Enhancing the Security of Applications using XML-based Technologies

Syed I Ahmed
Signal Stream Inc.

Signal Stream Inc.
580 Wilkie Drive Orleans,
Ontario. K4A 1N3
CANADA

Project Manager: S. Dahel 613-993-9949

Contract Number: W7714-3-08506

Contract Scientific Authority: S. Dahel 613-993-9949

DEFENCE R&D CANADA - OTTAWA

Contractor Report

DRDC Ottawa CR 2003-201

December 2003

Abstract

XML and associated core family of XML based W3C Recommendations have no built-in information security features but have rich characteristics that can be exploited to devise numerous security schemes. The evolving XML Security Standards and COTS tools that can enhance the security of applications are identified.

This page intentionally left blank.

TABLE OF CONTENTS

FIGURES AND TABLES

APPENDICES

1. EXECUTIVE SUMMARY

Information security concerns Authentication, Authorization, Confidentiality, Privacy, Digital Signature/Non-Repudiation, Information Integrity, Digital Rights Management, Integrity and Availability of information within critical time. However, vulnerabilities could remain in the information system even after security mechanisms are implemented.

To appraise how eXtensible Markup Language (XML) based technologies can be used to enhance the security of applications, at first it is helpful to form a general idea of XML as defined by the World Wide Web Consortium (W3C). XML is not a computer programming language but a language to encapsulate data with markups, or tags that provide hints about the meaning in the document. XML is both, *human and machine-readable*. It is unlike HTML, that is widely used in the current Internet where a finite set of HTML markups or tags are used essentially to indicate the *format* (not meaning) of the document to be displayed to humans. XML does not have a finite set of vocabulary. It is a nascent language from which unlimited numbers of user defined XML based languages can be created for different domains, sciences, communities or business groups. XML can potentially interconnect heterogeneous computer systems and their data structures to facilitate interoperability. XML is expected to become the foundation language for the next generation Internet that is being envisioned to evolve into the *Semantic Web* which will understand meaning to offer AI (artificial intelligence) based services to both, humans and computing machines.

XML is text based and without security mechanisms information written in XML could be easily read and the meaning conveniently derived from the hints provided by the XML markups. Therefore, security of information written in XML can be of major concern, especially if XML is considered for deployment in the defense sector. However, due to the increasing importance of XML in modern IT, the Network Information Operations section of DRDC Ottawa initiated this project to explore XML from security perspective. First, in Phase I, the features of XML and XML family of Recommendation/Standards was examined from information security viewpoint. Secondly, in Phase-II, it was required to demonstrate with a proof of concept (POC) how the security of an application could be enhanced with XML technologies. The findings may be summarized as follows:

- *Security features in W3C XML Recommendations:* No XML vocabulary, language element or document structure has been reserved for security in the root XML specification or in the core family of XML related W3C Recommendations. However, XML family of recommendations is rich in features that can be exploited to devise numerous innovative information security schemes. XML documents are *tree structured*, which starts with one root and where branches (or information elements or segments) have parent child relationships. This facilitates implantation of security mechanisms in whole or selected parts of the document that may reside locally or as XML linked sections in several remote sites.

- *XML based security standards:* Specifications for XML based information security standards, protocols and services are being developed by W3C, Organization for Advancement of Structured Information Standards (OASIS) and by the Apache Group. Currently, XML Security architecture includes 5 specifications: (i). W3C XML Encryption, (ii). W3C XML Digital Signature, (iii). W3C XML Key Management Specification (XKMS), (iv). OASIS Security Assertion Markup Language (SAML), (v).

OASIS XML Access Control Markup Language (XACML). W3C *Encryption* and *Digital Signatures* has features that are uncommon with other security standards, e.g. XML security allows signatures and encryption over selected parts of the document at any granularity level, while allowing the other parts to remain as is. Both XML and non-XML content can be signed, encrypted and encapsulated as an XML element. *XKMS* is a service that provides simplified interface to the complex Public Key Infrastructure (PKI). It verifies digital certificates and provides electronic encryption key registration and information service. *SAML* provides a means to express information about authentication, authorization and other attributes about the end user or machine to facilitate "Single Sign On" (SSO) to multiple web services. *XACML* defines vocabulary for making configurable rules for access control.

- *XML COTS Security tools & maturity level:* Most XML based Security COTS products are a set of software with optional developmental tool kits, such as SDK (Software Development Tools), JAVA Script and Application Programming Interfaces (APIs). Major vendors of XML Security products include IBM, Microsoft, Entrust, Baltimore, Netegrity, Phaos, Waveset, PureEdge, Conclusive and a few others. Product deployment requires vendor training and therefore the maturity level these products could not be experimentally determined due to limited time. As XML Security standards are still developing, some vendors include their proprietary solutions to implement secure transactions that are only partially compliant to the emerging W3C and OASIS XML Security Standards. These COTS are not necessarily interoperable with products of other vendors and many require that software from the same vendor be installed at both the Client and the Server side of the data network.

- *XML Application in the PKI environment:* An XML Applications can be treated just like any other information object and protected using Public Key Infrastructure (PKI) using XKMS which is a Web Service that hides many complexities of PKI.

In Phase–II, of this research, for the POC, *eform* (electronic form) was selected as an application. Products from several *eform* vendors was briefly studied from XML security perspective but it was unclear from the limited vendor's on-line documentation as to what extent these COTS were compliant to the W3C and OASIS XML Security standards. Adobe Acrobat™ Professional version 6 that was readily available at DRDC was selected as a candidate application as it has *eforms* capability. Most *eform* vendors expect the users to utilize their product in conjunction with their optional Software Development Kit (SDK) for configuration, customization and extension. An experimental POC with the Acrobat would require the use of Acrobat SDK and ASN (Adobe Solution Network). Using the Acrobat manual and with some experimentation it was recognized that that Acrobat *eform* besides use of passwords, has the capability for multiple Digital Signatures, Encryption, Digital Certificates and utilization of Default, Microsoft Windows or third party based Certificate Authority (CA) and to create a list of Trusted Entities. From XML perspective, Acrobat architecture uses XML based language called XML Data Package (XDP), Form Data Format (FDF) and Extensible Metadata Platform (XMP). The metadata can include security information. These provide a common XML framework to standardize the creation, processing and interchange of electronic document across workflow. In many respects, Acrobat security mechanisms are not compliant to XML Security Standards defined by W3C and OASIS.

Conclusion: More research and experimentation is required with XML & COTS products to determine how the security of applications can be enhanced with XML technologies.

2. REPORT ON PHASE- I WORK

2.1 Research Approach

At first, the concept of *information security* was established. Then the features of W3C XML and core XML related Recommendations were surveyed from information security viewpoint. It was followed by examination of XML Security Standards being developed by W3C and OASIS and its relationship to PKI. Attempt was made to assess the maturity levels of some the XML Security COTS (Commercial Off the Shelf) products. Finally, a POC (Proof of Concept) was attempted to show how XML technologies could be used to ensure security of a COTS Application like eforms.

2.2 The concept of information security

Information security deals primarily with the following issues:

1. Authentication (*e.g. who, or which machine or organization is it?)*
2. Authorization *(e.g. what resources and action are permitted ?)*
3. Integrity *(e.g. is the information intact or was inappropriately changed?)*
4. Digital Signature /non-repudiation (*e.g. has the user agreed and signed digitally?)*
5. Confidentiality (*e.g. is information unreadable to unauthorized reader?*)
6. Privacy (*e.g. is information know only within a few allowed?*)
7. Digital Rights Management (*e.g. is use of digital content as per license agreement*?)
8. Availability (*e.g. is the information being deliberately delayed or blocked?)*

The above are mechanisms to protect from malicious or inadvertent attacks on application programs or on digital information that may be resident in an information systems or in transit through a communication network. If only a few of the above are implemented then greater security loopholes can exist. Even with the implementation of all the above listed security mechanisms, 100% information security cannot be guaranteed. It was observed that XML based security technologies can support 1 to 7 of the above list. Item 8. The Availability problem can be a normal telecommunication network delay and it can become a security issue if information is deliberately delayed or blocked and such a problem is not a part of XML Security.

2.3 XML

XML [2] is a modified subset of Standard Generalized Markup Language (SGML) [5] which is ISO [3] standard 8879:1985 used by the publishing industry. XML documents are tree structured, where branches and sub-branches have parent child relations. In a XML document, markups or tag encapsulated data and provide hint about the meaning in the branch. Unlimited number of XML based language is can be created. An example is the DARPA Agent Markup Language [31]. XML is based on Unicode [6] and is not limited to English language only.

XML documents are tree structured that start with a single root and branches (or information elements) have parent child relationship.

2.4 XML W3C Recommendation

A large numbers of XML related Recommendations have been defined by W3C [1] and these may be classified into the following categories:

1. The 'core' or near family of W3C XML Recommendations defines rules for constructing XML based documents. It includes recommendations for naming, locating, linking and identifying segments of information and placing them on the Internet or Intranets.

