

Securing Web Services — Concepts, Standards, and Requirements

White Paper
October 2003



Table of Contents

Executive Summary1
Introduction2
Business Drivers for Securing Web Services4
Financial4
Legislative Compliance4
Trust and Privacy4
Security4
Technology5
Basic Concepts6
Security Vocabulary6
Threat Profile and Risk Analysis8
Security Challenges Specific to Web Services9
Defense Against Threats9
Web Services Security Specification Development and Standardization11
Why Standards?13
Specification Development, Standards, and Sun14
Requirements for Securing Web Services15
Conclusion21
Next Steps22
References23
Web Sites23
About the Author24

Chapter 1

Executive Summary

There is strong momentum for enterprises to bring Web services technologies into the mainstream of their Service Oriented Architectures (SOAs). Many executives and industry organizations see Web services as the next wave of fueling electronic business, application integration, and business-to-business (B2B) interactions. Studies have already shown that implementing a Web services architecture can reduce complexity and provide significant cost savings. With a potential for such cost reductions and standards becoming mature, Web services products and technologies are being assembled to create solutions and enable new business models.

With Web services, applications expose their internal workflows, business processes, and architectures. This calls for securing them against a wide range of attacks, both internal and external. This paper is an effort to understand the basic concepts for securing Web services, the standards associated to solve this puzzle, and the requirements for implementing secure Web services.

It is commonly said, "Security is as strong as the weakest link." Security in Web services cannot be bolted into an existing architecture. The architecture design must be modular, so that specific security technologies can be plugged in at appropriate parts of the infrastructure. The set of requirements provided in this paper helps in incorporating security throughout. The second white paper planned in this series, *Securing Web Services - Architect, Deploy, and Manage*, will provide more details on architecting, implementing, and deploying secure Web services.

Chapter 2

Introduction

Web services represents the next generation of distributed computing, building on and extending the current client-server model in some important ways. Web services adhere to a concept known as “loose coupling,” which means services are discoverable, platform independent, and are expressed with self-describing interfaces. Web services primarily rely upon the Extensible Markup Language (XML) as the means to express interfaces.

Properly developed, loosely coupled services can be accessible as discreet components of business logic, executed as standalone services, or combined with other services to create composite applications. Composing services in such a manner suggests that the various components may execute in a context that the component creator never envisioned.

This notion of loose coupling serves as the underpinning of a Service Oriented Architecture (SOA). It precipitates a migration from the current approach of tightly coupled and Application Programming Interface (API)-centric application stacks, where the execution environment is static, rigidly defined, and well understood, to a more fluid environment that will undoubtedly have an impact on how we think about security.

The flexibility that Web services provides to developers also makes it an attractive deployment platform for wireless applications such as gaming, text-based messaging, and mobile Internet access. Wireless applications and services have volume potential several orders of magnitude greater than anything deployed today. In addition to the obvious issue of providing a scalable quality of service (QoS), these wireless services will require new levels of capability related to authentication, access control, provisioning, and identity management.

Traditionally, companies have used application servers as central places to deploy their Web applications and business services. Such application servers become the access or integration points for all Web services. Application servers also play a major role in establishing trust context between the disparate Web services that need to integrate.

With Web services using the insecure Internet for mission-critical transactions with the possibility of dynamic, short-term relationships, security is a major concern. To secure Web services, a range of XML-based security standards are being developed to solve problems related to authentication, role-based access control (RBAC), messaging, and data security.

Chapter 3

Business Drivers for Securing Web Services

The importance of securing Web services is underscored by the following key business drivers and their impact on the enterprise.

Financial

- The need to contain and control costs while expanding channels of business, regardless of location of end users (for example, customers, suppliers, partners, and employees).
- Impact on profit and loss if there is a breach in security.

Legislative Compliance

- Mitigation of liability arising from regulations that protect consumer privacy.
- Governing the sharing of personal information without consumer consent.

Trust and Privacy

- Acceleration of data access and sharing creates more opportunity to infringe on personal data privacy, resulting in actual or perceived loss of trust in merchants.

Security

- Proliferation of Internet-based solutions has multiplied the number of access points to confidential information. Without an appropriate security policy and superior security controls, the possibilities for data compromise are greatly increased.

Technology

- The need for more flexible, standardized, and context-based forms of managing identity that are device- and application-independent.
- Implementations must now support a wide range of information technologies and devices with mission-critical levels of scalability and reliability.

