

# XML-Based Security – A New Challenge to I.T. Protection

U. Rahman, J. Lu and Jim Yip

School of Computing and Engineering, University of Huddersfield, UK, HD1 3DH

## Abstract

There has been a sudden rise in the use of XML and the development of tools and specifications using XML such as web services. Contemporary solutions were found to be lacking facilities to secure XML based applications. This paper is a review of XML security specifications. A comprehensive literature review was conducted to assess and summarize the validity of current and contemporary tools and technologies. The literature review showed that there is a void as far as academic papers are concerned. From the available articles on computer science 10% are based on computer security and of that 4% are on XML Security. There is, now, a dire need for the academics, professional bodies and industry to speed up the development of XML security.

## 1. Introduction

XML has been widely accepted as a platform independent format for data-representation. Information can, now, readily be stored in XML format. Microsoft products, such as Word, Excel, Visio and Infopath allow documents to be stored in XML format [1]. Microsoft has added detailed XML support in its 2005 edition of SQL Server [2].

There has been a sudden rise in the use of XML and the development of tools and specifications using XML. The XML family has been expanded by professional bodies such as W3C, OASIS, IETF and ISO. The industry has liaised with these professional bodies to develop much needed security solutions using XML. The specifications developed cover the following fields of information security: data security, access control and key management.

The key method deployed to obtain information is searching by keywords on information providing websites such as google, Ingenta, metalib and science direct. This survey is a systematic approach to review the current situation in information security as compared to the solutions proposed by the new tools and specifications developed using XML. The known limitations to the reviewed tools have also been discussed.

This paper is organized as follows. Section 2 focuses on the contemporary tools and technologies and their limitations. Section 3 details the current tools and specifications

developed using XML. Section 4 contains results and discussions based on the literature review. The last section details the conclusions derived from the research.

## 2. Contemporary Tools and Technologies

### 2.1 SSL

Secure Sockets Layer (SSL) is a protocol developed by Netscape for transmitting confidential documents over the internet. SSL uses cryptography for two keys to encrypt data, a public key known to everyone and a private key only known to the recipient [3]. This technology ensures a secure connection between two parties to promote data exchange. Many websites use this protocol to obtain private information such as credit card numbers. But information is rarely hacked or stolen while it is being transported over the internet, it is stolen either from the source or destination.

Another protocol for transmitting secure data over the internet is Secure HTTP (S – HTTP). SSL is used for sending any amount of data securely, while S-HTTP is used for sending individual messages securely. Therefore, SSL and S-HTTP are complementary technologies to each other [3]. Both have also been approved by the Internet Engineering Task Force (IETF) as a standard. However, applying SSL or S-HTTP for web services is a different issue.

### Limitations in SSL

SSL, firstly, is designed to protect data from one source to one destination, where as in normal internet transactions there are multiple destinations, or multiple route points. The problem in SSL arises in the situation when a web service is in use. SSL ensures transport level confidentiality, which the web service would use to authenticate and authorize the incoming SOAP request, which in turn means the web service has no visibility of the user, just the other side of the communication (Website) [3].

Secondly, SSL is not data aware; it encrypts the entire document for transport, transports it and then decrypts the whole document for everyone to see.

And lastly SSL was not designed for the complex security needs of web services and hence does not cater for them either.

XML data passed through SSL is particularly vulnerable due to fact that it is smart and content aware information. Without knowing XML, the data is still human readable. So when sensitive information is being passed in an SSL tunnel, it is vulnerable at all points that it is stored on: networks and databases. If the employees were all a 100% trustworthy in the industry, there would be no security issue here, but the FBI estimates that 70% of all thefts are inside jobs [4], carried out behind the firewall and not through it.

## 2.2 IPSec

Internet Protocol Security (IPSec) was designed by the Internet Engineering Task Force (IETF) as the security architecture for the IP and is accepted as an official standard. The first draft for IPSec dates back to April 1995 the core of which focuses on machine to machine tunnelling and encryption for network traffic by defining IP packet formats. The security architecture is based on two modes: transport and tunnel mode. Transport mode is used between two end hosts only. Tunnel mode is used when one of the end points is a gateway (an intermediate system such as a router that implements the IPSec functionality) [5]. It provides data privacy, integrity and authenticity for IP traffic in the following cases [6]:

1. End to end security between client and server transactions using the transport mode.
2. Security in remote access from client to gateway over the internet using Layer Two Tunnelling Protocol (L2TP).
3. Security in gateway to gateway connections across private wide area networks (WAN).