2. Underlying specification on which XML is developed e.g. Unicode *[6*] Namespace *[7*].

3. Other XML related W3C Recommendation, e.g. XML Security that facilitate development of secure interactive secure Applications over the web.

2.4.1. Core and underlying XML W3C Recommendations

The tables below list all the core family members of W3C XML Recommendations.

	W3C CORE family of XML Recommendations	Version	Status	Date
1	**Extensible Markup Language (XML)** http://www.w3.org/TR/REC-xml	1.0	R	6 Oct 2000
2	**XML** and DTD (Document Type Definition) http://www.w3.org/TR/REC-xml	1.0	R	6 Oct 2000
3	**XSL** (Extensible Stylesheet Language) http://www.w3.org/TR/xsl/	1.0	R	16 Nov 1999
4	**XSL Transformation (XSLT)** http://www.w3.org/TR/xslt	1.0	R	16 Nov 1999
5	**XML Schema Primer, Part1, Part 2** http://www.w3.org/TR/xmlschema-0/, http://www.w3.org/TR/xmlschema-1/, http://www.w3.org/TR/xmlschema-2/		R R R	2 May 2001 2 May 2001 2 May 2001
6	**XPATH** http://www.w3.org/TR/xpath	1.0	R	16 Nov 1999
7	**XPOINTER Element, Schema, Framework** http://www.w3.org/TR/xptr-element/ http://www.w3.org/TR/xptr-framework/ http://www.w3.org/TR/xptr-xmlns/		R R R	25 Mar 2003 25 Mar 2003 25 Mar 2003
8	**XLINK** http://www.w3.org/TR/xlink/	1.0	R	27 June 01

Table 1: Core XML family of Recommendations

XML relies on a number of underlying W3C/IETF standards and recommendations. Some of these are listed below.

	Underlying W3C /IETF Standards for XML	Version	Status	Date
1	**Namespaces in XML** http://www.w3.org/TR/REC-xml		R	14 Jan 1999
2	**XML Information Set**		R	24 Oct 2001
3	**Unicode**	3		

Table 2: Underlying specification/recommendation for XML

2.5 Security features in W3C XML & family of XML Recommendations

Examination of the root XML [2] specification, and associated core family of XML recommendations show that no XML mark-up, vocabulary, XML element, or document structure is reserved for information security. However, it can be observed that XML and XML family of Recommendations are rich in features that can be exploited to devise numerous information security schemes, with or without the use of encryption. This is made based on the observations in the following sections:

2.5.1. DTD and XML Schema Recommendations

Document Type Definition (DTD) [8] and XML Schema Recommendation [9] describes the prearranged "valid" and "well-formed" document format, structure and data type contents of an XML document that is agreed upon between the author and the reader (human or machine). The validity and well-formed-ness checks are made by comparing the received document with the associated Schema or DTD that is normally made in all XML based transactions between nodes to verify that the received document has valid structure and data types. The XML Schema has more advanced features for data type and document structure checking than DTD. Although no security is defined in XML Schema and DTD, these specifications can support some form of security due to the fact that tampering of the XML document can result in failure in "validity" and "well-formed-ness". However, the weakness is that if only the data is tampered and not the document structure, then it will pass "validity" and "well-formed-ness" checks at the receiving end. A security mechanisms based on XML Schema or DTD without encryption is much weaker compared to the use of Digital Signature that creates a "digest" which gets affected if any kind of change is made to the document.

2.5.2. W3C XPATH and XPOINTER Recommendations

W3C XPATH [10] and W3C XPOINTER [11] Recommendations describes the rules for placing and locating information in the XML document that could be segmented and where segments may located locally or in different parts of the world in the Internet or Intranet. XPOINTER is used to locate information within a XML document to the granularity of a single character. From security perspective, XPATH and XPOINTER these can be used to strategically place and locate information, such as Encryption Keys, Digital Signatures, passwords, and other security mechanisms. XPOINTER can

also be utilized to detect document tampering that causes a shift in the position even by one character.

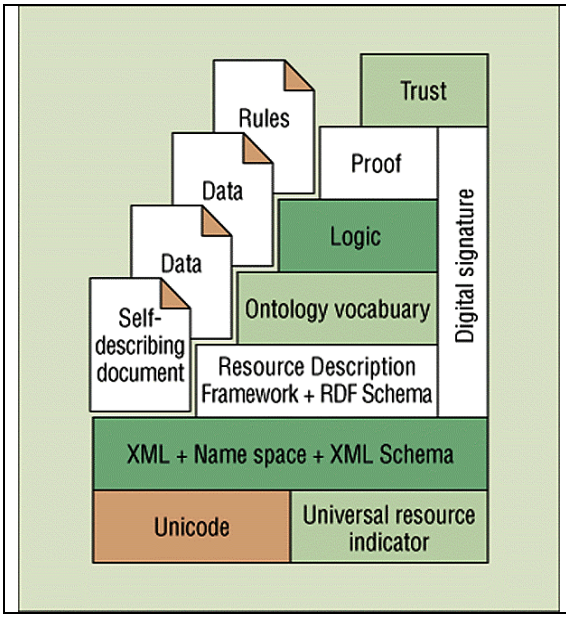
2.5.3. W3C XLINK Recommendation

Information in a XML document can reside as segments within the same network node or distributed among many different nodes in a network. W3C XLINK [12] describes how information different segments may be inter-linked. Unlike HTML, XML allows both unidirectional and multi-directional linking. This feature can be exploited to create variety of information security schemes. An instance could be that critical military information may not be stored in a single physical location, but split up, segmented and may be distributed to a different geographical location but inter-linked linked using the W3C XLINK standard.

2.5.4. W3C RDF Working Draft

W3C Resource Description Framework (RDF) [33] is for meta-data (data about data) that will facilitate interoperability between Applications running on the Internet and will become one of the foundations for the development of Semantic Web. Security information can be placed as a part of RDF.

2.6 XML Family of Recommendations & Security for Semantic Web perspective



A model proposed for the Semantic Web is the Layer Cake Model described in the article in the IEEE INTELLIGENT SYSTEMS March/April 2001 page 30 shown on the left.

The position of Unicode, URI (Universal resource indicator), XML, XML Schema and RDF is shown in the model.

Some of the layers in the model are still undefined

From the proposed model of the Semantic Web, it can be observed that there is ample emphasizes on *security* and the model suggests the use of Digital Signature to form trust.

Figure 1: W3C XML Recommendation from Semantic Web model

2.7 W3C Recommendations involved in XML Document & Application creation

The diagram below [ref: Figure 2 is an attempt to show the W3C Recommendations that can be used to create XML documents for building secure XML based Application.

