Experience XML Security

The XML-Security Plug-In for Eclipse

Dominik Schadow

Pasingerstrasse 28, 82152 Planegg, Germany
info@xml-sicherheit.de,
http://www.xml-sicherheit.de

Abstract. The XML-Security Plug-In is a freely available plug-in for the Eclipse platform and provides versatile XML security features in an educational software like environment. Users can experiment with XML security and inform themselves about the corresponding W3C recommendations. The aim of the plug-in is to raise interest in XML security functions and especially encourage users to sign and encrypt their personal data and messages. This is a common aim with CrypTool, an educational software for modern cryptography. This article will explain the basic features of the XML-Security Plug-In and give a perspective on the cooperation with the more general CrypTool.

1 Introduction

Security features for the Extensible Markup Language (XML) are basically defined by two recommendations of the World Wide Web Consortium (W3C): XML-Signature Syntax and Processing[1] and XML Encryption Syntax and Processing[2]. These two recommendations are available for almost three years now and used in lots of commercial and non-commerical products and services: Apache XML Security[1] and IBM XML Security Suite[2] are surely the most well-known. But as standard cryptography, XML security lacks of end user acceptance.

One reason for that is, besides the relatively short availability of only three years, certainly the absence of easy to use educational software in the XML (security) area with the main focus on practical execution. Normal users are not interested in reading extensive W3C recommendations and gaining all their knowledge from theoretical studies. Furthermore a deep and broad knowledge of XML security can only be achieved by a combination of practical experience and theory, not only by studying the corresponding recommendations.

The *XML-Security Plug-In* tries to bridge this lack of user acceptance. Therefore the main focus lies clearly on easy practical execution of the W3C recommendations on digital signatures and encryption for end users. The plug-in is the result of the authors diploma thesis 2004 at the University of Applied Sciences

[1] See http://xml.apache.org/security
[2] See http://www.alphaworks.ibm.com/tech/xmlsecuritysuite

J. Dittmann, S. Katzenbeisser, and A. Uhl (Eds.): CMS 2005, LNCS 3677, pp. 321–329, 2005.

Furtwangen. Since then the plug-in has been continuously enhanced with new features and possibilities.[3]

The plug-in requires an Eclipse[3] platform and extends it with versatile XML security features like digital signatures and encryption. As mentioned before, the main objective is to provide users an easy-to-use tool with rich functionality to experiment with XML security. Users can sign and encrypt their own existing XML documents; there is no limitation on sample files. The educational part - the theory behind XML security with its various recommendations and extensive information on how to use the plug-in - is located in the included online help.

In order to provide an easy and useful access to XML security, the XML-Security Plug-In consists of these five components

1. Canonicalization (with or without comments)
2. Digital Signatures
3. Verification
4. Encryption
5. Decryption

The XML-Security Plug-In addresses users with knowledge in XML and basic knowledge in modern cryptography, especially in the different signature and encryption algorithms (like AES or RSA). Experience in XML security is not necessary.

Users with no or little knowledge in modern cryptography can use the excellent educational software *CrypTool*[4] as a starter or reference tool. CrypTool brings out a good introduction for cryptographic knowledge as well as an extensive reference tool for cryptography and supports beginners in learning cryptographic basics needed for the XML-Security Plug-In.

2 Standard Cryptography

2.1 CrypTool

As in the XML-Security Plug-In, practical experience plays the important role in the software CrypTool. The application lets the user apply and analyse complex cryptographic mechanisms and reveals the inner working of digital signatures and hybrid encryption with interactive data flow diagrams. An extensive online help with information about digital signatures, hash functions and much more is available.[4]

CrypTool comes as a stand-alone application for Windows and addresses both cryptography beginners and advanced users. It is completely available in English and German. The used programming language is C/C++, therefore a Win32 environment is required.

[3] See http://www.eclipse.org
[4] See http://www.cryptool.com

The aim of CrypTool is to teach cryptographic methods and standards with a modern and interesting tool and to sensitize employees and end users in IT-security as well as to enable a deeper knowledge and awareness for cryptography. The methods available include both classical methods like Caesar encryption algorithm and modern cryptosystems like AES and DES algorithms. Asymmetric algorithms like RSA and DSA, based on both the factorization problem and on elliptic curves, are also available.

The development of CrypTool startet in 1996 and is basically done by the Deutsche Bank with the support of the universities of Darmstadt, Siegen and Karlsruhe. In this nine years history the software gained several awards like the European Information Security Award 2004[5] and some other German awards. The current version is 1.3.05, the next release is 1.3.10 and will be available soon.

