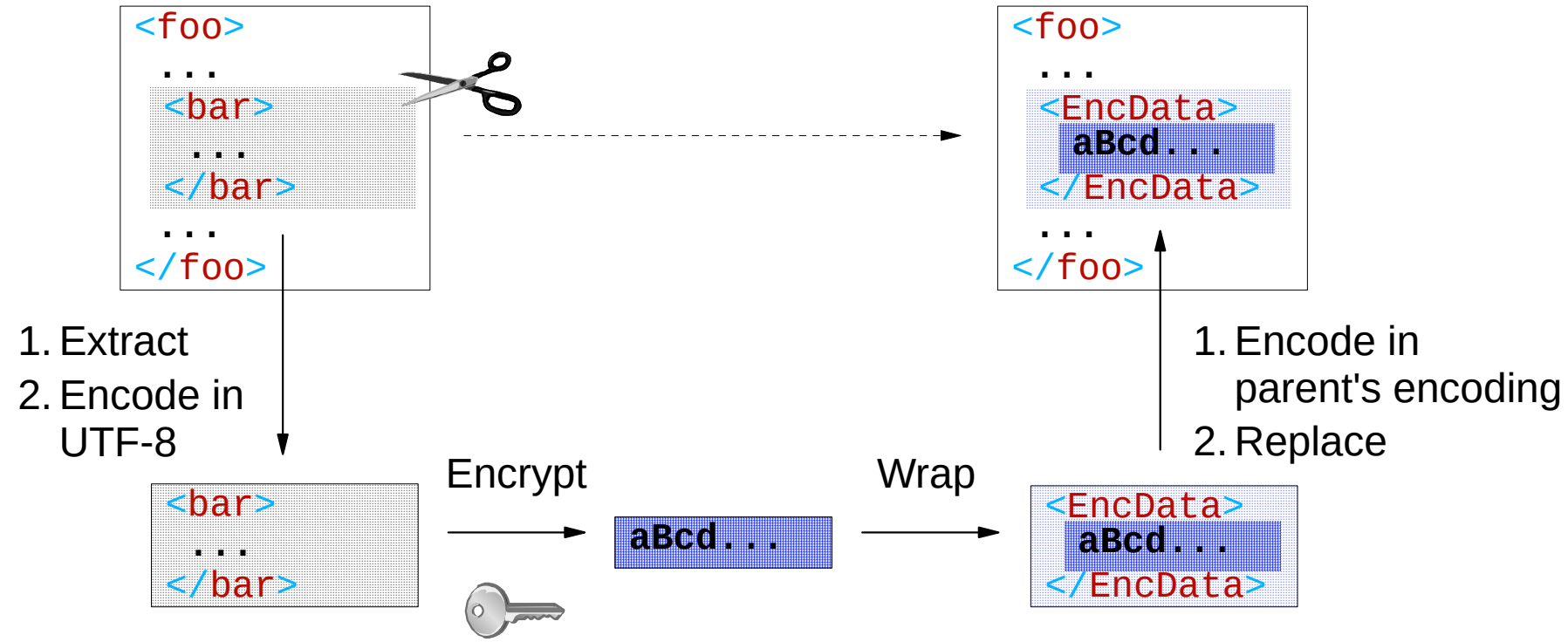
Design (1/4)