IETF has also defined the Internet Key Exchange (IKE) for IPSec. IKE has three authentication methods to enable trust between interacting computers: Kerberos v5.0 authentication, public/private key signatures using certificates, and passwords [6].

### Limitations in IPSec

IPSec tunnel mode is not designed for Virtual Private Networks (VPN) remote access, since VPNs rely on routers to forward packets to next hop IP addresses and the tunnel mode specification is ambiguous on this issue [5]. Once packets are encrypted they cannot be modified and hence the destination IP cannot be modified, which conflicts with the idea of intermediate routers [5 – 7].

IPSec encrypted packets cannot be compressed. Compression will typically do nothing but it reduces the data size and hence increases the data in a packet and promotes quicker transport [5].

Nested tunnels create very complicated structures and normally users concatenate IPSec tunnels instead which are protected by separate keys [5]. This means the encrypting and decrypting end points are potential attacking sites and due to complicated key management in this case traffic will experience heavy overhead [5].

Multicasting is one of the few things that are well suited on IP and one of the few things that IP does better than other protocols [6]. IPSec and IKE are point to point and hence they do not suit well to multicasting [6].

Putting security mechanisms on the network layer has not been popular as the network functions are problematic for security; IPsec tries to do it and is at least partially successful. However, so far IPsec has still some unsolved issues.

## 2.3 Public Key Infrastructure – PKI

In the basic sense, PKI was designed to provide for a third party vouching for a user identity. It also allows the binding of a public key, in the form of a certificate, to a user. The term PKI is used to refer to both the certificate authority along with the relative arrangements and the use of public key algorithms, which are the most commonly used methods on the internet for user authentication [8].

PKI enables users of an unsecure public network such as the internet to securely exchange information through the use of a public and private key pair that is obtained and shared through a trusted authority. A PKI from any vendor would consist of the following components [9]:

- Certificate authority that issues and verifies digital certificates. The certificate consists of the public key or relative information to the public key.
- Registration authority that verifies the certificate authority.
- Directory/directories where the certificates are held.
- Certificate management system.

PKI methods serve to provide for the following security functionalities [8]:

- Encryption and authentication of Emails.
- Encryption and authentication of documents.
- Authentication of users.
- Bootstrapping secure communication protocols such as SSL. The initial setup for a SSL tunnel uses asymmetric key methods whereas the actual communication uses symmetric key methods.

XML security specifications cannot replace PKI, although they can work congruently to produce a secure computing environment.

Since PKI was not designed for web services, it can be used in conjunction with XKMS.

### 3. Current Technologies

#### 3.1 Key Tools and Technologies

##### 3.1.1 XML Encryption – XML Enc

This specification for the latest version of XML Enc was developed and is licensed by World Wide Web Consortium (W3C) on 3 October 2002 [10]. The team of authors included employees from IBM, Microsoft and XMLSec Inc.

**Source-Level Encryption:** Within an XML source document, data content is described by its corresponding element tag which is defined inside the DTD or schema. A secure tag can be defined and placed around elements, which needs privatization, such that authoring tools can encrypt the data inside secure tags. Now, when an application processes the XML document and reads the secure tags it will decrypt the data using a key provided along with the source document (key management will be discussed later in the report) [10]. Source level encryption is useful when information transfer is rare and the content that needs to be encrypted is known. However, for documents that are transferred frequently, this approach requires a lot of time to secure individual parts inside each source document [11].

**Meta-Level Encryption:** Meta-level encryption involves defining a secure attribute within the DTD or Schema. Using this method the secure tag does not need to be defined in every occasion that data in a document needs securing. Instead, an element's secure attribute will simply need to be set to true.