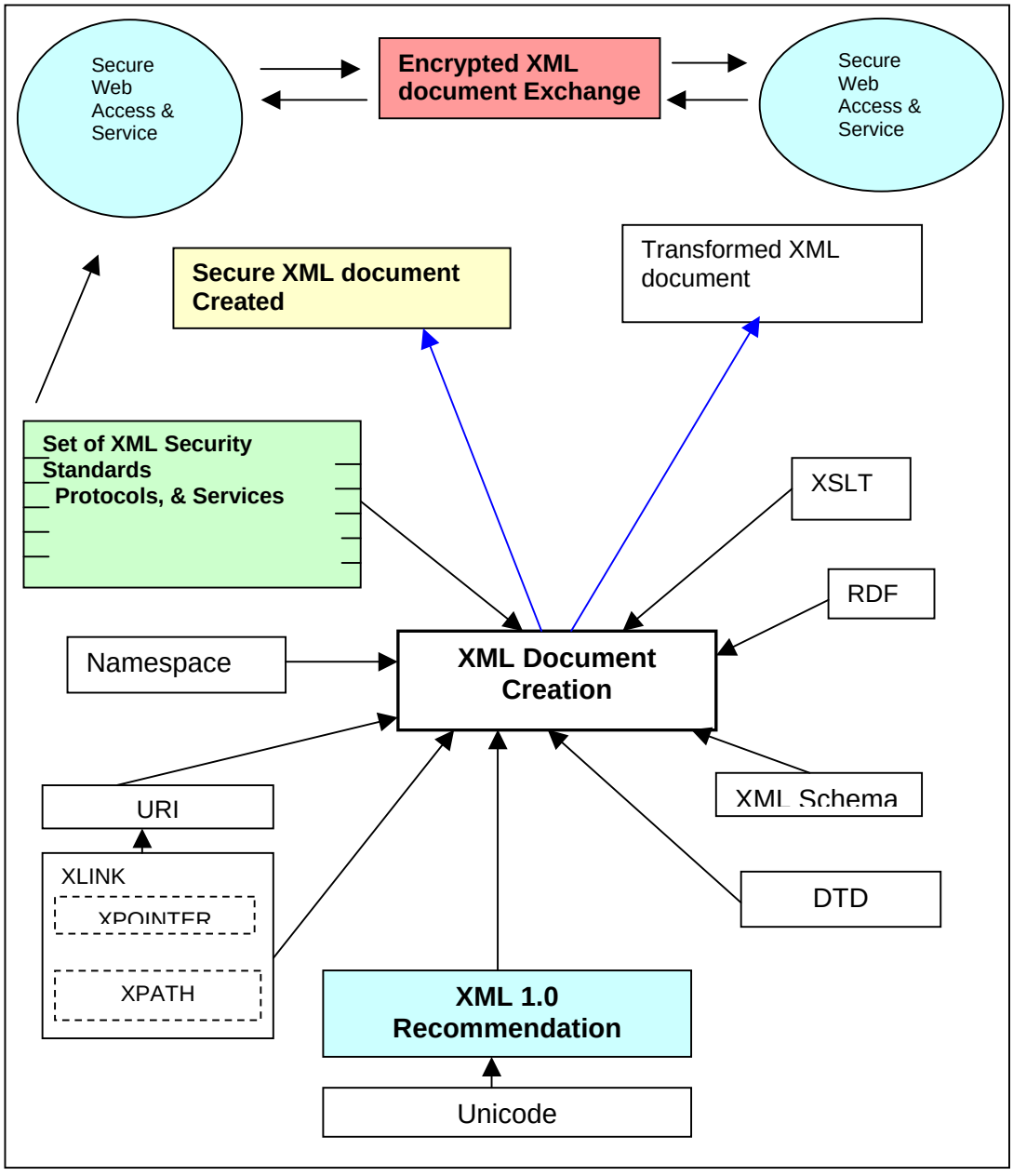


Figure 2: Influencing W3C Recommendation in XML document creation

2.8 XML Security Standards

The features of XML and in the XML based Recommendations are being used by the W3C, Organization for Advancement of Structured Information Standards (OASIS) [15] and the Liberty Alliance [[16] Project group to develop a set of standards for information security in XML. The present status of W3C XML Recommendations and OASIS Standards indicated by version number are listed below.

	XML Security Standards and W3C Recommendation	Version	Status	Date dd-mm-yy
1	**W3C XML Encryption** http://www.w3.org/TR/xmlenc-core/		R	10-12-2002
2	**W3C XML Digital Signature** http://www.w3.org/TR/xmldsig-core/		R	12-02-2002
3	**W3C XML Key Management (XKMS)** http://www.w3.org/TR/xkms2/		WD	18-04-2003
4	**OASIS Security Assertion Markup Language (SAML)** http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security	1.1	S	02-09-2003
5	**OASIS XML Access Control Markup Language (XACML)** http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml	1.0	S	06-02-2003

WD = Working Draft S = Standard R = Recommendation

Table 3: XML Security Standards

2.8.1. W3C XML Encryption Recommendation

W3C XML Encryption [17] Recommendation specifies the process and the meta-data for encryption and for representing encrypted data as an XML element. XML Encryption is compatible with and may be used in conjunction with XML Digital Signatures [18]. Encryption is used for achieving *confidentiality and access control.*

XML Encryption can be applied to a whole document, or to selected parts of a document. The result of encrypting is placed the XML element called <EncryptedData> that contains the cipher data or identifies reference Uniform Resource Indicator (URI). (The familiar URL associated with http is a subset of URI). The associated meta-data enables a processor of that document to determine when encryption occurred or what algorithm to use to encrypt the document or message. XML Encryption does not introduce any new cryptography algorithm or techniques. DES [19], RSA or Triple-DES [21], AES [22] can still be used. XML Encryption is not a replacement of Secure Socket Layer (SSL) [23] that is used with HTTP.

2.8.2. W3C XML Digital Signature Recommendation

Digital Signature is used for Authentication, Data Integrity and non-repudiation. XML Digital Signature [18] uses a short fixed-length digest that is designed to change if the

content changes. An XML Signature is a Digital Signature expressed in XML. The XML Digital Signature Recommendation allows XML or non-XML document to be signed. Whole or sections of the document can be signed and other sections can be left unsigned. The result of a XML Signature is placed in a XML element. Text and non-text data (e.g. photo or image file) can be signed.

There are 3 different kinds of XML Digital Signature signatures, *Enveloping Signature* (entire data item resides inside the signature block), *Enveloped Signature* (signature resides inside data item), *Detached Signature* (signature resides outside the signed document, e.g. on a different web site). There can be more than one signer. Encryption and hashing technologies are used to make the signature. Multiple documents can be signed with a XML Digital Signature.

XML Digital Signature [18] is versatile and a building block that forms the basis for Web Service Security technologies, such as XKMS [24] (see next section) and Web Services Security. When XML Digital Signature is tied to an identity of a person, a machine or an application, it provides non-repudiation. XML Digital Signatures provide data integrity and authentication, for example data being passed in SOAP [25] (Simple Object Access Protocol) message..

However, if Digital Signature is not implemented with due diligence there could be concerns. For example, when a human sees a data on the screen and signs that data, e.g. $ 2,000 and due to a bug or for malicious reason that may not necessarily be the actual data that gets transmitted to the server. (Instead data $ 200,000 is sent as approval). Such a problem would not exist with paper based signatures.

2.8.3. XML Key Management Specification (XKMS) Working Draft.

XKMS [24] is currently a W3C working draft. It may be considered as the XML interface to PKI [26] services. XKMS simplifies a number of complicated PKI features. In XKMS, PKI services are accessed by sending straightforward XML messages. It defines XML protocol and messages to convey and receive public and private electronic key registration information with other attributes with requests to a *trust server*. The key pair can be represented in different formats; such as by key name, digital certificate or key parameters. XKMS has two parts: X-KRSS: (Registration, recovery, and revocation of keys) and X-KISS (Retrieval of public keys and certificates and validation of certificates). A digital certificate can be registered with an XKMS service using another sub-component of XKMS – X-KRSS (XML Key Registration Service Specification).

2.8.4. OASIS Security Assertion Markup Language (SAML)

The SAML [27] specification from OASIS is to facilitate single “sign-on” capability to access multiple services. SAML describes how information is to be formatted into XML for *assertions* about the end user. Assertions are associated with a given "subject" or named entity. It defines an XML vocabulary for sharing security assertions, including authentication and authorization assertions to enable "single sign-on" and third party management of these functions. It also defines a Request/Response protocol definition and an XML protocol Simple Object Access Protocol (SOAP) [20] binding. SAML allows statements about how and when authentication and authorization occurred to be passed

among parties. It uses unique identifiers (URNs) [28] for different authentication mechanisms and authorization actions and describes how digital signatures are to be associated with assertions.

2.8.5. XML Access Control Markup Language (XACML)

XACML [29] is complementary to SAML [27] that allows access control policies to be expressed in XML. It is required because SAML provides only a mechanism for making authentication and authorization assertions and conveying these assertions by using XML protocol, but vocabulary is also needed for expressing the rules needed to make the authorization decisions. XML vocabulary created specifically for expressing authorization rules is the XML Access Control Markup Language (XACML). It defines XML vocabulary for expressing authorization rules, for expressing a variety of conditions to be used in creating rules, and how rules are to be combined and evaluated. It is a means for creating policy statements, a collection of rules applicable to a subject. XACML uses the SAML definitions for subjects and actions and defines rules as targets effects and conditions. A target includes resources, subjects and actions, as defined in SAML. An effect is either "Allow" or "Deny". Conditions are predicates and attributes defined in the XACML specification.

The advantage of using XML for access control means that policies from various access control products can be replicated easily, using XACML as a common data format.

2.9 Web Services

Applications running on the Internet or Intranet that provide service to the users (humans or machines) can be further facilitated and protected using additional two standards, Simple Object Access Protocol (SOAP) and WS-Security. XML Encryption [17] and XML Digital Signature [18] forms the basis for Web Services Security [30].

2.9.1. SOAP

SOAP [20] is XML based Application to Application protocol. It is a specification for invoking methods on servers, services, components and objects. It uses XML and HTTP as a method invocation mechanism and mandates a small number of HTTP headers that facilitate firewall/proxy filtering. Its is a lightweight loosely coupled protocol for exchange of information in a decentralized and a distributed environment.

2.9.2. WS-Security

WS-Security [30] describes enhancements to SOAP messaging to provide *quality of protection* through message integrity, message confidentiality, and single message authentication. These mechanisms can be used to accommodate a wide variety of security models and encryption technologies.

2.10 Maturity of COTS for implementing XML security

Due to the long learning curve required to experiment with the vendor COTS products and tools, no experimental determination was made to assess the maturity levels of the COTS products. Instead, only the product description and claims made in the vendor document and web site was noted.

2.10.1. Entrust

Entrust offers large numbers of information security products, mainly for PKI environment.

- Entrust Authority™
- Entrust GetAccess™ is a supports broadest range of authentication methods
- Entrust Secure Transaction Platform
- Entrust Entelligence
- Entrust TruePass
- Entrust Certificate Services
- Authentication Products

It was not clear to from the Entrust Web site, as to what extent these product were compatible to XML Security Standards. It is claimed Entrust GetAccess™ support XACML.

http://www.entrust.com/products/index.htm

2.10.2. Vordel

Vordel offers server based XML security product called VordelSecure which is a distributed platform for securing and managing XML and Web Services. It can be used inside and outside the enterprise. It uses open standards, including WS-Security and SAML to provide authentication, authorization, audit, and content validation for XML-based communications.

