Issues with XML-Signature Syntax and Processing, and Rectifying Approaches

14 August 2007

Jeff Hodges <Jeff.Hodges@NeuStar.biz>
Scott Cantor <cantor.2@osu.edu>

Introduction

Many people, often implementors, sometimes protocol designers, have an aversion to employing the technology specified in "XML-Signature Syntax and Processing"[1], hereafter referred to in this memo as "XMLdsig"

Others have written about various issues with this specification (and XML itself as a protocol message encoding format), for example Peter Gutmann's "Why XML Security is Broken,"[2] so we will not belabor details, or XML itself, here. Rather we will sketch a skeleton outline of issues (not all-encompassing), and briefly enumerate some approaches to rectifying them, with the hope that something more universally useful will be attained.

Issues

Some specific issues we wish to discuss are:

1 In terms of implementor uptake, the SAML community[3] has experienced push-back by various folks, who are working on web single sign-on, because SAML employs XMLdsig and there (at the time) were not commonly available open-source XMLdsig implementations for "scripting languages" (e.g. Perl, Python, PHP, Ruby). Plus, even if this were mitigated in some fashion, it was maintained that performing the XMLdsig-based signatures were expensive processing-wise and prone to interoperability issues.

2 In terms of issues with the XMLdsig specification itself from an implementor's perspective, there's items such as..

2.1 Lack of examples of how to reference signed elements from enveloped <ds:signature> elements.

2.2 The tradeoffs between employing "ID" and XPath techniques to referencing signed elements are unclear. For example, with the Xpath-based technique, one cannot possibly know what's been signed without recomputing and reparsing the reference.

---

1 *XML-Signature Syntax and Processing*. Eastlake, Reagle, Solo, et al. W3C Recommendation, IETF RFC 3275, 12-Feb-2002. <http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/>

2 *Why XML Security is Broken*. Peter Gutmann, University of Auckland, New Zealand, October 2004. <http://www.cs.auckland.ac.nz/~pgut001/pubs/xmlsec.txt>

It is worth noting that Peter's suggested approach of <PGP></PGP> and <SMIME></SMIME> tags wouldn't necessarily work as well or as easily as he contends, and that there exist today nominally interoperable XMLdsig implementations (this doument is now almost three years old). However, that is not to say having to employ the notion of complex canonicalization is not an issue in and of itself. Nor is it to say that there are not various interop issues with the use and processing of XML-encoded messages and digital signatures thereof. For example, we feel his argument with respect to one's inability to create generic XML security toolkit has merit.

3 OASIS Security Services (SAML) TC. <http://www.oasis-open.org/committees/security/>

2.3 The text on the use of RetrievalMethod is somewhat misleading and implies that what's being referenced is an actual child element of KeyInfo, but since it's an ID-based reference, and only KeyInfo carries an Id attribute, some have always (rightly or wrongly) assumed it was supposed to point at a KeyInfo, not a child.

2.4 Some real-world deployments (e.g. SAML-based) find using X.509-based PKI cumbersome, and find using "bare" public (and private) keys, as opaque blobs, a workable simplifying strategy. Though, the public and private key syntax employed in XMLdsig assumes knowledge of, and access to, the underlying mathematical quantities of which such keys consist, making it tough to migrate to the former approach.

Approaches

Some rectifying approaches for the above issues:

R1. With respect to issue 1, above: standardize some sort of generalized "sign-the-BLOB" approach for those situations where it is warranted and workable. Generally, such an approach is workable when the XML-encoded data is being placed in some other data structure, aka "packaging", that is itself encoded using something other than XML. An example of employing this type of technique is demonstrated in the "SAMLv2.0 HTTP POST "SimpleSign" Binding"[4] draft specification. It is worth noting that this technique is somewhat similar to Peter Gutmann's overall suggested approach.

R2. For issues 2.1, 2.2, and 2.3, enhance the specification appropriately.

R3. For issue 2.4, define an additional standard KeyInfo child element for carrying base64-encoded RSA or DSA public key(s). This approach would nominally entail simply leveraging the "key block" format used by OpenSSL, which appears to be the PEM (Privacy Enhanced Mail) message format defined in RFC 1421[5]. For example, here is such a public key block:

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxwIYr7YlgsxG5WKPaOeb
KPEDx2tYeRiHrPlGQ0FQHhCh9gnLmyw0+95fQ/c5bN9M4FW+hWFEqR6OiQr+z6Kt
bqSf0Bdi7fqCn7ShfToaD/1h3SRAaD2h+nqqtJxmWNBXcl7aoZWULPxMDLHzNUxu
BuMfGBKuMHR3AsHIMYKJTKH3euNFKUIqpf3AViXGGGe/l8VLR7dgaym/hFKO8ycH
mTwVY8Jgenwxfg+ecyKfrHR+zRX5XfLiceHsZhMUhysmiY87RJ6nrBpYktQmy3Us
Bc3ifMhdTDqU1RTGulf4r4OxaZ1Sp2h/MK3O0aUeqzhmDbKTmvz8oeYAKA8emQW9
UwIDAQAB
-----END PUBLIC KEY-----
```

4 *SAMLv2.0 HTTP POST "SimpleSign" Binding*. Hodges, Cantor. OASIS Security Servicecs TC, Working Draft, 14-Aug-2007. <http://www.oasis-open.org/committees/download.php/24974/draft-sstc-saml-binding-simplesign-03.pdf>

5 *Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures*. Linn. IETF RFC 1421, February 1993. <http://www.ietf.org/rfc/rfc1421.txt>