- Steps by the spec



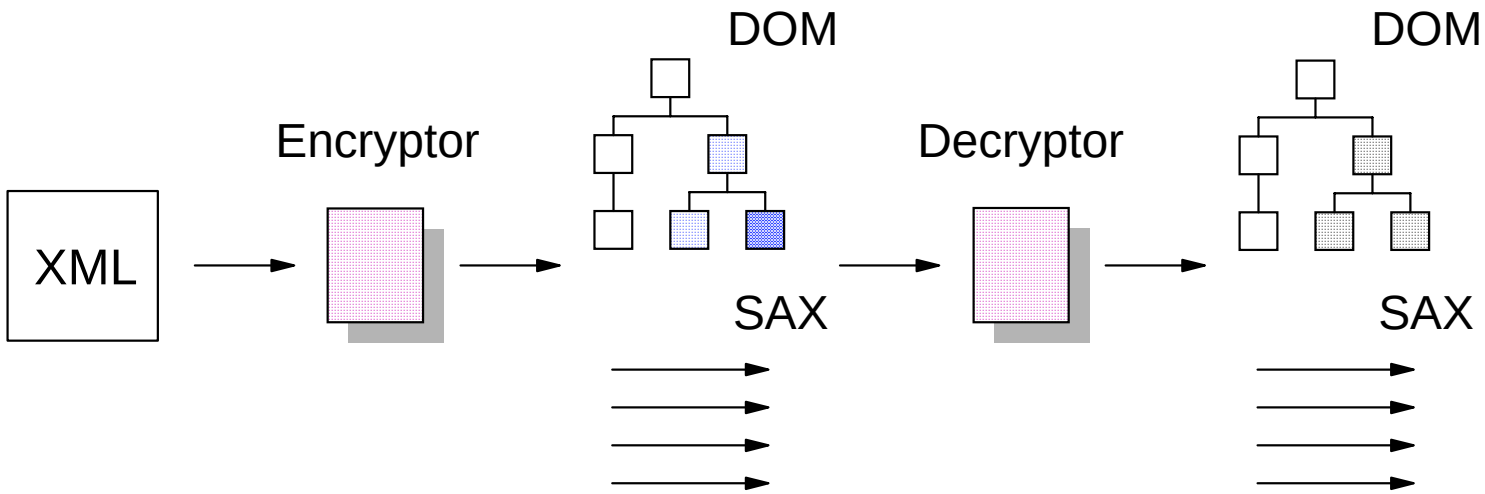
Design (1/4)