2.2 Cooperation of CrypTool and XML-Security Plug-In

As described before, CrypTool and XML-Security Plug-In have an identical focus on end users in different areas of cryptography. CrypTool addresses beginners as well as advanced users, whereas the XML-Security Plug-In focuses on advanced users.

Cooperation between the CrypTool and the XML-Security Plug-In is useful (and indeed considered in practice) because they share the same aim. Plus the XML-Security Plug-In provides a focus on XML which is missing in the more general CrypTool to date. Moreover plans for a redesign and redevelopment of CrypTool have been in mind of the CrypTool authors for a while now. The opinion poll[6] on the CrypTool home page came to a clear decision for the programming language Java: 213 votes totally, 126 people voted for Java.[7] 56 users voted for the combination Java and Standard Widget Toolkit (SWT) of Eclipse, 70 for the combination of Java and Swing. One reason for this decision may be the much longer availability of Java Swing, whereas SWT is relatively new and not so much known outside the Eclipse community yet.

Based on this voting, the new *JCrypTool* will, as the *J* indicates, be completely developed in Java and will be available as an Eclipse Rich Client Platform (RCP) for multiple platforms. Due to the advantages of SWT and Eclipse, better performance and integration, SWT is the prefered technology.

As CrypTool itself, JCrypTool will be available in German and English. For the first time extensive XML security features - signatures, verification, encryption and decryption as well as the theoretical information - will be available in this educational software.

More information or a release plan are not yet available, the development process is still in an early stage. All information will be published on the CrypTool and on the XML-Security Plug-in home page.

[5] See http://2004.rsaconference.com/europe/awards.aspx
[6] Poll available on CrypTool home page, results made available by CrypTool officers.
[7] As at June 2005.

3 XML-Security Plug-In

3.1 Plug-In Structure

The XML-Security Plug-In is a free of charge available plug-in for Eclipse version 3.0 and above. Besides Eclipse only an up to date Java Runtime Environment (JRE) or J2SE Development Kit (JDK) version 5.0 or newer is required. The XML-Security Plug-In can be used at no costs (freeware) in private, education and teachings as well as non-commercial and commercial environments. Redistribution is admitted and welcome, however commercial redistribution requires the explicit allowance of the developer.

To support the users in learning XML security, the design of the plug-in is as open as possible. This means that all the necessary data for a digital signature or encryption has to be entered or selected by the user. Depending on the wizard, different amounts of information are required before the XML document can be secured. By intention, the plug-in wizards provide many different functional choices. Therefore the user must perform quite a lot of explicit selection steps before a wizard can finish successfully.

In order to provide an easy access to its features, the XML-Security Plug-In extends the Eclipse workbench in different views. The most important extension point is available in standard Eclipse editors, as shown in the following figure. Moreover functions of the plug-in can be called via navigator or package explorer view.

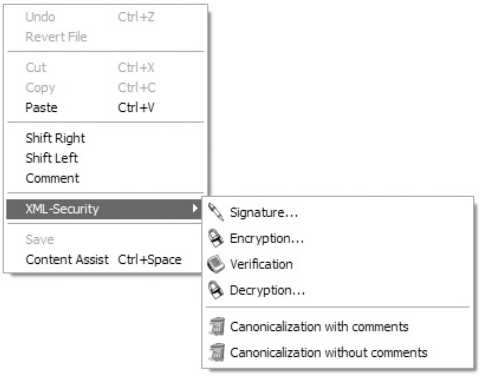

Fig. 1. Context menu in an Eclipse editor

Every extension point consists of the same menu and contains the same structure. Nevertheless there are some differences between these extension points. The most important one is that a text selection can only be signed with a call from the context menu of an editor. The possibility of signing or encrypting a text selection shows a special feature of XML security: to secure only one or more elements, element content or parts of a document (document fragment).

It is possible to combine the wizards of the plug-in. A signed document can be encrypted subsequently and vice versa. Both ways are possible deliberately, even though it is more common and suggestive to sign first and encrypt afterwards. By comparing both possibilities, first sign - then encrypt and first encrypt - then sign, the user can easily compare these two sequences and discover the pros and cons for each one by own experience.

The signed and/ or encrypted XML document can be used outside of the plug-in and shared with other users. It is also possible to verify foreign XML signatures which were not created with the XML-Security Plug-In. Each signature which contains a valid KeyInfo element can be verified. The decryption of XML documents encrypted with other tools should be possible too, but depends on the used algorithms.