[Ref: http://www.vordel.com/index.html]

2.10.3. IBM

IBM's XML Security Suite is a tool. It provides security features such as digital signature, encryption, and access control for XML documents. XML Security Suite has 3 XML Security technologies. These are:

- Digital signature implementation based on "XML-Signature Syntax and Processing" by W3C/IETF
- XML encryption implementation based on "XML Encryption Syntax and Processing" by W3C
- XML Access Control Language (XACML) and implementation

[Ref: http://www.alphaworks.ibm.com/tech/xmlsecuritysuite]

2.10.4. Microsoft

Microsoft 's Passport products provide "single sign on" authentication service to perform all their online activities. A user can use single Passport credential among several Web Applications. In this respect, Passport is similar to SAML but Passport is not a standard.

2.10.5. VeriSign

VeriSign provides a tools kit for working with SAML, called Trust Services Integration Kit (TSIK). It provides a platform for creating client/server applications for Web Services. It supports PKI.

[Ref: http://www.verisign.com/]

2.10.6. Baltimore Technologies

Baltimore Technologies offers the following security products.

- Baltimore UniCERT: is PKI based product
- Baltimore KeyTools simplifies development of PKI handling capabilities.
- SelectAccess Version 5.1 provides Authentication and Single Sign-On capabilities

[Ref: http://www.baltimore.com/]

2.10.7. Phaos Technology

Phaos Technology has the followings products:

- Phaos XML – is the core security tools for encryption and signing and is claimed that it complies with W3C. XML Signature and XML Encryption specifications.

- Phaos XKMS is complementary to Phaos XML and provides support for key registration, location and validation and certificates using a trusted Web service.

- Phaos SAML provides XML-based security assertions for entity attributes, authentication and authorization.

- Phaos Liberty provides secure federated network identity thereby enabling single sign-on convenience for users.

[Ref: http://www.phaos.com/products/category/xml.html]

2.10.8. Netegrity

Netegrity has the following security products as COTS.

- *Netegrity SiteMinder:* provides the capability to administer and enforce user authentication and authorization management as well as by providing single sign-on (SSO) to users.
- *IdentityMinder*: is for enforcing enterprise security policies
- *TransactionMinder*: provides policy-based shared services that centralize authentication, authorization, and audit activities for all Web services transactions.

The vendor provides SDK and Client side development tools.

[Ref: http://www.netegrity.com/index.cfm]

2.10.9. Waveset

Waveset has the following products:

- *Lighthouse*™ product for authentication and authorization.
- *Waveset Directory Master for* identity management to manage directory services in the context of an identity infrastructure.

[Ref: http://www.waveset.com/Solutions/Lighthouse/index.html]

2.10.10. Conclusive

Conclusive's TrustLogic product is PKI based.

- TrustLogic conforms to XML encryption standards. It allows users to save their forms locally as encrypted documents. TrustLogic uses standard XSL/T technologies to display the form, and allows for off-line editing of e-forms in standard browsers. Users can download a copy of the form, or work on it off-line and upload it when ready. While they have possession of the form and work on it off-line, all of the cryptographic protection is applied (signature and encryption).

[Ref: http://www.bea.com/framework.jsp?CNT=index.htm&FP=/content/products]

2.10.11. BEA Web Logic Enterprise Security

- WebLogic product provides message-level security for Web services through an implementation of the WS-Security Web service security standard. WebLogic implementation of WS-Security includes secure the Simple Object Access Protocol (SOAP) messages passed between Web services using, Security tokens, Digital Signatures, Encryption.

http://www.bea.com/framework.jsp?CNT=overview.htm&FP=/content/products/security/

2.11 XML Application in the PKI environment

Traditionally, an Application accessible over the Internet is CGI (Common Gateway Interface) based with which a user typically interacts with using a Web Browser (but not necessarily). In such an environment, to safeguard attacks on applications vulnerabilities firewall are used. However, firewalls are considered as weak security mechanisms.

A solution better than firewall is PKI. PKI recommends the use of Encryption for achieving Confidentiality and access control and Digital Signature for Authentication, Data Integrity and non-repudiation. Applications can be treated just like any other information object needing protection and protected under the PKI infrastructure. The deployment of XML Application in PKI environment would normally use of XML Key Management Specification (XKMS) which provides interfaces to PKI and is illustrated in the figure below:

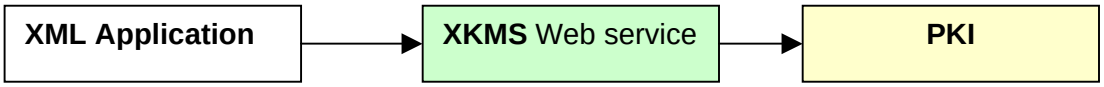


Figure 3 : XML Application in PKI environment

XKMS [24] simplifies interface to complex PKI [26] by moving the complexities of managing the PKI from the application end and transferring it to the XKMS service. XKMS is designed to interface to any form of PKI and can handle different types of certificates, e.g. X.509, SPKI, PGP, etc. Key binding associates the public key credentials additional information such as cryptography, the application protocols that the credentials may use, validity time, etc. The XKMS *locate* function is equivalent to *directory services* associated with PKI and is simpler than X.500 and LDAP.

XKMS enables PKI services such as trust worthily registering, locating, and validating keys through XML-encodes messages. In XKMS, the PKI services could be accessed by sending and receiving straightforward XML messages and therefore would not requirement for yet another Toolkit, such as SDK (mentioned earlier) to implement the security feature. Both PKI and XKMS deal with managing multitudes of keys.

XKMS like XML Signature eliminates the need for ASN.1 [34] functionality in software that deals with digital certificates. The difference between PKI and XKMS may be conceptualized by thinking that PKI is certificate orientated whereas XKMS is key

orientated. XKMS allow XML software to use digital certificates and PKI without the need for cryptography algorithms.

3. REPORT ON PHASE-II WORK (POC)

In this phase of the research, XML based application, such as, *eforms* (electronic forms) was selected as an application for the proof on concept (POC). Vendors that have used XML in their eform product design were briefly examined.

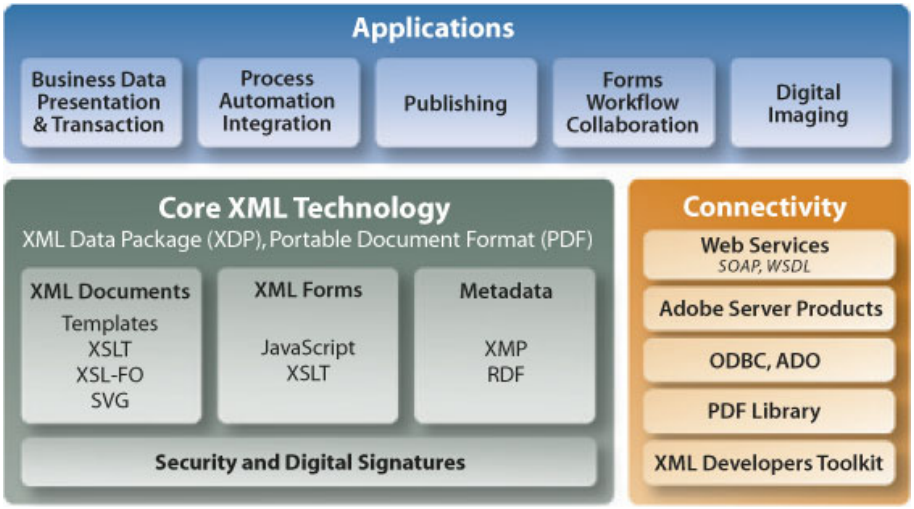
3.1 Some eform COTS products

The eform COTS products include, Cardiff Software™, PureEdge™, and TrustLogic™ (from Conclusive) and InfoPath 2003 (Microsoft Office). The features of these products were noted from XML perspective. For example, it was observed that Microsoft's InfoPath forms is built around XML Schema [9] or Web Services Description Language (WSDL) [35] definitions and are capable of outputting XML documents conforming to an XML Schema Definition (XSD) [9] or communicating with a Web Services.

XForm [32] is new W3C standard that was released on 14 October 2003 which is an XML application that represents the next generation of forms for the Web.

3.2 Examination of Adobe Acrobat Version 6

The Adobe Acrobat™ Professional version 6.0 includes eform capability. It was readily available at DRDC was selected for simple experimentation on it's security features. The product architecture is shown in the figure below:



--Courtesy http://www.adobe.com/enterprise/xml.html

Figure 4: Adobe Acrobat XML Architecture

3.2.1. Acrobat Security features

The above figure shows that various Acrobat Applications use “Core XML Technology” that includes an underlying Security and Digital Signature mechanism. Besides the use of password [Ref: Acrobat manual page 261], the product has a variety of security features that include *Digital Signatures and Encryption* [Ref: Acrobat manual page 265] *and Digital Certificates.*

In the Acrobat Professional version 6, the Digital ID can be obtained from Default Certificate Security, Windows Certificate Security, and Third Party Certificate Security. The product provides features to build a list of trusted identities. When a self-signed digital ID is created using Default Certificate Security, the resulting file stores an encrypted private key used for signing or encrypting documents, a public key contained in a certificate used for validating signatures, and a time-out value if a password is required for signing. Text string cannot be searched in an encrypted PDF document.