- Steps by the spec



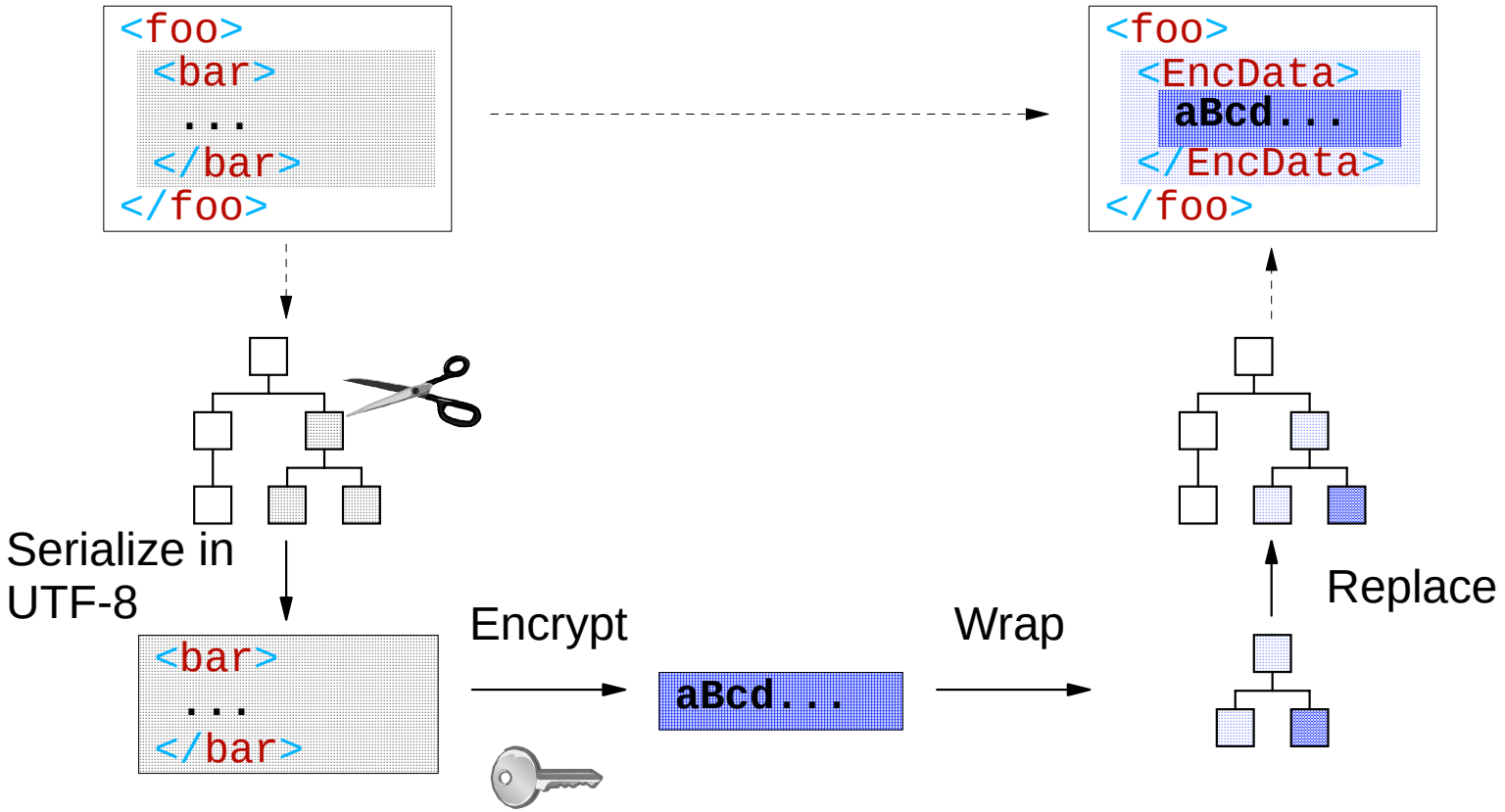
Design (2/4)