**Security Sheets:** Just like an XML document references style sheets, it may reference security sheets. Security sheets would contain encryption rules for the source document. This way we would separate the encryption rules from the actual information [11]. Therefore, the XML document would reference the DTD and the security sheet, but for the information to be secured by the security sheet, it needs to be valid in accordance with its DTD.

##### 3.1.2 XML Key Management System - XKMS

XKMS was a joint effort between Microsoft, VeriSign and webMethods, proposed to W3C on 30 March 2001 [12].

Every time a secure XML document is transferred to a consumer, an encryption key is also transferred separately through another process. This key is used to decrypt the

encrypted information. Key management is to control the: generation, filing, modification, use and destruction of keys. The better the key management, the better the security of the information; when the key is available, the information is available [12].

Detail on XKMS can be found in the specification [12]

##### 3.1.3 XML Digital Signature XMLDSig

The key to enable secure transactions is the concept of digital signatures, which ensure the integrity and authenticity of the origin of documents [13].

The XML Signature specification was created at W3C and the contributors were Microsoft, JetForm, PureEdge and Entrust on the 12<sup>th</sup> of October 2000 [10].

XML signatures use XML syntax and are applied to any digital content including XML itself. There are three distinct signature types [13]:

Detached	Over data external to the signature document.
Enveloped	Signature within content being signed.
Enveloping	Content within signature being signed.

Detailed information and specification on XML Signatures can be found in [13].

##### Limitations of XML Digital Signature

XML, due to its extensible format, is inherently unstable and hence un-signable. XML signatures are an attempt to forcefully sign un-signable data. At the most basic syntax level, ignoring the semantic problems for now, text canonicalization needs to deal with white spaces, line endings, word wrapping etc. X.509 spent its first 10 years to achieve canonicalization but failed and gave up. Even then, the X.509 canonicalization rules are simpler than the XML ones. The semantic level XML is ten fold the complication rank of syntax level XML for canonicalization. The subjects that need to be dealt with include XSLT, XPath, external forces such as style sheets, schemas and DTDs, namespace declaration and attributes and many more.

There is only one correct structured way to encapsulate data cryptographically. For this, signed data would be in the following sequence:

```
Signature algorithm indicator;
Data;
Signature;
```

And for encrypted data it would be:

```
Recipient/key-exchange information;
Encrypted data;
```

When the XML security developers realised the irrevocable flaws in XML Digital Signature, they decided to reinvent XML to make it securable. What they came out with in the end was complete syntax extensibility to the extent where the following encapsulations were allowed:

*Data;*  
*Signature algorithm indicator;*  
*Signature;*

And:

*Encrypted data;*  
*Recipient/key-exchange information;*

### **3.1.4 Extensible Access Control Markup Language – XACML**

XACML is an OASIS standard, entirely written in XML [14]. It is a general-purpose access control policy language. It provides a definition for policies dealing with access rights and request of access rights by describing two languages [15]:

**Policy language:** Used to describe general access control requirements. It has standard extension points that allow for defining aspects such as new functions, data types, combining logic etc.

**Request/response language:** allows the construction of a query to ask whether or not a given action should be allowed and then interpret the result. The answer is one of the following four: Permit, Deny, Indeterminate (Not enough data available or an error has occurred) and Not Applicable (This service does not deal with the request in question).

Typically, a user will need to access a resource and they will make a request to whichever entity protects that resource. This resource is called the Policy Enforcement Point (PEP). The PEP will create a request based upon the available information on the request. This PEP is sent to the Policy Decision Point (PDP), which will compare the request to the applicable policy and return an answer as to whether access should be granted [14]. This answer is returned to the PEP, which enforces the answer i.e. allow or deny access to the user.

The detailed XACML specification can be found in [14].

### **3.1.5 Security Assertion Mark-up Language – SAML**

SAML is designed for the exchange of authentication, attribute and authorization information across security domains. This security information is expressed in the form

of assertions about subject, where a subject can be a person or a computer [16, 17].

Assertions are issued by SAML authorities, who are: authentication authorities, attribute authorities and PDPs. SAML has a protocol that clients use to request assertions from SAML authorities and get responses from. This protocol can be bound to other transport protocols; currently SAML defines only one binding, to SOAP over HTTP [16].