Chapter 4

Basic Concepts

Security Vocabulary

- **Entity** – An active element of a computer or network system.
- **Relying Party** – A system entity that makes a decision contingent upon information or advice from another system entity.
- **Identity** – The electronic representation of a real-world entity (human, organization, application, service, or network device).
- **Identity Management** – Describes the technology infrastructure and business processes for managing the life cycle and usage of an identity, including attributes, rights, and entitlements.
- **Key** – A value (random number) used by a cryptographic algorithm to alter (encrypt or decrypt) information.
- **Key Management** – The process by which a key is generated, stored, protected, transferred, loaded, used, and destroyed. This is the most difficult part of cryptography, as keeping keys secret is not easy. It is also the most important part, because one mistake with a private key and the entire infrastructure may be compromised, rendered insecure and useless.
- **Trust** – The characteristic that makes one entity willing to rely upon a second entity to execute a set of actions and make a set of assertions (usually dealing with identity) about a set of subjects or scopes. Trust depends on the ability to bind unique attributes or credentials to a unique entity or user.
- **Trust Domain** – A security space where the target of a request can determine whether a particular set of credentials from a source satisfies the relevant security policies of the target. The target may defer trust to a third party, thus including the third party in the trust domain.

- **Trust Model** – The process performed by the security architect to define a complementary threat profile and a model based on a use case-driven data flow analysis. The result of this exercise integrates information about the threats, vulnerabilities, and risks of a particular information technology architecture. Further, trust modeling identifies the specific mechanisms that are necessary to respond to a specific threat profile. A security architecture based on an acceptable trust model provides a framework for delivering security mechanisms. A trust model provides an assurance that the trust binding is reliable to the level of satisfaction required by the relying party, as specified by security policy.

It is also important to note what a trust model is not. It is not the particular security mechanisms utilized within a particular security architecture. Rather, it is the enforcement of security mechanisms in conjunction with the security policy, such that they address all business, technical, legal, regulatory, or fiduciary requirements to the satisfaction of a relying party.

- **Establishing Trust** – To establish trust or confidence, there must be a binding of unique attributes to a unique identity, and the binding must be able to be corroborated satisfactorily by a relying party. When a satisfactory level of confidence in the attributes is provided by an entity, a trust relationship is established. This element of trust is commonly called authentication.
- **Public Key Infrastructure (PKI)** – Relies upon public key cryptography, also known as asymmetric key cryptography. It uses a secret private key that is kept from unauthorized users and a public key that is handed to trusted partners. Both keys are mathematically linked. Data encrypted by the public key can be unencrypted only by the private key, and data signed by the private key can be verified only by the public key.

A PKI is a foundation upon which other applications and network security components are built. The specific security functions for which a PKI can provide a foundation are confidentiality, integrity, non repudiation, and authentication. The primary function of a PKI is to allow the distribution and use of public keys and certificates with security and integrity.

It should be noted that a PKI is not by itself an authentication, authorization, auditing, privacy, or integrity mechanism. Rather, it is an enabling infrastructure that supports these various business and technical needs. In particular, a PKI allows only for the identification of entities.

- **Confidentiality** – Ensures that the secrecy and privacy of data is provided with cryptographic encryption mechanisms. Customer personal information and legal or contractual data are prime examples of data that should be kept secret with confidentiality mechanisms. Encryption of data is possible by using either public (asymmetric), or secret (symmetric) cryptography. Since public key cryptography is not as efficient as secret key cryptography for data encipherment, it is normally used to encipher relatively small data objects such as secret keys used by symmetric-based encryption systems. Symmetric cryptographic systems are often incorporated into PKIs for bulk data encryption; thus, they are normally the actual mechanism used to provide confidentiality.
- **Integrity** – Ensures that data cannot be corrupted or modified, and transactions cannot be altered. Public key certificates and digital signature envelopes are good examples of information that must have an assurance of integrity. Integrity can be provided by the use of either public (asymmetric), or secret (symmetric) cryptography. An example of secret key cryptography used for integrity is the Data Encryption Standard (DES) in Cipher Block Chaining mode where a Message Authentication Code (MAC) is generated. Note that in the PKI environment, utilizing symmetric cryptographic systems for implementing integrity does not scale particularly well. Public key cryptography is typically used in conjunction with a hashing algorithm, such as Secure Hash Algorithm 1 (SHA-1) or Message Digest 5 (MD5), to provide integrity.