- Approaches
  - Application-level implementation
    - DOM-based, SAX-based, ...
  - Parser-level implementation
    - XNI (Xerces Native Interface) -based, ...



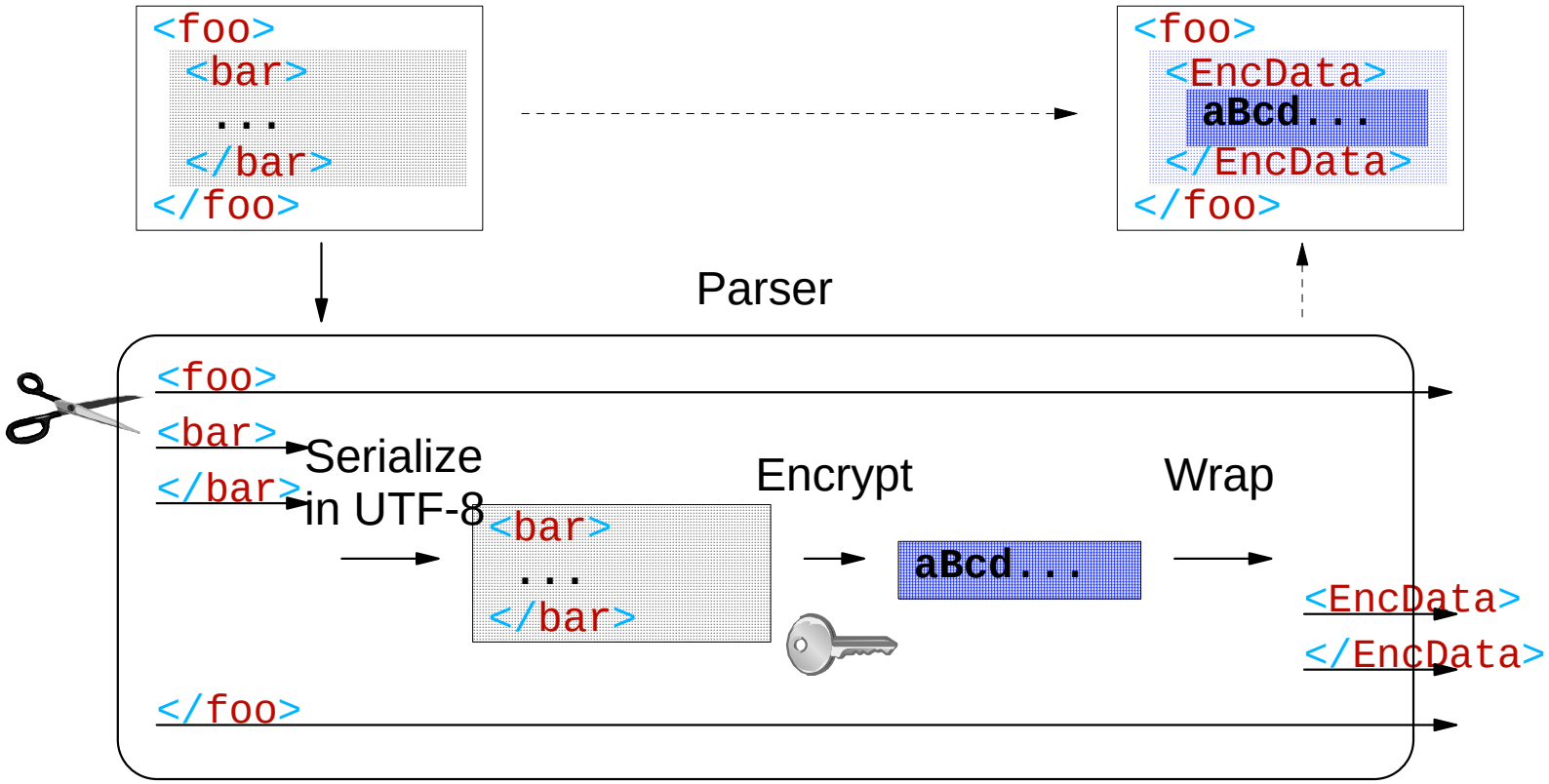
Design (3/4)

- Steps by application-level implementation



Design (4/4)

- Steps by parser-level implementation



Status

- DOM-based implementation
- Environment
  - Java 2 SDK 1.3
  - Java Cryptography Extension (JCE) 1.2
  - Xerces Java Parser 1.2
- Supported algorithms
  - 3DES, RSA-v1.5, base64
- Supported data to be encrypted
  - XML element, XML element content, arbitrary binary data
- Reference
  - IBM alphaWorks
    - http://www.alphaworks.ibm.com/tech/xmlsecuritysuite

Code Fragment for Encryption

```
// 0. Already given
Element elem = ...;  // Element to be encrypted
Key key = ...;  // Key named "key"
AlgorithmFactory algFac = ...;  // Factory for algorithm implementations

// 1. Create <EncryptedData> as template
EncryptionMethod em = new EncryptionMethod();
em.setAlgorithm(EncryptionMethod.TRIPLE_DES_CBC);
KeyName kn = new KeyName();
kn.setValue("key");
KeyInfo ki = new KeyInfo();
ki.addKeyId(kn);
EncryptedData ed = new EncryptedData();
ed.setType(EncryptedData.ELEMENT);
ed.setEncryptionMethod(em);
ed.setKeyInfo(ki);
Element encData = ed.createElement(elem.getOwnerDocument());

// 2. Create and set up encryption context
EncryptionContext encCont = new EncryptionContext();
encCont.addData(elem, false, encData);
encCont.setKey(key);
encCont.setAlgorithmFactory(algFac);

// 3. Perform encryption
encCont.encrypt();
```

Created <EncryptedData>

```
<EncryptedData
  xmlns="http://www.w3.org/2001/04/xmlenc#"
  Type="http://www.w3.org/2001/04/xmlenc#Element">
  <EncryptionMethod
    Algorithm="http://www.w3.org/2001/04/xmlenc#3des-cbc" />
  <KeyInfo
    xmlns="http://www.w3.org/2000/09/xmldsig#">
    <KeyName>key</KeyName>
  </KeyInfo>
</EncryptedData>
```