3.2 Apache XML Security API

The XML security application programming interface (API) shipped with the plug-in is the already mentioned Apache XML Security API version 1.2.1. This API is available as open source and comprises an extensive implementation of the W3C recommendations. Digital signatures are completely and stably implemented, every part of the W3C recommendation can be used. Encryption functionality grows with every release and catches up to digital signatures. Not every part of the encryption recommendation is completely and stably implemented yet. The range and the quality of the whole API are surely a result of the continuous enhancement by the large Apache community.

One point of criticism is the incomplete documentation and the basic code samples. Corresponding to the todo list this will be changed in future versions.[8]

There are special terms of license[9] for the Apache XML-Security API contained in the plug-in. These terms of license are only valid for the Apache XML-Security API (*xmlsec.jar*).

3.3 Basic Security Profile

Both the Digital Signature Wizard and the Encryption Wizard support the Basic Security Profile version 1.0 working group draft (BSP)[5] by the Web Services-Interoperability Organization (WS-I)[10]. The WS-I is an open industry organization, the BSP therefore consists of a set of non-proprietary web services specifications. The main objective of both the profile and the organization is higher service interoperability across platforms, operating systems and programming languages.

Using this profile in the plug-in activates different restrictions in the respective wizard. Due to the fact that the BSP does not allow enveloping signatures and discourages enveloped signatures, the Digital Signature Wizard preselects

[8] See Todo List on the Apache XML Security home page.
[9] See http://www.apache.org/licenses/LICENSE-2.0
[10] See http://www.ws-i.org

detached signatures for example. Other restraints affect the available algorithms; here the profile encourages the usage of RSA algorithms.[11]

3.4 Canonicalization

Canonicalization provides a comprehensible way to XML security. The user can select between the two types of exclusive XML canonicalization; one removes comments, the other one maintains them.[6] The intention of these commands is to point out to the user what happens to an XML document during the canonicalization process and to make the differences between a canonicalized and a non-canonicalized XML document visible.

A canonicalized (normalized) XML document has some significant differences to a standard XML document: among other things the canonicalized XML document is encoded in UTF-8, empty elements are converted to start-end tag pairs and whitespace outside of the document element and within start and end tags is normalized.[7]

3.5 Digital Signatures

Digital signatures are the most complex part of the plug-in. A wizard supports the user in the creation process and lets him enter step by step the necessary information to create a digital signature. With its versatile possibilities the wizard encourages users to combine different settings of digital signatures. Short annotations in this and every other wizard inform the user about the current wizard page and the required settings.

The first choice on the first wizard page refers to the data to be signed. Possible selections are *document*, *selection* and *XPath*. *Document* signs the complete XML document, *selection* an existing and well-formed text selection in the openend editor and *XPath* a document fragment specified by the entered or selected XPath expression.

The XML digital signature recommendation offers different kinds of signatures. All in common is that the signature object itself appears in XML syntax, but the data to secure can consist of XML or arbitrary data. Everything that can be referenced through a Uniform Resource Identifier (URI) can be signed. Because of that the user can choose between the standard W3C signature types *enveloping*, *enveloped* and *detached* in the wizard.

In an enveloping signature, the signed data appears inside the signature element. Signed data surrounding the XML signature is called enveloped signature. A separated signature (in the same document as the data or outside) is called a detached signature.[8]

The complete first page of the digital signature wizard is shown in the next figure. Every wizard has the same standard layout, with short help information at the top, content in the middle and navigation buttons at the bottom.

[11] See section 8 of the Basic Security Profile working draft.

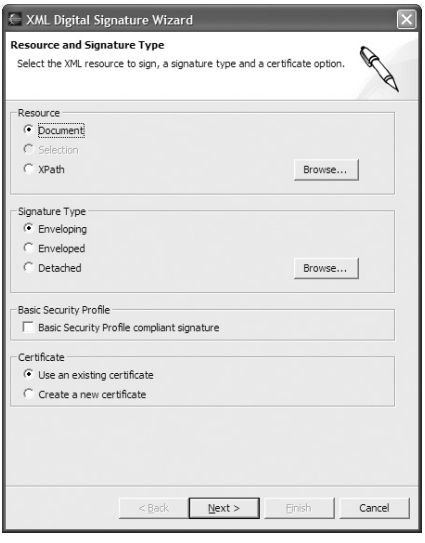

Fig. 2. First page of the digital signature wizard

On the following wizard page the user has either to enter the necessary information for an existing Java KeyStore or to create a new one with the help of the wizard. A newly created certificate is stored in the current project directory and can be reused in future signing processes inside and outside of the plug-in.

The third and last page finally allows the user to select the algorithms to use for the signature. Necessary information are the algorithms for canonicalization and transformation as well as a message digest and a signature algorithm.