SAML authorities can compile together assertions through various sources such as security policies and assertions received through requests and then use these to create their response. Hence, SAML authorities are not only consumers they are also producers of assertions.

One of the major goals for SAML is the setup of a Single Sign-On – SSO environment [16], which promotes the authentication of the user in one domain but allow access in other domains without re-authentication.

Industrial giants are homing in on the idea of developing new protocols for federated identity management. SAML is part of this new class of protocols and will be used in the near future between businesses to reduce user management costs [17].

SAML is an open standard and is very extensible. This attribute is utilised by the Liberty Alliance Project and the Shibboleth project; both these projects are built on SAML [17]. The Liberty Alliance Project has had many public proposals but to date has not gone through a standardization process, its protocols have also been found to be vulnerable and have been attacked on. The Shibboleth Project is an interuniversity federation project built on SAML. This project has proposed another protocol: Browser based attribute exchange – BBAE (concentrates on attribute exchange and privacy issues), which will also be built on SAML [17].

The IBM research laboratory in Zurich carried out a security analysis of SAML SSO and found it vulnerable [17]. OASIS has recently admitted to be working on the weaknesses stated in IBM's research report [17].

A detailed SAML specification can be found in [16] and a security analysis of SAML can be found in [17].

### **3.1.6 XML Common Biometric Format – XCBF**

XCBF was approved as a standard by OASIS in August 2003 [18]. Biometrics are automated methods of authenticating a person based on physical or behavioural characteristics. These can be used to verify a claimed identity. XCBF defines encrypted messages in XML for

transactional biometric information [18]. XCBF provides a means for data integrity, authentication and privacy of biometric information in XML based applications [19]. The mechanisms defined in the specification include secure transmission, storage, integrity, and privacy protection for biometric information.

The cryptographic processing requirements are defined in XCBF to support XML encryption and decryption [19].

XCBF can be used along with web services and WS-Security to provide additional security [19]. The integration of biometrics with web services can achieve security requirements further. On the other hand it is important to maintain web services simplicity while adding the complexities of biometrics. It will take some time for this specification to be in wide use, but the importance of not increasing vulnerabilities while introducing XCBF cannot be stressed upon enough [19]. XCBF can possibly provide a new range of cost effective and secure applications.

The detailed specification can be found at [18].

### 3.1.7 XML Advanced Electronic Signatures – XadES

XadES is a submission made to W3C in February 2003 and has not achieved the recommendation status yet. XadES is an extension to the XMLDSIG specification, which consolidates it by adding features that were not originally offered by the XML Signature specification [20].

XadES defines XML formats for advanced electronic signatures that remain valid over long periods of time. It specifies two main types of properties: signed properties and unsigned properties. Signed ones are additional data objects that are secured by the original signature, implying that the signer has these data objects. Unsigned ones are data objects added by the signer, by the verifier or third parties after the creation of the original signature. These are not secured by the original signature but can be secured by another signature produced by a third party.

XadES has three primary forms:

<b>XadES</b>	Some additional signed and unsigned properties.
<b>XadES-T</b>	Adds a time stamp to the signature for long term validity.
<b>XadES-C</b>	Adds to the XadES-T references to the data supporting the validity of the electronic signature.

The detailed specification can be found at [20].

### 3.1.8 Web Services Security – WS Security

In April 2002, IBM, Microsoft and VeriSign published a new web services security specification, Web Services Security (WS-Security). This specification was approved as a standard in March 2004 by Organization for the Advancement of Structured Information Standards (OASIS) [21]. There are some 36 member companies/individuals of the WS-Security specification and 15 companies that are participating in its inter-operability tests. Sun Systems, Microsoft and IBM have released developer kits that support WS-Security specification.

WS-Security supports, integrates and unifies several popular security models, mechanisms and technologies [21]. It defines a standard set of SOAP extensions. These message headers can be used to implement confidentiality, secure data exchange and signed messages. It provides for a generic method to associate security tokens with messages. XML contains some primitive security features that were utilized by WS-Security to ascertain clandestine transactions: XML Signature, XML Encryption and XKMS.

