Frequently Asked Questions

Questions

Answers

1. Mailinglist

1.1. Where's the archive for the list?

Currently, Gmane holds the messages of the last two weeks. This service also makes the mailinglist reachable with a news reader.

You can use the ezmlm mailing list controller to recieve previous messages by email. Send an empty email to security-dev-help@xml.apache.org for detailed information on how to use this service

2. Required background

2.1. Where can I learn about XML?

There are plenty of resources on the web, just use any search engine. You might start at XMLFAQ or ZVON.

2.2. Where can I learn about XML Digital Signatures?

The best place to start is W3C XML-Signature Syntax and Processing . Links on XML security in general can be found on The XML Security Page .

2.3. Where can I learn about XML Encryption?

The best place to start is W3C XML Encryption Syntax and Processing. Links on XML security in general can be found on The XML Security Page.

2.4. Where can I learn about Cryptography in general?

A lot of resources exist on the web, including the 'green bible' for cryptography: Handbook of Applied Cryptography. The Handbook of Applied Cryptography is completely online and it should satisfy most of your cryptographic hunger. Disadvantage of it is that it goes rather deep, so it isn't a executive overview or a "Learn XYZ in 21 days"-book

3. XMLDSig questions

3.1. What is the enveloped transform?

The enveloped transform is a special transform that enables the use of so-called enveloped signatures.

Enveloped signatures are signatures over an entire XML document, for which the `<Signature>` element is included in the document itself. An example could be:

```
<?xml version="1.0" encoding="UTF-8"?>
   <Root>
     <SomeContent>
       ...
     </SomeContent>
       <ds:Signature>
         <ds:SignedInfo>
           <ds:Reference URI="">
             <ds:Transforms>
               <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-sig
             </ds:Transforms>
           </ds:Reference>
         </ds:SignedInfo>
         ....
       </ds:Signature>
   </Root>
```

The `Reference` indicates that `Root` and it's descendants (except for comments) are signed, but the `Transform` element says to throw out the `Signature` element (that is the parent of this `Reference`) from the stream that is to be signed. Note that if there are other

`Signature` elements in `Root`, they will remain untouched.

3.2. What's the difference between C14N and ExclC14N?

C14N was introduced to solve some problems that arise when signing XML. Because XML allows to change the representation of an XML document without changing the actual content, signatures may break when different parsers are used to generate and verify the signature. A simple example of such an allowed change is changing the order of attributes within an element. (That is solved by C14N by sorting the attributes by alphabet)

Because a C14N'ed XML fragment inherits all the namespace declarations from it's ancestors, it is not possible to embed a signed XML fragment into a document that has other namespace declarations.

This is solved by ExclC14N. ExclC14N takes extra information as input in which you can specify which of the ancestor's namespaces should be included.

For more information on this topic, have a look at the C14N and ExclC14N sections of the W3C XMLDSig WG.