After all the information is given, the wizard signs the selected document (-fragment) and shows the secured XML document to the user.

3.6 Verification

Verification lets the user quickly verify any signed XML document that contains a valid KeyInfo element. Corresponding to the XML digital signature recommendation, the KeyInfo element is an optional element that enables the recipient to obtain the key needed to validate the signature.[12] In the plug-in the KeyInfo element is required for verification and contains the users' certificate.

The result of the verification, a valid or invalid signature, is directly shown to the user. The fast verification result makes it easy to manipulate parts of an XML document and to see, which manipulation breaks a signature and which does not. This way the user can experience the special feature of XML signatures where only document fragments are signed and the rest of the XML document remains in an insecure state.

[12] See section 4.4 of the XML-Signature Syntax and Processing recommendation

3.7 Encryption

The encryption of an XML document is supported by another wizard. As in the digital signature wizard, the user must first select which data should be encrypted. Possible selections are again *document*, *selection* and *XPath*.

Besides some optional declarations the user must select an encryption algorithm and a file to store the encryption key in. Available algorithms are AES or Triple-DES for example. This key file is very important and required to decrypt the XML document later.

After all information is given and the key file is stored, the wizard takes care of the encryption process and shows the secured XML document to the user.

3.8 Decryption

An encrypted XML document or a fragment is decrypted with the help of a small wizard. The user only has to enter the path to the correct key file and to select the transport algorithm used for encryption.

The wizard then decrypts the XML document and shows the result to the user. Decryption will fail if a wrong key file is used.

3.9 Online Help

The online help contains the educational and theoretical parts of the XML-Security Plug-In and consists of two main sections. First it includes extensive information about the two W3C recommendations on digital signatures and encryption as well as related specifications or recommendations like canonicalization and the Basic Security Profile. The second section of the online help contains extensive support and hints for the first steps with the plug-in, its usage and possibilities.

The online help is easily accessible in the Eclipse workbench all the time and is completely in German. In the new Eclipse platform version 3.1 the online help will be available in a special view as *Dynamic Help* too. This way help information can always be visible in the same perspective and not only in a separate window.

4 Conclusions

The XML-Security Plug-In provides an easy and practical access to XML security and the W3C recommendations for end users without much previous knowledge or experience. Users can get in touch with cryptography for XML in an interesting and practical way and learn all aspects of XML security by experimentation. In addition to the practical experience the theoretical background is available in the online help.

CrypTool is a recommendable starting and lookup tool for classic and modern cryptography and contains all the basics needed to understand XML security and to work reasonable with the XML-Security Plug-In. Both tools can easily be used together and extend each other in various ways.

The intention of the plug-in is to interest and sensitize more users for the need of cryptography and to show them, how easily XML documents can be secured with the W3C recommendations. To support this aim, the usage of the XML-Security Plug-In is totally free of charge and continuously enhanced with new possibilities and features.

The latest version of the XML-Security Plug-In, tutorials, documentation, user discussion group and much more are available on the plug-in home page www.xml-sicherheit.de. Critical user feedback and suggestions are always welcome and appreciated.

References

1. Eastlake III, D., Reagle, J., Solo, D.: XML-Signature Syntax and Processing. RFC 3275, March 2002. http://www.ietf.org/rfc/rfc3275, and W3C Recommendation, 12 February 2002. http://www.w3.org/TR/xmldsig-core/
2. Eastlake III, D., Reagle, J.: XML Encryption Syntax and Processing. W3C Recommendation, 10 December 2002. http://www.w3.org/TR/xmlenc-core/
3. Schadow, D.: Ein praktisches XML Signatur- und Verschluesselungstool. Datenschutz und Datensicherheit, **4** (2005) 193–196
4. Deutsche Bank: CrypTool. eLearning Program for Cryptology, 29 March 2005. http://www.cryptool.com
5. Barbir, A., Gudgin, M., McIntosh, M., Morrison, K.: Basic Security Profile - Version 1.0 (WGD), Work-ing Group Draft, 12 May 2004, http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html
6. Boyer, J.: Canonical XML. W3C Recommendation, 15 March 2001. http://www.w3.org/TR/xml-c14n
7. Boyer, J., Eastlake III, D., Reagle, J.: Exclusive XML Canonicalization. W3C Recommendation, 18 July 2002. http://www.w3.org/TR/xml-exc-c14n
8. Eastlake III, D., Niles, K.: Secure XML - The New Syntax for Signatures and Encryption. Boston: Addison-Wesley (2003)