

## W3C Workshop on Next Steps for XML Signature and XML Encryption

### Position Paper

Sean Mullan

Sun Microsystems

[sean.mullan@sun.com](mailto:sean.mullan@sun.com)

### Abstract

This position paper is focused on what we believe is one of the most significant issues affecting usage of XML Signature and Encryption in the industry today: performance. We shall briefly describe some performance and scalability issues with respect to implementations of XML Signature and Encryption, and then outline some of the challenges related to implementing a one-pass, or streaming implementation of XML Signature.

### Overview

Many early implementations of XML Signature and Encryption were (and still are) based on the Document Object Model (DOM). DOM was chosen as it provided capabilities that made it suitable for supporting XML Signature and Encryption, such as the ability to navigate the document in any direction and to easily represent XPath node-sets .

However, DOM requires an in-memory representation of the document, which can be a major factor that affects performance. The same capabilities of DOM that made it attractive for implementing XML Signature and Encryption also made it unattractive for certain applications of XML Signature and Encryption, such as those that need to validate/decrypt large messages, those that run in a constrained environment, or those where performance, scalability and throughput are of paramount importance, such as Web Services Security.

To address these performance problems, a streaming, or one-pass implementation of XML Signature and Encryption is a solution that should provide better performance and reduced memory footprint. However, it is difficult or perhaps impossible to implement a streaming, general purpose XML Security library without imposing various restrictions on the algorithms and structures supported, or by adding additional buffering or caching of data objects such that you lose many of the overall performance or memory reduction benefits.

### One-Pass Implementation Challenges

Several profiles of SOAP Digital Signature [1, 2] have attempted to address these performance issues by imposing restrictions on the XML Signature algorithms, structures and data to facilitate one-pass processing.

Here is a list of potential problems and challenges with developing a one-pass implementation of XML

## Signature:

1. Local data objects (the referenced content that will be transformed and digested) can be located anywhere in the document, which is problematic. To facilitate one-pass validation processing, all local data objects that are to be digested should appear after (forward references) the `Signature` element in the document, though some minimal caching (ex: the digest and transform algorithms and input parameters) is still required. Data objects that appear before (backward references) the `Signature` element are problematic because the validation application has not parsed the `Signature` element yet and therefore does not know the location (fragment identifier) of the data objects or the algorithms and parameters that are needed to transform, canonicalize and digest the data.

Thus, validation of enveloped and detached signatures with backward references to local data are difficult to support.

2. Unfortunately, one-pass generation of signatures with forward references has the opposite problem as validation. More specifically, you cannot generate the signature until you have processed the data objects. Therefore, additional caching or an extra pass may be unavoidable when generating the signature.
3. The `KeyInfo` element occurs after the `SignedInfo` element. This is problematic because the key that is needed to verify the signature over the `SignedInfo` element may depend on processing the `KeyInfo` contents. Thus, caching of the bytes to be verified is usually required before the signature can be verified.
4. Canonicalization algorithms that depend on ancestor context (namespaces, “xml:” attributes, etc) are difficult to support in a streaming fashion.
5. Some transform algorithms are not streaming compatible (for example, those that need to navigate the document in any direction such as XPath) and are difficult or impossible to support in a streaming implementation.

## Conclusion

To conclude, we would like to see the next revision of XML Signature (and Encryption, if necessary) support a restricted form that facilitates one-pass implementations. Other standards such as PKCS #7 [3] and PGP [4] recognized the importance of this feature and are designed to support one pass processing of signatures.

In general, a more restricted, simpler form of XML Signature and Encryption is desirable as many applications do not need all the flexibility and features that are required today.

Also, we have noticed that XML Signatures often contain redundant information. For example, a `Signature` element may contain several `Reference` elements with the same set of `DigestMethod` and `Transform` algorithms. A simpler form of XML Signature that eliminated this redundancy would help improve performance by reducing the size of the messages and the amount of data that needs to be processed and canonicalized.

## References

- [1] W. Lu, K. Chiu, A. Slominski, D. Gannon, [“A Streaming Validation Model for SOAP Digital Signature”](#)

- [2] A. Nadalin, et al., [“SOAP Message Security: Minimalist Profile \(Mprof\)”](#)
- [3] B. Kaliski, [RFC 2315: PKCS #7: Cryptographic Message Syntax Version 1.5](#)
- [4] J. Callas, L. Donnerhackle, H. Finney, R. Thayer, [RFC 2440: OpenPGP Message Format](#)