Acrobat PDF eform has Digital signature field [Ref: manual page 110]. PDF document can be signed more than once and by more than one person and sections can be locked or marked as read-only. Selective security includes password protected restricted editing, printing, copying of text and images.

However, it was not clear from the vendor’s manual if these features were compliant to the set of W3C/IETF or OASIS XML security standards

3.2.2. Acrobat Version 6 from XML Technology prespective

From the XML technology viewpoint, it can be noted that Acrobat Professional version 6 uses XML based languages called XML Data Package (XDP) and Extensible Meta-data Platform (XMP). A XDP is simply a file that packages PDF file in an XML file. XDP files are XML files that contain XML form data, XML form templates, PDF documents, and other XML information. The architecture of XDP file is shown in the figure below:

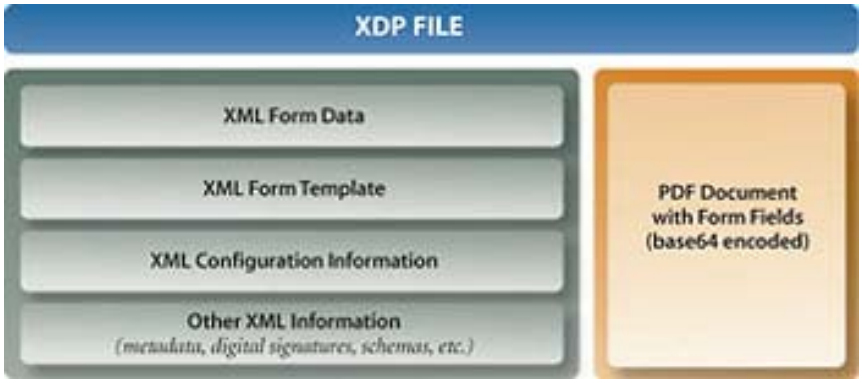


Figure 5: Adobe Acrobat eForm architecture

XDP and XMP provide applications with common XML framework standardizes the creation, processing, and interchange of document meta-data across publishing

workflow. The meta-data can include security information. Acrobat 5.0 or later contains document meta-data in XML. Document meta-data can be extended and modified using third-party products. Security is a feature of document property.

4. CONCLUSION

1. XML and the core families of XML Recommendations have no mark up or tags reserved for information security, but have rich features that can be exploited to develop variety of security mechanisms. W3C, OASIS and the Apache Group are using the features in XML to develop XML based security mechanism, standards and services.

2. XML Security comprises of (i). W3C XML Encryption, (ii). W3C XML Digital Signature, (iii). W3C XML Key Management Specification (XKMS), (iv). OASIS Security Assertion Markup Language (SAML), (v). OASIS XML Access Control Markup Language (XACML). There are additional standards/recommendations for security services.

3. XML COTS Security software requires may require use of vendor supplied Software Development Kit (SDK) and tools for deployment and configuration. Present XML Security COTS products will need to be updated to make them compliant to W3C and OASIS standards.

4. XML Security COTS software are not necessarily interoperable and most requires that software from the same vendor be installed at both the client and sever end.

5. Adobe Acrobat Professional version 6, eForm allows multiple signatures. The encryption is applicable to the entire document only and it is not possible to encrypt only a part of the eForm. The security features are not fully compliant to W3C and OASIS security standards.

6. More research and experimentation is required to determine the reliability, compatibility to XML security standards and interoperability of XML based COTS security products and eforms.

(The new W3C XForm Recommendation released in October 2003 needs to be explored and compared to presently available COTS eForms from various vendors to determine the extent to which COTS are compliant to W3C XForm)

5. REFERENCES

[1]. World Wide Web Consortium http://www.w3c.org

[2]. Extensible Markup Language (XML) 1.0 (Second Edition)
W3C Recommendation 6 October 2000 http://www.w3.org/TR/REC-xml

[3]. International Organization for Standardization (ISO)
http://www.iso.ch/iso/en/ISOOnline.openerpage

[4]. Semantic Web http://www.w3.org/2001/sw/

[5]. Information processing -- Text and office systems -- Standard Generalized Markup Language (SGML)
http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=16387

[6]. UNICODE http://www.unicode.org/standard/WhatIsUnicode.html

[7]. Namespaces in XML http://www.w3.org/TR/REC-xml-names/
http://xml.coverpages.org/xml-spec-report20.html

[8]. DTD (Document Type Definitions) http://www.w3schools.com/dtd/

[9]. XML Schema http://www.w3.org/TR/xmlschema-0/;
http://www.w3.org/TR/xmlschema-1/
http://www.w3.org/TR/xmlschema-2/

[10]. W3C XML Path Language (XPATH) http://www.w3.org/TR/xpath

[11]. W3C XML Pointer Language (XPOINTER) XPointer element() Scheme 25 March 2003, Paul Grosso, Eve Maler, Jonathan Marsh, Norman Walsh XPointer Framework 25 March 2003, Paul Grosso, Eve Maler, Jonathan Marsh, Norman Walsh XPointer xmlns() Scheme 25 March 2003, Steven J. DeRose, Ron Daniel Jr., Eve Maler, Jonathan Marsh

[12]. W3C XML Linking Language (XLink) Version 1.0
(XLINK) 27 June 2001, Steven DeRose, Eve Maler, David Orchard

[13]. W3C XSL http://www.w3.org/TR/xsl/

[14]. W3C XSLT http://www.w3.org/TR/xslt

[15]. OASIS (Organization for the Advancement of Structured Information Standards)
http://www.oasis-open.org/who/

[16]. Liberty Alliance http://www.project-liberty.org/

[17]. XML Encryption Syntax and Processing
10 December 2002, Donald Eastlake, Joseph Reagle
http://www.w3.org/TR/xmlenc-core/

[18]. XML-Signature Syntax and Processing
12 February 2002, Donald Eastlake, Joseph Reagle, David Solo
http://www.w3.org/TR/xmldsig-core/
http://www.w3.org/TR/xmlenc-decrypt
http://www.w3.org/TR/xmldsig-filter2/

[19]. Data Encryption Standard http://www.itl.nist.gov/fipspubs/fip46-2.htm

[20]. Ronald L. Rivest, Adi Shamir, Len Adelman, "*On Digital Signatures and Public Key Cryptosystems*," MIT Laboratory for Computer Science Technical Memorandum 82 (April 1977).

[21]. Triple DES http://www.rsasecurity.com/rsalabs/faq/3-2-6.html
http://www.tropsoft.com/strongenc/des3.htm

[22]. Advanced Encryption Standard (AES) http://csrc.nist.gov/CryptoToolkit/aes/

[23]. Secure Socket Layer http://wp.netscape.com/eng/ssl3/draft302.txt

[24]. XML Key Management Specification (XKMS) http://www.w3c.org/TR/xkms2/

[25]. Simple Object Access Protocol (SOAP)
Part 0 Primer; http://www.w3.org/TR/2003/PR-soap12-part0-20030507/
Part 1: Messaging Framework, http://www.w3.org/TR/2003/PR-soap12-part1-20030507/; Part 2: Adjuncts; http://www.w3.org/TR/2003/PR-soap12-part2-20030507/
Assertions and Test Collection.
http://www.w3.org/TR/2003/PR-soap12-testcollection-20030507/

[26]. Pubic Key Infrastructure (PKI) http://csrc.ncsl.nist.gov/pki/

[27]. Security Assertion Markup Language (SAML)

http://www.oasis-open.org/committees/download.php/1371/oasis-sstc-saml-core-1.0.pdf
http://www.oasis-open.org/committees/download.php/1376/oasis-sstc-saml-schema-assertion-1.0.xsd
http://www.oasis-open.org/committees/download.php/1377/oasis-sstc-saml-schema-protocol-1.0.xsd
http://www.oasis-open.org/committees/download.php/1372/oasis-sstc-saml-bindings-1.0.pdf
http://www.oasis-open.org/committees/download.php/1374/oasis-sstc-saml-conform-1.0.pdf

[28]. URN RFC-2141 http://www.ietf.org/rfc/rfc2141.txt

[29]. XML Access Control Markup Language (XACML) 1.0 Specification OASIS Standard. http://xml.coverpages.org/ni2003-02-11-a.html

[30]. Web Services Security (WS-Security)

http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss
http://msdn.microsoft.com/library/?url=/library/en-us/dnglobspec/html/ws-security.asp

[31]. DARPA Agent Markup Language (DAML) http://www.daml.org

[32]. W3C XForm http://www.w3.org/TR/xforms/

[33]. Resource Description Framework (RDF)

http://www.w3.org/TR/rdf-primer/
http://www.w3.org/TR/rdf-syntax-grammar/
http://www.w3.org/TR/rdf-mt/
http://www.w3.org/TR/rdf-schema/

[34]. ASN.1 (Abstract Syntax Notation) ISO Standard
http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=32298

http://renoir.vill.edu/mnt/a/cassel/html/netbook/asn1only/node1.html

[35]. Web Services Description Language (WSDL),

http://xml.coverpages.org/ni2003-06-12-a.html

APPENDIX A: SYNOPSIS OF XML CORE W3C RECOMMENDATIONS

XML

XML 1.0 specification [Ref: http://www.w3.org/TR/REC-xml.html] is a document marking language that was derived as a subset from the Standard Generalized Markup Language (**SGML**, formally ISO 88791), an international standard for electronic document exchange.