IBM and Microsoft, in a joint effort, have created a road map consisting of other security specifications that can be used alongside WS-Security. These specifications include the following [22]: WS-Policy, WS-Trust, WS-Privacy, WS-Secure Conversation, WS-Federation and WS-Authorization.

## 4. Results and Discussions

The information reviewed for this survey is classified in to the following categories:

- Tool – A description of self-developed tool or a proposal for a tool.
- Theory – A paper containing theories and algorithms to resolve current issues.
- Descriptive Overview – A description of the theory and methodology behind existent tools or technologies.
- Critical Overview – Pros and cons of existent tools or technologies.



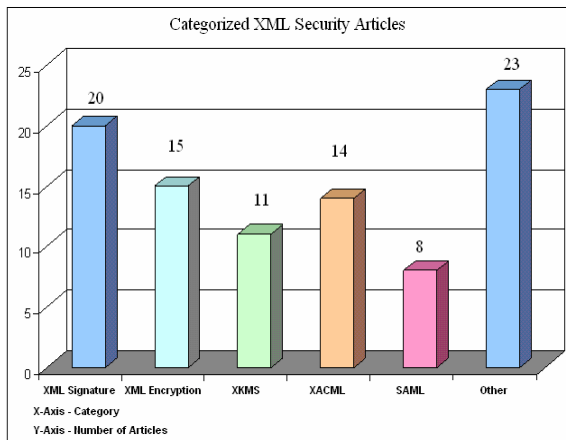
**Figure 1 Pie Chart: Categorized XML Security Articles.**

The literature review (Figure 1) proves that the research being carried out today in the field of XML security is focussed on developing tools and producing theories. The tools and theories in this field are relatively new and have not been evaluated as such, hence the shortage of critical overviews.

Computer Science	Computer Security	XML Security
13,987	1,459	53

**Table 1 Number of articles published relative to computer security and XML security.**

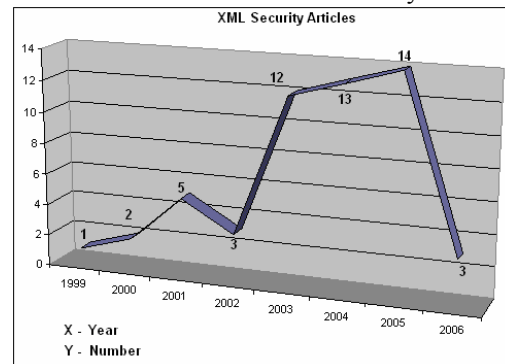
From the available articles on computer science 10% are based on computer security and of that 4% are on XML Security. The figures in table 1 show that the computer security field is not focussed on XML Security to the degree that it should be. The specifications that have been recognized as standards, XML Signature, XML Encryption, XACML and SAML, are not widely in use. The articles on XML security are further categorized in to the specifications they review in figure 2.



**Figure 2 XML Security Articles Categorized according to their content.**

According to current trends (Figure 2) in academia, only 3.7% of papers on computer

security are based on security specifications provided through XML. 22 % XML security papers are on XML signature making it the most popular specification among other specifications (XML Encryption 17%, XKMS 9%, XACML 23% and SAML 15%). This is because XML Digital Signature is a standard and a W3C specification. On the other hand, the internet utilizes these signatures and the number of internet users has increased phenomenally over the past few years, giving excessive rise to the use of signatures. The other technologies are relatively new or have not achieved recommendation status yet.



**Figure 3 XML Security Articles organized by publication year.**

A general review of the papers reveals that most of these articles were written and published in recent years (Figure 3). Research in XML security is a developing field that is gaining rapid popularity.

According to information collected the most work carried out by any industrial organisation is Microsoft, since .Net (A Microsoft product) uses XML for SOAP messages. After Microsoft, IBM is in second place; both are dedicating many employees to the task of achieving the goals of their road map for web services security. Microsoft (.NET) and IBM (WebSphere) are currently making use of these specifications in their products. IBM's WebSphere uses the web services technology and supports WS-Security.

