

# Web Services Security: What's Required To Secure A Service-Oriented Architecture

*An Oracle White Paper  
October 2006*

# Web Services Security: What's Required To Secure A Service-Oriented Architecture.

## INTRODUCTION

The service-oriented architecture (SOA) concept is now embraced by many companies worldwide. However, because of its nature (loosely-coupled connections) and its use of open access (HTTP), SOA adds a new set of requirements to the security landscape.

Many companies rely on the Secure Socket Layer (SSL) protocol to protect access to SOA deployments. SSL provides authentication, confidentiality and message integrity. However, when the data is not "in transit," the data is not protected, which makes the environment vulnerable to attacks in multi-step transactions. As a result, there is a need to address more specific SOA security challenges by relying on additional, application-level industry standards:

Because they are predicated on industry standards, Oracle Fusion Middleware components can be "hot-pluggable" with other standards-compliant vendor platforms.

- Content Security: XML Encryption, XML Signature.
- Message-Level Security: WS-Security.
- Secure Message Delivery: WS-Addressing, WS-ReliableMessaging, WS-ReliableMessaging Policy Assertion.
- Metadata: WS-Policy, WS-PolicyAssertions, WS-PolicyAttachment, WS-SecurityPolicy.
- Trust Management: SAML, WS-Trust, WS-SecureConversation, WS-Federation.
- Public Key Infrastructure: PKCS, PKIX, XKMS

Some of these standards have been around for several years, originally designed for web applications and later leveraged by SOA, for example SSL (mentioned above), and Kerberos, a cross-platform authentication and single sign-on system. Other standards have specifically been created to provide security to networks of web services, for example WS-Security and WS-Policy, described in more detail in the following section.

Because they are predicated on industry standards, Oracle Fusion Middleware components can be "hot-pluggable" with other standards-compliant vendor platforms.

## EVOLVING REQUIREMENTS

Until recently, the onus was on the developer to tackle web services security. Developers used to code security into web services thus creating environment “silos” difficult to manage and costly to maintain.

SOA deployments have become more and more complex, creating additional challenges that developers alone cannot meet anymore, such as cryptographic data protection, identity management, and governance.

Vendors such as Oracle have created solutions to help enterprises meet these new challenges. However, companies do have legacy environments and existing infrastructures that must accommodate the SOA paradigm, and a single vendor may not be able to provide every single piece of the SOA security, management, and governance puzzle. As a result, vendors must rely on industry standards to make their offerings interoperable.

Oracle implements key industry standards in its products to make them “hot pluggable”, and in many cases, Oracle is a driving force behind new standards designed to meet ever-growing SOA security and management challenges.

## STANDARDS

The main goal of SOA security standards is to provide a basis for interoperability among multiple products used in heterogeneous customer environments. Standards-based implementation strategies accelerate development, simplify integration, and reduce administrative costs over time.

Most SOA industry standards are defined in XML frameworks commonly referred to as WS-\* specifications.

Most SOA industry standards are defined in XML frameworks. The last few years have seen the emergence of a plethora of XML-based specifications addressing various aspects of SOA security. Most of these specifications are part of the so-called WS-\* (Web Services specifications) stack.

Most WS-\* specifications started as proprietary industry initiatives. Some of these specifications (e.g., WS-Security, WS-Trust, WS-Policy) have been transferred over to standards bodies such as the Organization for the Advancement of Structured Information Standards (OASIS) or the World Wide Web Consortium (W3C). WS-\* specifications often depend on each other, for example, WS-Policy uses WS-PolicyAssertions. WS-\* specifications also leverage non-WS specifications, for example, WS-Security uses XML Encryption and XML Signature.

This section describes the standards that are key to providing secure and manageable SOA environments.