XML documents are tree structured that start with a root node and several branches can emanate for a branch. XML defines rules for structuring information in a document with pairs of opening and closing tags to encapsulate words, sentences, paragraphs, chapters and even the whole document. The tags can imply meaning. Unlike HTML, which has fixed number of tags to indicate the format of the document, e.g. <b> for **bold,** and <i> for *italics*, in XML the users can define their own set of tags. An example of pair of XML tags could be <city> Toronto </city>. There is no limit to the number of tags in XML. XML tags are intended to imply meaning rather than the format of the document. XML documents are aimed to be machine-readable and machine comprehensible, where pairs of XML tags could provide clues to the meaning in the document.

The successful transfer of information or meaning from a source (writer or originator) to a destination (reader), require that the sets of XML tags must be mutually agreed a prior between the sender and receiver. However, XML tags alone are not sufficient to enable machines to determine the semantics contained in the document.

There are no explicit security features defined in the XML 1.0 specification, or in the draft XML 1.1 specifications. There are no markups reserved to indicate security. However, since XML tags are user definable, XML tags could be defined by the XML document author to devise proprietary security schemes.

Namespace are used to distinguish between similar names. In XML, it used to distinguish between similar XML tags and define the context in which sets of tags are used. There are no explicit security features in the Namespace specification. However, the user can define secure namespaces.

XML-SCHEMA

XML-Schema provides greater features than DTD for structuring the XML document and for defining the different data types that are valid or invalid in the various sections and fields of the XML document. The format of the XML document can be validated by the reader (by human or machine) if at first, the Schema of the particular XML document is agreed upon between the writer and the reader. XML Schema Recommendation is described is 3 part document

W3C Recommendation	Status	Date yy/mm/dd	Source Document
XML Schema Part 0: Primer	R	01/05/02	http://www.w3.org/TR/xmlschema-0/

XML Schema Part 1: Structures	R	01/05/02	http://www.w3.org/TR/xmlschema-1/
XML Schema Part 2: Datatypes	R	01/05/02	http://www.w3.org/TR/xmlschema-2/

Table 4 : W3C Recommendation for XML Schema

Although security is not a built-in part of the Schema specification, a user can define any field in the XML document to be a security field. XML Schema could be used to devise various security mark-up in the document.

The XML Security standards define XML vocabularies using XML technologies, such as XML Schema, for definition. An example is the ***<KeyInfo>*** element defined in the XML Digital Signature recommendation for carrying signing or encryption.

XPATH

XPATH [Ref: http://www.w3.org/TR/xpath] is a language for addressing parts of an XML document. An XML document may not necessarily reside as a whole at just one location. Different parts of one XML document may distributed in other web sites and servers on the Internet or Intranet, or in different sectors of the hard disk.

It is designed to be used by both XSLT and XPointer (described in later sections). There are no explicit security features built in the XPATH specification. However, the XML Security standards (e.g. *XML Digital Signature Recommendation*) are taking advantage of XPATH features. The XML Digital Signature Recommendation allows XPATH expression to extract portions for processing as indicated in XMLDigSig.

XLINK

XLINK [Ref: http://www.w3.org/TR/xlink/] is XML Linking Language, which allows elements to be inserted into XML documents in order to create and describe links between resources. Unidirectional and multi-directional links can be specified.

In HTML, links can be easily broken when the document is changed. In XML, links are in a database. Because XLink allows the creation of such "third-party links,"
We can expect products to be created that consist solely of huge databases of links, with no content actually provided by the link authors.

The unidirectional and multi-directional link can be used to devise security features. For example, one or more of the multi-directional link could be decoy link and if taken, intrusion may be detected and logged by a hacker.

XPOINTER

XML Pointer or XPOINTER is used for referencing to points and ranges inside the XML document. It is the language that is to be used as the basis for fragment identifier for any URI (URL is subset of URI) reference that locates a resource.

The XPOINTER Recommendation dated 25 March 2003, is described in 3 parts:

W3C XPOINTER Recommendation	Status	Date yy/mm/dd	Source Document
XPointer Framework	R	03/03/25	http://www.w3.org/TR/xptr-framework/
XPointer element() Scheme	R	03/03/25	http://www.w3.org/TR/xptr-element/
XPointer XML Namespace Scheme	R	03/03/25	http://www.w3.org/TR/xptr-xmlns/

Table 5: W3C Recommendations for Xpointer

Unlike HTML, where link transition can take place only to the predefined (anchor) point, in XML the XPointer mechanism allows transition to sections of the document where no specific anchor points was provided by the author. An example would be to say access the fifth line, 3rd word after certain anchor. Xpointer allows unidirectional, bi-directional, and multi-directional links.

No inherent security mechanism is built into the Xpointer specification. However, novel use of Xpointer can be made by the author of the XML document to construct proprietary security schemas, an example could be multidirectional links to passwords.

RDF

RDF (Resource Description Framework) is about semantics and will play a very an important part in the development of the envisioned Semantic Web. It is still a developing ad Working Draft and is not a released as a recommendation. RDF is about meta data that will facilitate intelligent search on the web. It is a universal format for data on the Web. Using a simple relational model, it allows structured and semi-structured data to be mixed exported and shared across different applications.

The working draft consists of the following parts:

W3C Working Draft	Date yy/mm/dd	Document Source
RDF Primer	03/01/23	http://www.w3.org/TR/rdf-primer/
RDF/XML Syntax Specification	03/01/23	http://www.w3.org/TR/rdf-syntax-grammar/
RDF Semantics	03/01/23	http://www.w3.org/TR/rdf-mt/
RDF Vocabulary Description Language 1.0: RDF Schema	03/01/23	http://www.w3.org/TR/rdf-schema/
An RDF Schema for P3P (Platform for Privacy Preferences)	02/01/25	http://www.w3.org/TR/p3p-rdfschema/

Table 6 : W3C Working draft for RDF

Complex sophisticated form of information security development is possible with RDF.

APPENDIX B: XML SECURITY STANDARDS

Synopsis of XML Encryption

The XML Encryption [Ref: http://www.w3.org/TR/xmlenc-core/] W3C Recommendation defines an XML vocabulary and processing rules enabling confidentiality to be applied to a variety of contents. Secure sockets layer (SSL)/ transport layer security (TLS) or virtual private networks (VPNs) only provide confidentiality while the information is in transit, not while it is stored at a server. XML Encryption serves the purpose of maintaining the confidentiality of information, both while in transit as well as when stored. The XML Encryption recommendation defines the framework and processing rules for XML encryption and decryption.

1. XML and non-XML content may be encrypted
2. Confidentiality may be applied at a fine level of granularity to XML content. It may be applied to XML elements and XML element content as well as entire XML documents.
3. XML Encryption produces well-formed XML from well-formed XML. This allows portions of XML content to be encrypted yet subsequently processed by XML tools.
4. XML Encryption is compatible with and may be used in conjunction with XML Digital Signatures.
5. XML Encryption allows for encryption of a symmetric key that may be packaged with encrypted content.
6. XML Encryption supports a variety of encryption algorithms and techniques.
7. When an XML element or element content is encrypted, it is replaced by an <EncryptedData> element.
8. When non-XML content is encrypted, the result is a new XML document containing an <EncryptedData> element.
9. An <EncryptedData> element may include a Type attribute to assist the recipient in decrypting it. This Type may indicate that an XML element or element content was encrypted, or give the type of other information, such as images for example. This is done using an existing standard for mail attachments, known as MIME types.
10. The <EncryptedData> element defines the algorithm used for encryption, provides the encrypted content, and may include information necessary to determine the key needed for decryption.
11. The symmetric key used to encrypt content may be conveyed in an <EncryptedKey> element.
12. XML Encryption supports the selection of appropriate encryption algorithms and defines XML identifiers for common cases and may be extended for others.
13. Definitions for identifying key information are based on XML Digital Signature definitions and extended.
14. User-defined properties may be associated with an encrypted element, such as a timestamp or log reference.
15. The actual cipher text, the result of encryption, is specified using a <CipherData> element. This may contain the actual encrypted data within a <CipherValue>

Synopsis of Digital Signature in XML

Digital signatures [Ref: http://www.w3.org/TR/xmldsig-core/] are useful for two purposes:

1. To provide persistent content integrity, and
2. To create and verify portable electronic signatures

The XML Digital signature recommendation defines mechanisms to support the full range of digital signature creation and verification, including the ability to sign and verify:

1. Entire XML documents as well as element and element content portions of XML documents,
2. Arbitrary documents, including binary documents,
3. Compound documents including multiple documents and/or XML elements and element contents,
4. Properties to be included with a signature,
5. Counter-signatures (signatures that include other signatures)

In addition, the XML Signature recommendation supports the application of multiple XML Signatures to an XML document or to different sections of a document, supporting a variety of use cases. An XML <Signature> element may be handled in different ways, based on the desired application. It may be placed in a document apart from what is signed. This is known as a "detached" signature, and is used when signing non-XML content. When XML content is signed, the <Signature> element may be added to the XML. When placed in an XML document, the <Signature> element may be added to the document being signed under the document element (an "enveloped" signature).

