Release Information

1. Changes

This section describes the changes that have occurred between the various releases of the library

1.1. Version 1.1.0

Version 1.1 provides bug fixes to signature functionality + beta support for XML Encryption.

Changes from version 1.0.0 include :

- Beta implementation of XML Encryption, using an interface similar to that used in the Apache Java xml-security library
- Fix for bug where large text elements would be truncated during canonicalisation
- Provision of a *cipher* tool that can be used to encrypt and decrypt XML documents
- Updated and improved the Windows Crypto API interface
- Other bug fixes to signature functionality
- Updated to support Xerces 2.4/2.5 and Xalan 1.7

1.2. Changes from 0.10 - 1.00

Version 1.00 is the first release of the library considered basically stable. The interface is fairly simple, but all the mandatory requirements of the XML Digital Signature standard, canonicalised XML, exclusive canonicalised XML and XPath-Filter2 are implemented.

Changes from version 0.20 include :

- Implementation of remaining KeyInfo elements (SPKIData, PGPData and MgmtData
- Re-implementation of XSECXPathNodeList using a binary search to speed up list searches
- Support for Intel Compiler 6.0 on Linux and Forte CC (CC 5.4) on Solaris
- Limited support (i.e. without Xalan integration for NetBSD, FreeBSD and Cygwin
- A number of minor bug fixes

1.3. Changes from 0.10 - 0.20

Includes a number of bug-fixes and a first cut at a provider for the Windows Crypto API.

1.4. Beta release 0.10

The current release of the xml-security-c library is a beta of the XML Digital Signature code, and is the first version of the library produced within Apache's XML project.

Previous versions were created within Sourceforge as the xml-security-c project at that site.

1.5. Changes from 0.03-0.10

The following changes occurred between versions 0.03 and 0.10 :

- A *threadTest* tool, which is primarily used to show how multiple threads can access the library under Windows
- Windows and UNIX URI Resolvers, based on the Xerces resolver, but which can handle HTTP re-directs
- Completed a basic level of API documentation
- Reviewed library to ensure usage of UTF-16 internally
- Closed a number of memory-leaks caused by exceptions being thrown

1.6. Changes from 0.02-0.03

The following changes occurred between 0.02->0.03

- A *txfmout* tool to output the transformed references
- Updates to *templatesign* to support KeyInfo elements and RSA signatures
- Library support for RSA signatures
- API support form manipulating KeyInfo elements
- Pluggable KeyInfo Resolvers (to allow an application to supply an object that will resolve a given KeyInfo to a key)
- Pluggable URI resolvers
- API support for creating references and transformations
- Re-written Makefiles for *NIX
- Improved API docs
- Envelope Transform that no longer uses XPath transforms (much quicker)
- Uses new Xerces DOMNode objects - has made for significant speed improvements

1.7. Changes from 0.01-0.02

- Signing functionality
- Update C14n canonicalisation
- Exclusive Canonicalisation
- Basic ability to create a signature via the API (rather than just from an XML template file
- SHA-1 HMAC support

- Basic Documentation of core API
- A *templatesign* tool

2. Future Release Plans

A number of items are planned for after the 1.0 major release.

- XML Encryption Support
- decoupled, pluggable interface for transforms and signature types. Currently these are hard coded into the library, so it is not possible for calling applications to "plug-in" their own signature types or transforms
- Implement a SAX based canonicaliser for situations where an XML document needs to be read in and directly canonicalised (i.e. where the signature is not part of the document being processed
- Improved *KeyInfo* resolver interface leading to...
- XKMS client support
- Implementation of libgcrypt as a crypto provider
- Support for PGP/GPG key based signatures (using libgcrypt and gpgme)
- Bring the API in-line with JCP 105 (if appropriate)
- Eventually (the GRAND PLAN) an XKMS server implementation
- Implementation of a core set of encryption primitives to allow decoupling from OpenSSL for "light" applications