### *Web Services Security (WS-Security)*

WS-Security specifies SOAP security extensions that provide confidentiality using XML Encryption and data integrity using XML Signature. WS-Security also includes profiles that specify how to insert different types of binary and XML security tokens in WS-Security headers for authentication and authorization purposes:

- Username with optional Password digest (defines how a web service consumer can supply a username as a credential for authentication; the username can be accompanied by a hashed password).
- X.509 Certificate (a signed data structure designed to send a public key to a receiving party).
- Kerberos ticket (an authentication and session token).
- Security Assertion Markup Language (SAML) assertion (see more detail on SAML later in this document).
- REL document (rights expression language (REL) license tokens inserted in WS-Security headers are used for authorization).
- XCBF document (defines how to use the XML Common Biometric Format (XCBF) language for authentication with the WS-Security specification).

### *Web Services Addressing (WS-Addressing)*

WS-Addressing provides an XML framework for identifying web services endpoints and for securing end-to-end endpoint identification in messages. A web service endpoint is a resource (such as an application or a processor) to which web services messages are sent. Oracle Web Services Manager, for example, uses WS-Addressing for metric correlation.

### *Web Services Reliable Messaging Protocol (WS-ReliableMessaging or WS-RM).*

WS-RM is often confused with WS-Reliability. WS-RM and WS-Reliability have a very similar purpose (i.e., the guarantee that a message that has been sent will actually be delivered) and they may be merged in the future, however, today WS-RM seems to hold sway.

WS-RM defines a framework for identifying and managing the reliable delivery of messages between web services endpoints. WS-RM is predicated on the SOAP messaging structure (SOAP binding) and relies on WS-Security, WS-Policy, and WS-Addressing to provide reliable messaging.

WS-RM defines a reliable messaging (RM) source (the party that sends the message) and an RM destination (the party that receives the message). WS-RM mandates prerequisites. For example, trust between endpoints must be established, and the message and endpoints must be formally identified (this is achieved through the use of the complementary WS-\* specifications mentioned in the previous paragraph).

### *WS-ReliableMessaging Policy Assertion (WS-RM Policy)*

WS-RM Policy defines a policy assertion that leverages the WS-Policy framework in order to enable an RM destination and an RM source to describe their requirements for a given sequence.

### *Web Services Policy (WS-Policy)*

A web service provider may define conditions (or policies) under which a service is to be provided. The WS-Policy framework enables one to specify policy information that can be processed by web services applications, such as Oracle Web Services Manager. A policy is expressed as one or more policy assertions representing a web service's capabilities or requirements. For example, a policy assertion may stipulate that a request to a web service be encrypted. Likewise, a policy assertion can define the maximum message size that a web service can accept. WS-Policy expressions are associated with various web services components using the WS-PolicyAttachment specification.

### *Web Services Security Policy (WS-SecurityPolicy)*

WS-SecurityPolicy is part of the Web Services Secure Exchange (WS-SX) set of specifications hosted by OASIS (in addition to WS-SecurityPolicy, the WS-SX technical committee defines two other sets of specifications: WS-Trust and WS-SecureConversation, described later in this document).

WS-SecurityPolicy defines a set of security policy assertions used in the context of the WS-Policy framework. WS-SecurityPolicy assertions describe how messages are secured on a communication path.

Oracle has submitted to the OASIS WS-SX technical committee several practical security scenarios (that will be implemented in a future version of Oracle Web Services Manager). Each security scenario describes WS-SecurityPolicy policy expressions.

### *Security Assertion Markup Language (SAML)*

SAML is an open framework for sharing security information on the Internet through XML documents. SAML was originally designed to address the following:

- Limitations of web browser cookies: SAML provides a standard way to transfer cookies across multiple Internet domains.
- Proprietary web single sign-on (SSO): SAML provides a standard protocol to implement SSO within a single domain or across multiple domains.
- Federation: SAML enables identity management (a user can have several identities on the Internet).
- Web services security: SAML provides a standard security token (a SAML assertion) that can be used with the WS-Security framework.

