Implementation and Performance of WS-Security

Satoshi Makino, Kent Tamura, Takeshi Imamura, and Yuichi Nakamura
IBM Tokyo Research Laboratory, Japan

ABSTRACT
This article describes performance improvements for the Web Services Security (WS-Security) specification. In the course of its development and performance measurement, we identified bottlenecks in the XML parsing and public key operations such as RSA signature and encryption. We are working on minimizing the impact of both of these bottlenecks. We implemented a stream-based WS-Security processor and showed its efficiency in XML parsing, in terms of both the processing time and the memory usage. In addition, we introduced the Web Services Secure Conversation (WS-SecureConversation) to avoid expensive public key operations.

Keywords: Web services; security, WS-Security; performance

INTRODUCTION
Web services often employ intermediary nodes between clients and service providers. In order for the exchanged messages to be protected against altering or eavesdropping even by these intermediaries, it is insufficient to protect only the connections between each node. The message path should be protected all the way through, beginning from the client to the service provider. To do this, we cannot simply rely on the security mechanisms that are provided in the transport layer,
and mechanisms need to be added to the SOAP messages themselves. Message-level security makes it possible for specific parts of the SOAP messages (e.g., credit card numbers) to be digitally signed or encrypted and provides users with fine-grained security control for the messages.

This article describes our implementation of the WS-Security specification that was written to protect SOAP messages. We also point out some performance bottlenecks and propose some possible solutions for the bottlenecks.

**WS-SECURITY SPECIFICATION**

The Web Services Security (WS-Security) specification (Web Services Security, 2002) is intended to realize message-level security for the exchange of SOAP messages and is undergoing standardization by the OASIS Web Services Security Technical Committee. This section outlines the specification.

**Overview**

The WS-Security specification defines mechanisms such as digital signatures and encryption so that the SOAP messages will be protected. This specification also defines the syntax for encoding the arbitrary format of the security tokens that are to be embedded into the messages.

**Security Tokens**

A security token represents the abstract concepts such as name, key and privileges owned by the message sender.

The WS-Security specification defines a token containing the username and password (a UsernameToken) and one containing binary formatted objects such as X.509 certificates and Kerberos tickets (a BinarySecurityToken).

**Digital Signature and Encryption**

The syntax for digitally signing and encrypting SOAP messages is based on the W3C Recommendations for XML signature (XML Signature, 2002) and XML encryption (XML Encryption, 2002), respectively. The syntax of digital signatures and encryption according to the WS-Security specification is shown in Figure 1.

In Figure 1, the `<wsse:BinarySecurityToken>` element contains an X.509 certificate and it is referred to by the `<ds:Signature>` element for the signature verification.

**Other Functions**

The WS-Security specification also defines the format of the message timestamps, the method to send the passwords securely with nonces and timestamps, and so on.

**IMPLEMENTATION**

We implemented some of the functions defined in the WS-Security Specification, such as the digital signature, encryption, UsernameToken, and so on. This implementation is bundled within IBM WebSphere Application Server, V5.0 (WAS V5; http://www.ibm.com/software/
The Assurance Point Model for Consistency and Recovery in Service Composition
www.igi-global.com/chapter/assurance-point-model-consistency-recovery/59927?camid=4v1a

F-DRARE: A Framework for Deterministic Runtime Adaptation of Cyber Physical Systems
www.igi-global.com/chapter/drare-framework-deterministic-runtime-adaptation/69478?camid=4v1a