## 5. Conclusions

After the review of contemporary tools and technologies, it can be deduced that there is a dire need for XML/web services, especially since XML is human readable. The professional bodies have stepped up to the challenge and have produced many security specifications, although the majority are still not standards or recommendations. The industry has also contributed whole heartedly to the development of specifications. The applications, however, developed by the industry do not deploy XML security to its full potential. Instead of using XML security

specifications, they are making use of their self developed solutions, which does not help the professional bodies in achieving recommendation status for their specifications. The academic world, on the other hand, is slowly turning to this field, but the advances are coming in small quantities. There is a need for critical evaluations of XML security tools and specifications.

## References

1. Pal, S. et al. (2005), XML Support in Microsoft SQL Server 2005 [online] Available at: <<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsq190/html/sql2k5xml.asp>> [Accessed March 2006].
2. Homer, A. (2005), SQL Server 2005 XQuery and XML-DML [online] Available at: <<http://www.15seconds.com/issue/050803.htm>> [Accessed March 2006].
3. VORDEL (2005), SSL and Web Services – Myths and Facts [online] Available at: <[http://www.vordel.com/knowledgebase/vordel\\_view4.html](http://www.vordel.com/knowledgebase/vordel_view4.html)> [Accessed December 2005].
4. Forum Systems (2002) SSL: Not Enough for Today's Web Services [White Paper].
5. Dunbar, N. (2001) 'IPsec Networking Standards — An Overview' Information Security Technical Report, Vol 6, No. 1: pp. 35-48.
6. Stanton, R. (2005) 'Securing VPNs: Comparing SSL and IPsec' Computer Fraud & Security, September 2005: pp. 17-19.
7. Molva, R. (1999) 'Internet security architecture' Computer Networks, Vol. 31, pp. 787-804.
8. Farrell, S. and Zolotarev, M. (2001) 'XML and PKI - What's the story?' Network Security, vol. 2001, no. 9, pp. 7-10.
9. Selkirk, A. (2001) 'Using XML security mechanisms' BT Technol J, vol. 19, no. 3, pp. 35 – 43.
10. Haas, H. (2005), Seminar. [XML Security: Signature, Encryption, and Key Management] [online] Available at: <<http://www.w3.org/2004/Talks/0520-hh-xmlsec/>> [Accessed December 2005].
11. Brandt and Bonte (2000), Towards Secure XML [online] Available at: <[http://lists.w3.org/Archives/Public/xml-encryption/2000Oct/att-0016/02-Discussion\\_paper\\_sXML.doc](http://lists.w3.org/Archives/Public/xml-encryption/2000Oct/att-0016/02-Discussion_paper_sXML.doc)> [Accessed December 2005].
12. Ford et al. (2001), XML Key Management Specification [online] Available at: <<http://www.w3.org/TR/xkms/>> [Accessed December 2005].
13. Simon et al. (2001), An Introduction to XML Digital Signatures [online] Available at: <<http://www.xml.com/pub/a/2001/08/08/xmldsig.html>> [Accessed December 2005].
14. Godik, S. and Moses, T. eds. (2003) eXtensible Access Control Markup Language (XACML) Version 1.0. OASIS Standard.
15. Ardagna, C.A. et al. (2004) 'XML-based Access Control' Information Security Technical Report, vol. 9, no. 3: pp 35 – 46.
16. Hallam-Baker, P. and Maler, E. eds. (2002) Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML). OASIS Standard.
17. Gross, T. (2003) Security analysis of the SAML single sign-on browser/artifact profile In: IEEE Computer Society. 19th Annual Computer Security Applications Conference (ACSAC 2003): IEEE Computer Society Press.
18. Larmouth, J. ed. (2003) XML Common Biometric Format (XCBF). OASIS Standard.
19. Rragami, L. and Edwards, N.H. (2003) 'Securing web services with biometrics' Biometric Technology Today, pp. 6 – 8.
20. Cruellas, J.C. et al. (2003) XML Advanced Electronic Signatures (XAdES). W3C Note.
21. Organization for the Advancement of Structured Information Standards (2004) Web Services Security:2004 SOAP Message Security 1.0. OASIS.
22. Apshankar, K. (2002), WS-Security: Security for Web Services [online] Available at: <<http://www.webservicesarchitect.com/content/articles/apshankar04.asp>> [Accessed December 2005].