Code Fragment for Decryption

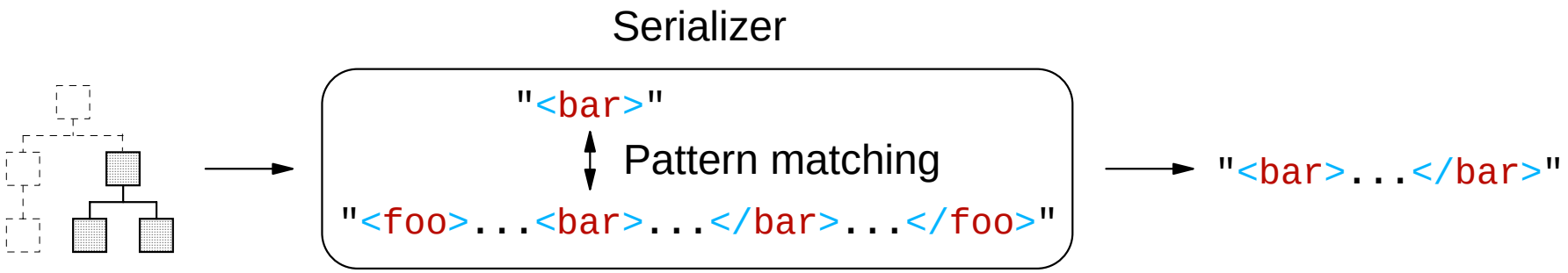
```
// 0. Already given
Element encData = ...;  // <EncryptedData> to be decrypted
KeyInfoResolver kiRes = ...;
                        // Resolver from <KeyInfo> to key
AlgorithmFactory algFac = ...;
                        // Factory for algorithm implementations

// 1. Create and set up decryption context
DecryptionContext decCont = new DecryptionContext();
decCont.addEncryptedData(encData);
decCont.setKeyInfoResolver(kiRes);
decCont.setAlgorithmFactory(algFac);

// 2. Perform decryption
decCont.decrypt();
```

Sample Program - XMLCipher

- XML element is being encrypted with key stored in keystore
- All parameters are provided using configuration file, which consists of:
  - `<data>` for input file, type of element to be encrypted, and output file
  - `<template>` for template specifying encryption algorithm and key (used only for encryption)
  - `<keyinfo>` for keystore name, keystore password, key alias, and key password

Challenges (1/2)

1. How to obtain octet sequence corresponding to DOM tree in encryption
   - Serialize DOM tree >> representation not preserved, e.g.,
     - Attributes' order
     - Whitespaces in attribute value
     - Quotation marks
     - ...
   - Extract octet sequence from XML document



Challenges (2/2)