SAML is key to Oracle's strategy in terms of web services security, federation, and identity propagation. Oracle is a very active participant in the OASIS technical committee that hosts SAML (in fact, the original co-authors of the SAML specification are current Oracle employees).

SSL, SAML, WS-Security are proven, widely used standards supported by all market-leading SOA security vendors.

### *Web Services Trust (WS-Trust)*

In a message exchange using WS-Security only, it is assumed that both parties involved in the exchange have a prior agreement on which type of security tokens they must use for sharing security information. However, there are cases where these parties don't have such an agreement, as a result trust must be established before exchanging messages. Trust between two parties exchanging SOAP / WS-Security-based messages is established by implementing the WS-Trust specification.

Essentially, WS-Trust enables security token interoperability using a security token service (STS) that allows one party to request a security token from a trusted authority. For example, Company A using simple username / password credentials may request a SAML assertion from a trusted authority to do business with Company B, which is expecting SAML assertions as security credentials.

Because of its importance in the federation world, WS-Trust (together with ancillary standards such as WS-SecureConversation) is key to Oracle's identity federation strategy. Oracle Access Manager, Oracle Identity Federation, and Oracle Web Services Manager will fully support WS-Trust in a forthcoming release.

### *Web Services Secure Conversation (WS-SecureConversation)*

WS-SecureConversation leverages WS-Security and WS-Trust. WS-SecureConversation defines the creation and sharing of security contexts between communicating parties. Security contexts mitigate the overhead involved in multiple-message exchanges. WS-SecureConversation defines a Security Context Token (SCT) element to support the requirements of security contexts. An SCT involves a shared secret used to sign and/or encrypt messages.

### *Web Services Federation (WS-Federation)*

WS-Federation provides support for the secure propagation (across Internet domains) of identity, attribute, authentication, and authorization information. By relying on the models defined in WS-Security and WS-Trust, WS-Federation enables brokering of trust and security token exchange, support for privacy by hiding identity and attribute information, and federated sign-out. WS-Federation leverages WS-Policy and WS-MetadataExchange to describe and acquire metadata.

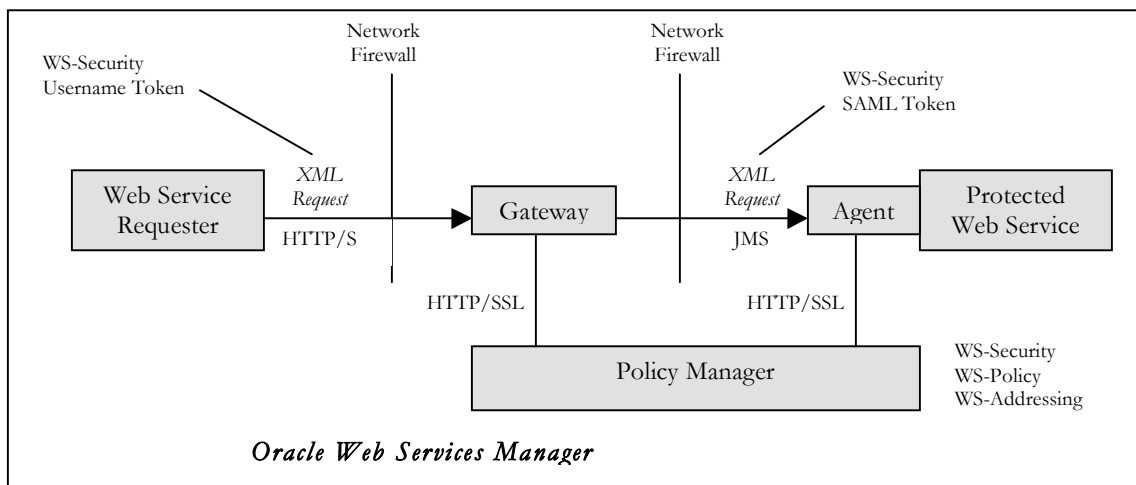
Oracle Identity Federation currently supports the WS-Federation Passive Requester Profile for browser-based federation.

