Making Reliable Web Services Message Exchanges Secure and Tamper Proof

Alan J Weissberger

Data Communications Technology

aweissberger@sbcglobal.net

I. Composability of WS Reliability with WS Security

IBM, Microsoft, and their partners have coined the term "*composability"* in their definition of Web Service specifications. The term has been accepted throughout the industry and now also applies to emerging Web Services standards (e.g. WS-Reliability and WS-Security) which may be combined to co-operatively work with each other.

We use the term *composability* to describe independent standards or specifications that can be combined to provide more powerful capabilities. WS middleware providers can support composed capabilities by integrating two (or more) WS standards in a specified way in the same or different SOAP header processing nodes, e.g. providers can integrate WS Reliability support for communicating WS Security message exchanges (or vice versa). The purpose is to combine two independent WS standards or specifications to realize the desired functionality that each provides in a single set of WS message exchanges.

NEC, Fujitsu and Hitachi have recently submitted a new document to the **OASIS WS-RM Technical Committee (TC)**, for review as a possible Application Note or Technical Report on use of WS-Reliability (see Reference 1) together with the WS-Security standard. Both standards are needed if web services are to extend beyond the firewall and used for company to company, SOAP/XML based transactions. By composing WS-Reliability and WS-Security, it is possible to construct Web Services which are both secure and provide reliable message delivery. ***This is vital for the success of Web Services in eCommerce/ eBusiness environments, where the integrity and reliability of external communications is of utmost importance.***

The referenced "Composability Guideline of WS-Reliability and WS-Security" document is available for free download from:

http://www.oasis-open.org/committees/documents.php?wg_abbrev=wsrm

(Click on the link of 10 Jan 2005 submitted by Alan Weissberger)

This new document describes several ways to make WS-Reliability and WS-Security work together and describes XML implementations for each. The purpose of the document is to show that it is possible to use both specifications together to construct secure and reliable Web Services. The composability aspects (i.e. combining header processing functionality WS-Reliability and WS-Security) are illustrated by considering the proper combination of their respective features. **The focus is on making Reliable Message Exchange Patterns**

secure and tamper proof. [Making Security Message exchanges reliable is a potential study item for the OASIS WS- Security TC.]

As noted in Reference 1, the **WS-Reliability** standard provides various levels of message delivery: **Guaranteed Delivery, Duplicate Elimination and Guaranteed Message Ordering.** The **WS-Security** standard (see Reference 2.) provides message integrity and message confidentiality. ***Both of these specifications are vital for use of Web Services over the public Internet or even Intranets.*** However, there has been no definitive description of how they both could be used in the transport of Web Services (SOAP/XML) messages.

Key assumption: the new document assumes that both WS-R and WS-Security are contained in the same SOAP processing node, with a combined implementation where each entity knows of the existence of the other. Also assumed, is that the WS-Reliability payload may be encrypted, but hot the headers. This is to permit the use of intermediaries, such as routers, which would not possess encryption keys for the multiple pair-wise SOAP/XML message exchanges that they relay between the WS end points. Making all the WS Security message exchange patterns reliable is outside the scope of the document (e.g. encryption key management or requesting a security token are not considered). Which Security Message Exchange Patterns should be made reliable is an item for further study for the either the WS-Security TC or the WS-I Security WG (see reference 2.).

II. WS-RM TC Work plan for Composability of WS-R with WS-Security

The **WS-RM TC** has just agreed on a two phase approach to the Composability of WS-Reliability (WS-R) with other WS standards/specs:

- **Part 1: High level "Composability Concepts"** document will describe: composability model(s), baseline assumptions for SOAP HDR processing entities, composability requirements/ functionality, implications, etc. Some examples and illustrations of ways to compose will be provided. Implementation details will be contained in part 2.

This document is targeted at a wide audience, which includes Web Services (WS) architects, WS implementers that are using modular WS-R middleware (open source or otherwise), and technical decision makers (that probably have not participated in the WS-RM TC and are not intimately familiar with the WS-R standard).

- **Part 2. Composability Case Studies for WS-R and WS-Security**

  Detailed model and implementation guide for selected methods to compose WS-R and WS-Security, based on high level principles enumerated in Part 1. "Composability Concepts" document

This document is aimed at WS implementers that are working with WS-R and WS-Security middleware (open source or otherwise). Sections of the referenced document: "Composability Guideline of WS-Reliability and WS-Security" will be included in this

Case Studies document, as appropriate. The former document assumes that the WS-R entity has intimate knowledge of the WS-Security entity and is implemented in the same SOAP HDR processing node. Other models, such as pipelined SOAP HDR processing - in different nodes that are linearly chained together- may also be considered as a separate case of composability.