The following concepts are central to understanding XML digital signatures:

A signature is only valid if the signed content has not changed. This content is represented using a short, fixed-length digest, designed to change if the content changes. Thus, a signature will only be valid if a digest used to create a signature is the same as a digest used to verify it later. A verifier can create a digest to see if it is the same.

An XML <Signature> element is an XML structure that contains a cryptographic signature value in a <SignatureValue> element as well as an XML structure that has been signed, the <SignedInfo> structure. This means that the contents of the <SignedInfo> structure should not change for the signature to be valid.

The signer creates a <Reference> for each item to be included in a signature. Each <Reference> includes a digest of the item and a unique identifier (URI) for the item. It also identifies how to recreate the digest, specifying the algorithm and other necessary information. Each Reference is part of the <SignedInfo> structure.
To verify a signature, a recipient must validate each <Reference> by independently generating the same digest for the item. The verifier may use the URI to aid locating the item and the algorithm information to know how to generate the digest. If the item has not changed, the digest should be the same.

A reference may refer to anything using a URI, including non-XML content such as image and text files. It is not required to obtain the item using the URI, but it is often useful. A special form of URI may be used to refer to XML elements within the same document as the signature, allowing signatures to be transferred along with XML content to be signed.

A <Reference> may specify one or more transforms to be applied to an item before creating the digest. One use is to sign parts of an XML document that are known not to change - such as boilerplate for example. This may be done by defining transform to extract the portion of the document to be signed, using standard XML XPath expressions for example.

Digest algorithms require content to be the same to produce the same digest. Even a minor change that does not change the meaning, such as adding an extra space, will invalidate the digest. XML, on the other hand, allows some variation in the syntax of the XML text without changing the document. In other words, two XML documents may be considered the same even if they do not have the exact same text. For example, one XML document may use single quotes for an attribute and another double quotes. These are the same to an XML parser, but very different to a digest algorithm. There is an entire list of such potential issues for digests. To get around this problem, a Canonicalization transform may be used, one that converts any XML document to a form using a single set of rules, such as always using a certain type of quote for attributes.

Synopsis of Security Assertion Mark-up Language (SAML)

Security Assertion Markup Language (SAML) was developed by Baltimore Technologies, BEA Systems, Computer Associates, Entrust, HP, Hitachi, IBM, Netegrity, Oblix, OpenNetwork, Quadrasis, RSA Security, Sun Microsystems, Verisign, and other members of the OASIS Security Services Technical Committee. SAML 1.0 is an XML based framework for exchanging *authentication* and *authorization* information.

SAML allows authentication information to be shared by creating an assertion that a subject was authenticated in a specific manner at a specific time. Different techniques for establishing identity are supported, ranging from use of a password to use of hardware tokens and personal physical attributes (biometrics). It allows assertions to specify what type of authentication mechanism was used.

To interact with a Web service, a program sends a SOAP request message, which is a type of XML document.

Comparing SAML and WS-Security

To protect confidentiality, WS-Security relies on XML Encryption, while SAML uses the slower HTTPS. WS-Security protects individual transactions, and the substantial infrastructure required by SAML pays off with single sign-on capability.

The Liberty Alliance's authentication solution—Liberty 1.0—builds on SAML, while Microsoft's competing technology, .NET Passport, uses WS-Security. No matter whether these two standards converge or remain separate, the success of Web services in e-business could depend on them. (OASIS, www.oasis-open.org.)

The request/response language expresses queries about whether a particular access should be allowed (requests) and describes answers to those queries (responses).

The SAML standard consist of the following parts:

OASIS Standard	Date yy/mm/dd	Source Document
Assertions and Protocol	02/11/05	http://www.oasis-open.org/committees/download.php/1371/oasis-sstc-saml-core-1.0.pdf
Assertion Schema		http://www.oasis-open.org/committees/download.php/1376/oasis-sstc-saml-schema-assertion-1.0.xsd
Protocol Schema		http://www.oasis-open.org/committees/download.php/1377/oasis-sstc-saml-schema-protocol-1.0.xsd
Bindings and Profiles	02/11/05	http://www.oasis-open.org/committees/download.php/1372/oasis-sstc-saml-bindings-1.0.pdf
Conformance Program Specification	02/11/05	http://www.oasis-open.org/committees/download.php/1374/oasis-sstc-saml-conform-1.0.pdf

Table 7: OASIS SAML

Summary

1. SAML is an XML vocabulary for expressing authentication and authorization assertions, allowing statements about how and when authentication and authorization occurred to be passed among parties.
2. A request response protocol for conveying SAML assertions, as well as an XML protocol (SOAP) binding.
3. Unique identifiers (URNs) for different authentication mechanisms and authorization actions.
4. How digital signatures are associated with assertions.

Synopsis of XACML

Although SAML provides a mechanism for making authentication and authorization assertions and conveying these assertions using XML protocol, a vocabulary is also needed for expressing the rules needed to make authorization decisions. One XML vocabulary created specifically for expressing authorization rules is the XML Access Control Markup Language XACML

XACML is a security standard for expressing XML *policies* for information access over the Internet. Using XACML, developers can enforce policies for information access. It was ratified as an OASIS Open Standard by the OASIS in February 2003. It defines a

core schema and corresponding namespace for the expression of authorization policies in XML against objects that are themselves identified in XML. It is an access control policy language request / response language.

Summary

1. XACML is XML vocabulary for expressing authorization rules
2. An XML vocabulary for expressing a variety of conditions to be used in creating rules.
3. How rules are to be combined and evaluated
4. A means for creating policy statements, a collection of rules applicable to a subject.
5. XACML uses the SAML definitions for subjects and actions
6. XACML defines rules as targets, effects and conditions
7. A target includes resources, subjects and actions, as defined in SAML
8. An effect is either "Allow" or "Deny".

Synopsis of XML Key Management Specification (XKMS)

The XML Key Management Specification (XKMS), is a W3C Working Draft and is listed below:

W3C Working Draft	Date yy/mm/dd	Document Source
XML Key Management Specification (XKMS)	03/04/18	http://www.w3.org/TR/xkms2/
XKMS Binding	03/04/18	http://www.w3.org/TR/xkms2-bindings/
XML Key Management Specification Bulk Operation (X-BULK)	02/03/18	http://www.w3.org/TR/xkms2-xbulk/

Table 8: XKMS spec documents

XKMS defines protocols for Public Key management services. It takes advantages of the Web services framework for inter-application communication using public key infrastructure (PKI). XKMS is a base specification for secure Web services. It enables Web services to register and manage cryptographic keys used for digital signatures and encryption. KMS defines XML message formats to support requests and responses for public key management, including registration, revocation and updates.

Summary

1. XKMS defines XML protocol messages to convey key registration and information requests to a *trust server* and to convey responses from the server. The specification defines the binding of these messages to the XML Protocol (SOAP). It defines the relationships among the messages using the Web Services Definition Language (WSDL).

2. XML <ds:KeyInfo> element processing is delegated to the trust service by the client, minimizing the complexity of the client. How the trust service is implemented is dependent on the service.

3. Registration supports the requirements of smart card manufacturing, including bulk processing and pending responses.

4. The specification supports the use of XML Digital Signatures for message integrity and authentication. The specification also defines other authentication mechanisms, support for proof of key ownership and other security functionality.

5. A Locate or Validate request may include a <KeyInfo> element and <RespondWith> element in the request. The <RespondWith> element is used to specify what the <KeyInfo> element is to be resolved to, possibly more than one item. For example, the request <KeyInfo> might contain an X.509 certificate and the <RespondWith> might indicate that the KeyName and KeyValue are to be returned. Possibilities include KeyName, KeyValue, and Certificate, Certificate Chain (collection of certificates needed to trace a signature back to a trusted party) among the possibilities outlined in the specification.

6. A <KeyBinding> element is used to associate information with a key. This is what is returned in a Locate or Validate response. Every <KeyBinding> includes a <ValidityInterval> (NotBefore, NotOnOrAfter) and may also include <KeyInfo>, <ProcessInfo> (opaque data), <KeyUsage> and <UseKeyWith> elements.

7. Key usage definition is deliberately limited to Encryption, Signing and Key Exchange.

8. A <KeyBinding> <UseKeyWith> element defines which application and application entity the key is intended for. For example, a key may only be appropriate for authentication of an SSL server. In this case, the application is HTTPS, and the identifier is the URL of the server. Applications listed in the specification include S/MIME, HTTPS, SMTP, IPSec, PKIX and others.