Finally, this section would not be complete without mentioning essential public key infrastructure (PKI) standards contributing to SOA security.

- PKCS (Public Key Cryptographic Standards): A set of specifications developed and maintained by RSA Security (now part of EMC). There are currently 12 PKCS specifications. The most common are PKCS#1 (RSA Cryptography Standard), PKCS#5 (Password-Based Cryptography Standard), PKCS#7 (Cryptographic Message Syntax Standard), PKCS#8 (Private-Key Information Syntax Standard), PKCS#10 (Certification

Request Syntax Standard), PKCS#11 (Cryptographic Token Interface Standard), and PKCS#12 (Personal Information Exchange Syntax Standard). Oracle supports PKCS#1, PKCS#7, PKCS#8, and PKCS#11 in Oracle Security Developer Tools (OSDT). OSDT provides libraries used across all the Oracle Fusion Middleware products.

- PKIX (Public-Key Infrastructure - X.509): An Internet Engineering Task Force (IETF) working group. The goal of the PKIX work group is to develop Internet standards to facilitate the use of X.509-based PKI environments. These standards cover areas involving X.509 certificates, Certificate Revocation List (CRL), Lightweight Directory Access Protocol (LDAP), Certificate Management Protocol (CMP), Online Certificate Status Protocol (OCSP), Time-Stamp Protocol (TSP), and Cryptographic Message Syntax (CMS). PKIX standards are supported by the Oracle Security Developer Tools (OSDT).
- XKMS (XML Key Management Specification): Defines protocols for distributing and registering public keys. Applications or web services supporting XKMS don't have to deploy a PKI solution locally. The application or web service sends XML requests (in SOAP messages) to PKI components installed at a trusted third-party site (e.g., Verisign) which executes the XML requests on behalf of the requesting party (typically, XML requests are for issuing, retrieving, or revoking a certificate). XKMS works in conjunction with XML Encryption and XML Signature.



## ORACLE'S PRODUCT STRATEGY

Oracle takes a holistic approach to protecting SOA deployments, including an identity management infrastructure (Oracle Access Manager, Oracle Identity Federation, Oracle Web Services Manager), development and deployment tools (Oracle Security Developer Tools (OSDT), JDeveloper (Java Integrated Development Environment), Application Development Framework (ADF), and a

secure, governance-aware runtime environment (Oracle Components for Java – OC4J, including a UDDI registry).

As part of Oracle Fusion Middleware, Oracle provides an encompassing SOA security and management solution that allows companies to externalize security outside applications and web services, combine transport-level and application-level protection, and use a layered defense system (i.e., security gateways reside in the DMZ to ensure “perimeter” security, and local interceptors (or “agents”) provide “last-mile” security behind the inner network firewall).

All the key industry standards described previously are supported by the various components of the Oracle solution.

## **CONCLUSION**

The increasing popularity of service-oriented architectures has introduced the need for additional standards to help support the new security challenges involved with this new paradigm.

Oracle has been instrumental in contributing to emerging standards, in particular the specifications hosted by the OASIS Web Services Secure Exchange technical committee.

In terms of identity management and service-oriented architectures security, Oracle's strategy is to focus on well established standards such as SAML, essential for identity federation, identity propagation, and end-to-end security from the user's web browser all the way across SOA-based transactions involving multiple web services.

Oracle Fusion Middleware components provide full support for key SOA security standards, enabling interoperability with heterogeneous, multiple-vendor security infrastructures.



Web Services Security  
October 2006  
Author: Marc Chanliau

Oracle Corporation  
World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

Worldwide Inquiries:  
Phone: +1.650.506.7000  
Fax: +1.650.506.7200  
[oracle.com](http://oracle.com)

Copyright © 2006, Oracle. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice.

This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.