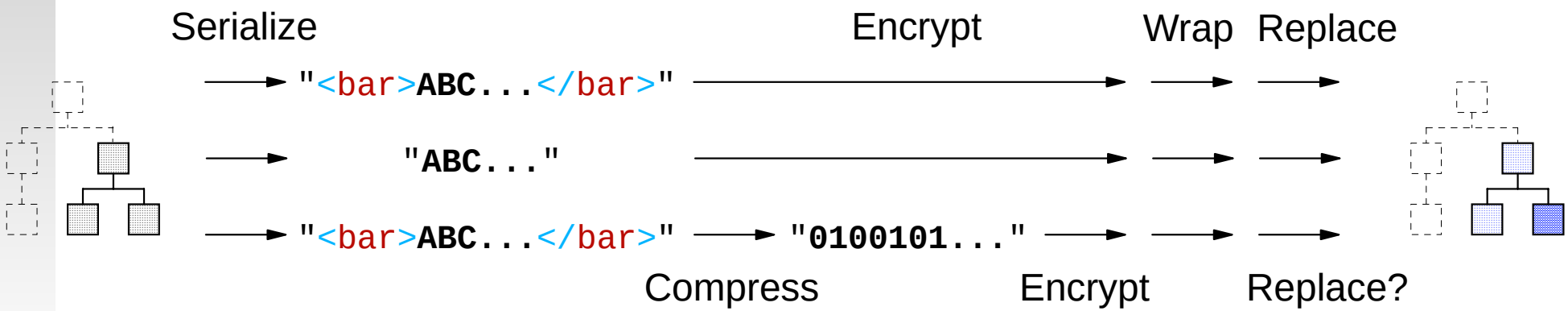
2. How to obtain DOM tree corresponding to octet sequence in decryption*
   - Place octet sequence in place of <EncryptedData> and then re-parse the whole XML document >> high cost
   - Parse octet sequence in context of <EncryptedData>

```
<?xml version="1.0"?>
<!DOCTYPE p:root (reference to external subset) [
  <!ELEMENT p:root ANY>
  (internal subset)
]>
<p:root xmlns:p="(some namespace)"
        (namespace declarations)
        (xml:lang declaration)
        (xml:space declaration)
        (xml:base declaration)>
  (octet sequence)
</p:root>
```

* Do not occur for parser-level implementation

Questions (1/2)

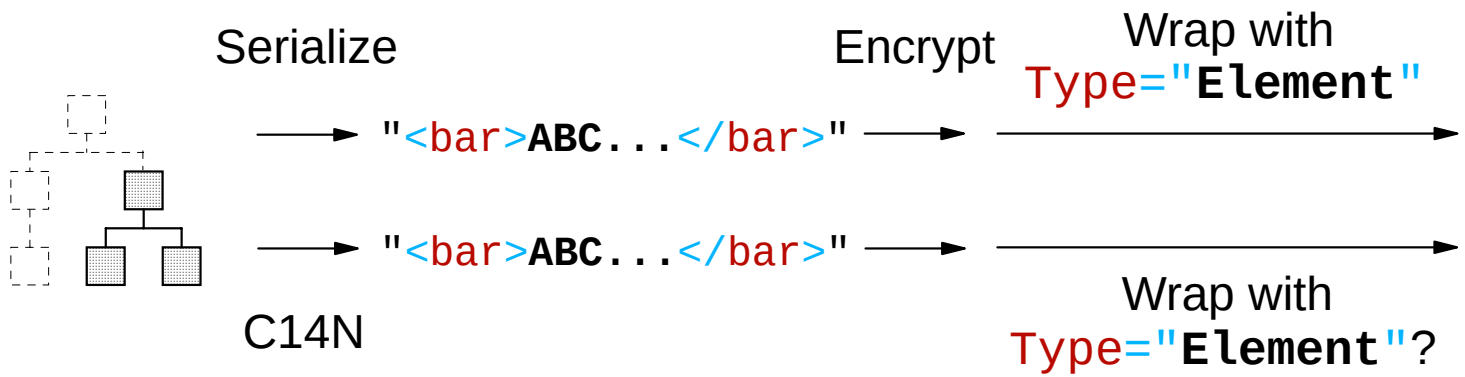
1. Spec says in Section 4.1 "Encryption"
   - If the data to be encrypted is an XML element or XML element content, the octet sequence is an UTF-8 encoded string representation of the element or its content ...
   - If the data being encrypted is an XML element or XML element content, the unencrypted data is removed and replaced with the new XML structure ...



Questions (2/2)

2. Spec says in Section 4.3 "XML Encryption"
   - If the application wishes to canonicalize or encode/compress the data in an XML packaging format, the application needs to marchal the XML accordingly and identify the resulting type with optional the `EncryptedData Type` attribute. ...
   - Element 'http://www.w3.org/2001/04/xmlenc#Element'

     `"[39] element ::= EmptyElemTag | STag content ETag"`



Backup

Processing Steps