Also note that similar work is ongoing in the OASIS **eBXML** TC. Specifically, to make eBXML (version 3) messages delivered reliably and securely (with signatures and encryption, if necessary).

III. Issues in the Composition of WS-Reliability and WS-Security

Note that the composability document only illustrates selected ways to use the two specifications together. There are many other possible ways to use the two specifications together. In particular, a point to point model (with no intermediaries) would allow for encryption of the WS-Reliability headers.

There is a special dependency between WS-Reliability and WS-Security that involves the security of the WS-Reliability headers within WS-Security messages.

It is strongly recommended that the integrity of the WS-Reliability headers be guaranteed, as this is a pre-requisite for a Reliable Message Processing (RMP) entity to be able to enforce the reliability contracts. This generally requires that some digest of reliability headers be created and inserted in WS-Security headers. Confidentiality may be achieved in the same way, although less critical than integrity of headers.

We define a "supporting" specification as a specification that may compose with WS-Reliability to better support some of its features. Besides WS-Security, which is essential in guaranteeing the integrity of reliability headers, **we see two areas where an RMP implementation can leverage supporting specifications and the available implementations of those:**

a] **Addressing**: the representation of the address which specifies where a Call Back response should be sent to.

Because such specifications as WS-Addressing or WS-Message Delivery are not finalized yet (both have been submitted to W3C), WS-Reliability keeps open the representation of new reply address formats, via extensibility points in the XML schema.

b] **Agreement or contract between WS endpoints**: the representation of a Reliable Message (RM) agreement. Because no open specification is available yet to support this (WS Agreement from the GGF may come close), an abstract representation of the RM agreement is provided in WS-Reliability. This representation can be related to any future

specification that expresses RM agreements via an appropriate binding.

Here are a few other points to consider when composing WS-R with WS-Security (or other WS standards/ specifications:

-SOAP HDR processing entities may be independent of each other and might be implemented in different nodes (or optionally, in the same node). That is, both integrated WS-R and WS-S processing as well as pipelining of the SOAP HDR processing entities are permitted. However, the modularity of SOAP HDR processing entities should always be assumed, even if extra HDR processing is the result.

-Routing of WS-R messages should not affect composability and the routing should be independent of the SOAP message contents. That is, composability should not be constrained by any routing scheme.

-There are various ways to order the WS-R and WS-Security (WS-S) entities. For example, there may be cases where we have: WS-R then WS-S, WS-S then WS-R, WS-S then WS-R then another WS-S (say for encryption or authentication purposes). Duplicate instances of WS-S processing – before and after the WS-R entity- are permitted. In cases where there is no WS-S header, the WS-R entity will bypass the WS-S entity and pass on the received SOAP message body to the WS application entity.

-Encryption of only the SOAP message body, or the entire SOAP envelope (headers along with message body) should be considered as separate cases. See IV. below.

-Redundant processing of duplicate SOAP messages is permitted. In some cases, redundant processing is a function of how the entities are ordered and that they are independent of each other.

IV. Encryption of the WS-Reliability Headers

As previously noted, the referenced composability document, does not address how to encrypt the WS-Reliability headers, as that might preclude the use of intermediaries, e.g. a router, which wouldn't necessarily posses the encryption key(s) for multiple pair-wise connections. Another example of an intermediary could be a "cipher engine" that receives unencrypted outbound reliable messages at the edge of the enterprise network and encrypts/ decrypts them for external transmission/ reception over the public Internet or Intranet.

We suggest consideration of a point to point model (with no SOAP-aware intermediaries) where the headers would be encrypted by Reliable Message Processing (RMP) entities. Thus, header integrity would be preserved, even if there was a "man in the middle" that attempted to corrupt or pick off the header information.

References:

1. **WS Reliability Now an OASIS Standard**, Alan J Weissberger

http://www.webtorials.com/main/newsletters/dcti/WS-Reliability.pdf

http://www.gridtoday.com/04/1115/104252.html

2. **WS-I November Meeting Report- Section III.,** Alan J Weissberger

http://www.webtorials.com/main/newsletters/dcti/WS-I-Nov04.pdf