APPENDIX C: XML SECURITY SERVICES AND PROTOCOLS

Synopsis of SOAP (Version 1.2 May 07, 2003)

SOAP (Simple Object Access Protocol) is XML based Application to Application protocol. It is a specification for invoking methods on servers, services, components and objects. SOAP codifies the existing practice of using XML and HTTP as a method invocation mechanism. The SOAP specification mandates a small number of HTTP headers that facilitate firewall/proxy filtering. The SOAP specification also mandates an XML vocabulary that is used for representing method parameters, returns values, and exceptions.

On 7 May 2003, W3C announced the advancement of SOAP Version 1.2 to Proposed Recommendation.

W3C Document SOAP	Date yy/mm/dd	Status	Source
Part 0 Primer	03/05/07	PR	http://www.w3.org/TR/2003/PR-soap12-part0-20030507/
Part 1: Messaging Framework,	03/05/07	PR	http://www.w3.org/TR/2003/PR-soap12-part1-20030507/
Part 2: Adjuncts,	03/05/07	PR	http://www.w3.org/TR/2003/PR-soap12-part2-20030507/
Assertions and Test Collection.	03/05/07	PR	http://www.w3.org/TR/2003/PR-soap12-testcollection-20030507/

Table 9: SOAP Spec Documents

Synopsis of WS-Security

Web Services Security (WS-Security) is an improvement and an extension of SOAP proposed by IBM, Microsoft and VeriSign to improve the quality of protection. It defines how to extend SOAP to provide integrity and confidentiality and how to include security tokens in messages. This includes defining how to encode binary formats, including X.509 certificates and Kerberos tickets.

WS-Security describes how Applications can construct secure SOAP messages and includes use of:

- several types of existing encryption
- multiple tokens and certificates
- multiple trust domains
- digital signature

- end to end message level security
- error classification & reporting

WS-Security is not a complete security specification but it can be considered to be a building block from which a range of security protocols can be constructed. The security elements defined are User Name Token Element, Encoding Binary Security Token, Security Token Reference, ds:KeyInfo, ds:Signature, & Encryption.

To protect confidentiality, WS-Security relies on XML Encryption, while SAML uses the slower HTTPS. WS-Security protects individual transactions, and the substantial infrastructure required by SAML pays off with single sign-on capability.

Some of the concepts defined in WS-Security include:

- *Principal*: Person, application or business entity that can send or receive web service messages

- *Claim*: A statement (or assertion) about a subject that associates the subject with a property, such as the subject's identity, authorization, or other information. A claim may be made by a subject or some other party.

- *Token:* A token is a representation of security related information and may be used to represent and substantiate a claim. A token may be unsigned (such as a shared secret password used to support an identity claim) or signed (such as a PKIX Identity certificate, a Kerberos ticket, or an authorization certificate). Use of an unsigned token may require secure transport such as provided by SSL/TLS or a VPN.

Having a token is often not enough - a signature is also required to demonstrate proof of possession of material associated with a token. An X.509 certificate, for example, may serve to demonstrate the binding of an identity with a public key, but including a signature using that private key may provide proof of ownership.

Synopsis of Web Service Description Language (WSDL)

"WSDL (Web Service Description Language) is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information" – source: W3C

WSDL is described in a set of recent documents (06 June 2003) listed below.

W3C Working Draft	Date yy/mm/dd	Document Source
Web Services Description Language (WSDL) Version 1.2 Part 1: Core Language	03/06/11	http://www.w3.org/TR/wsdl12/
Web Services Description Language (WSDL) Version 1.2 Part 2: Message Patterns	03/06/11	http://www.w3.org/TR/wsdl12-patterns/
Web Services Description Language (WSDL) Version 1.2 Part 3: Bindings	03/06/11	http://www.w3.org/TR/wsdl12-bindings/

Table 10 : WSDL

The XML Security standards define XML vocabularies for representing security information, using XML technologies, such as XML Schema, for definition. An example is the <KeyInfo> element defined in the XML Digital Signature recommendation for carrying signing or encryption.

Universal Description Discovery & Integration (UDDI)

Universal Description Discovery & Integration (UDDI) is the definition of a set of services supporting the description and discovery of (1) businesses, organizations, and other Web services providers, (2) the Web services they make available, and (3) the technical interfaces which may be used to access those services. Based on a common set of industry standards, including HTTP, XML, XML Schema, and SOAP, UDDI provides an interoperable, foundational infrastructure for a Web services-based software environment for both publicly available services and services only exposed internally within an organization.

[Ref: http://uddi.org/pubs/uddi_v3.htm]

Synopsis XML Application in PKI environment

PKI (Public Key Infrastructure) is an archirecture for security on the Internet.
It language independent of platform and does not depend on XML.

PKI is based on the use of public essential cryptography and digital signatures. It is a framework of policies, services, and encryption software that provides the assurances users need before they can confidently transmit sensitive information over the Internet and other networks.

Public key cryptography is used to ensure the confidentiality of sensitive information or messages by using a mathematical algorithm, or key, to scramble (encrypt) data, and a related mathematical key to unscramble (decrypt) it. In public key cryptography, authorized users receive special encryption software and a pair of keys, one an accessible **public key**, and the other a **private key**, which the user must keep secret. The two keys are related so that a message encrypted with a user's public key can only be decrypted using the corresponding private key.

A **Certification Authority** (CA) is a main component of a PKI. It is a trusted third party responsible for issuing digital certificates and managing them throughout their lifetime.

Digital certificates are electronic files containing the user's public key and specific identifying information about the user. They are tamper-proof and cannot be forged. A **digital signature** is an electronic identifier comparable to a traditional, paper-based signature - it is unique, verifiable, and only the signer can initiate it. Used with either encrypted or unencrypted messages, a digital signature also ensures that the information contained in a digitally signed message or document is not altered during transmission.

Certificate management is the strength of a PKI's Certification Authority. Around the world, enterprises large and small are adopting Public Key Infrastructures as their preferred solution for enabling the centralized creation, distribution, management, renewal and revocation of certificates.

OASIS has formed a technical committee to advance PKI (public key infrastructure) adoption for Web services and other applications.

APPENDIX D: FEW WEB SITES ON XML and WEB SECURITY

http://www-106.ibm.com/developerworks/library/ws-secure/

http://www.brics.dk/~amoeller/XML/linking/

http://www.secinf.net/websecurity/The_World_Wide_Web_Security_FAQ/

APPENDIX E: XFORM AND TUTORIALS

XFORM TUTORIAL http://www.w3schools.com/xforms/xforms_intro.asp

W3C XForm http://www.w3.org/TR/xforms/

6. ACRONYMS

ACRONYMS	
DAML	DARPA Agent Markup Language
DARPA	Defense Advanced Research Projects Agency
DRDC	Defense Research and Development Canada
HTML	HyperText Markup Language
IT	Information Technology
OASIS	Organization for the Advancement of Structured Information Standards
P3P	Platform for Privacy Preferences
PKI	Public Key Infrastructure
RDF	Resource Description Framework
RDF-S	RDF-Schema
RDF-S	RDF-Schema
SAML	Security Assertion Mark up Language
SOAP	Simple Object Access Protocol
W3C	World Wide Web Consortium
WSDL	Web Service Description Language
WS-S	Web Service Security
XACML	eXtensible Access Control Markup Language
XKMS	XML Key Management Specification
XLink	XML Linking Language
XML 1.0	Extensible Mark up Language version 1.0
XML-S	XML Schema
XPATH	XML Path Language
XPOINTER	XML Pointer Lanuage
XSL	Extensible Style sheet Language
XSLT	Extensible Style sheet Language Transformations

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.)

Signal Stream Inc.
580 Wilkie Drive Orleans,
Ontario. K4A 1N3
CANADA

2. SECURITY CLASSIFICATION (overall security classification of the document including special warning terms if applicable).

UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C,R or U) in parentheses after the title).

Enhancing the Security of Applications using XML-based Technologies

4. AUTHORS (Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)

Ahmed, Syed I

5. DATE OF PUBLICATION (month and year of publication of document)

December 2003

6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc).

43

6b. NO. OF REFS (total cited in document)

35

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered).

Contractor Report

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include address).

DEFENCE R&D CANADA - OTTAWA
3701 Carling Avenue, Ottawa, Ontario, K1A 0Z4

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Specify whether project or grant).

15BF27

9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written).

W7714-3-08506

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique.)

DRDC Ottawa CR 2003-201

10b. OTHER DOCUMENT NOs. (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
( ) Defence departments and defence contractors; further distribution only as approved
( ) Defence departments and Canadian defence contractors; further distribution only as approved
( ) Government departments and agencies; further distribution only as approved
( ) Defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution beyond the audience specified in (11) is possible, a wider announcement audience may be selected).

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

XML and associated core family of XML based W3C Recommendations have no built-in information security features but have rich characteristics that can be exploited to devise numerous security schemes. The evolving XML Security Standards and COTS tools that can enhance the security of applications are identified.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title).

XML Technologies, Web Services, Security Services, e-forms