- **Authentication** – Verifies that the identity of entities is provided by the use of public key certificates and digital signature envelopes. Authentication in the Web services environment is performed very well by public key cryptographic systems incorporated into PKIs. In fact, the primary goal of authentication in a PKI is to support the remote and unambiguous authentication between entities unknown to each other, using public key certificates and trust hierarchies. Authentication in a PKI environment relies on the mathematical relationship between public and private keys. Messages signed by one entity can be tested by any relying entity. The relying entity can be confident that only the owner of the private key originated the message, because only the owner has access to the private key.
It should be noted that the most common form of authentication is by employing a username and password. For many Web services, this is sufficient. But, when a PKI is used, the level of assurity is the greatest.
- **Non repudiation** – Ensures that data cannot be renounced, or a transaction denied. This is provided through public key cryptography by digital signing. Non repudiation is a critical security service of any application where value exchange and legal or contractual obligations are negotiated. Non repudiation is a by-product of using public key cryptography. When data is cryptographically signed using the private key of a key pair, anyone who has access to the public key of that pair can determine that only the owner of the key pair itself could have signed the data in question. For this reason, it is paramount that end entities secure and protect the private keys that they use for digitally signing data.
- **Authorization** – Verifies that the identity has the necessary permissions to obtain the requested resource or act on something before providing access to it. Normally, authorization is preceeded by authentication. As an example, for a given system, a system administrator defines which users are allowed access and their privileges (such as access to which file directories, hours of access, amount of allocated storage space, and so forth).

Threat Profile and Risk Analysis

Threat profiles and risk analysis are intrinsically related. One without the other is of limited value. Threat profiles identify the specific threats that are most likely to put the environment at risk. The most common types of threats fall into categories such as:

- Actual or attempted unauthorized probing of any system or data
- Actual or attempted unauthorized access
- Introduction of viruses or malicious code
- Unauthorized modification, deletion, or disclosure of data
- Denial of service attacks

Looking at the above list, one might initially assume that all threats come from external sources, and that a system not on the Internet is not at risk. However, remember that poorly trained, careless, or malicious employees can represent every one of the threats mentioned.

To build and evaluate your specific threat profile, the recommended tool is a use case-driven data flow analysis. This is a process of methodically tracing the flow of various use cases and their data throughout the system to identify threats and vulnerabilities. It should be noted that threats are dependent on the specifics of a system's implementation, and are different from vulnerabilities, which are intrinsic to a system.

Security Challenges Specific to Web Services

With Web services, more of the application internals are exposed to the outside world. As the application is closer to the data than to the perimeter or the network, it opens room for security threats not only to the host system and application, but also to the entire infrastructure.

Traditionally, Secure Sockets Layer (SSL), Transport Layer Security (TLS), Virtual Private Networks (VPNs), and Internet Protocol Security (IPSec) are some of the common ways of securing content. However, these are point-to-point technologies. They create a secure tunnel through which data can pass. With the Secure Multipurpose Internet Mail Exchange (S/MIME) protocol, data could be sent digitally signed and encrypted over the insecure Internet.

Web services require much more granularity. They want to maintain secure context and control it according to their security policies. Although traditional technologies are commonly used in Web services, they are not sufficient.

The following is a set of challenges specific to Web services:

- Inter-enterprise Web services are dealing with untrusted clients. The Remote Procedure Call (RPC)-style services have special needs. For example, is the caller authorized to ask for this computer action?
- End to end isn't just point to point:
 - The creator of the message wrote the payload, but intermediaries may touch or rewrite the message afterwards.
 - Long-running choreographed conversations with multiple requests, responses, and forks.
- Clients and services do not have a way to negotiate their mutual constraints and capabilities before interacting.
- Securing Web services infrastructure needs XML's granularity:
 - Encrypting or digitally signing select portions
 - Acting on rewritten individual headers
- Standards for securing Web services are heavily PKI oriented.
- Trust management must be more robust for distributed computing to scale.
- Authorization policies are more difficult to write as environments become more loosely coupled.
- Intermediaries, particularly combined with attachments, make full protection more difficult.

Defense Against Threats

Some common threats to any Web application include: Denial of service attack, man in the middle attack, Trojan horse, virus via e-mail, buffer overflow attack, improperly configured client browsers, improperly configured Web and e-mail servers, dictionary attack, brute force attack, smurf (denial of service by flooding the network) attack, replay attack, and domain name server (DNS) attacks.

Depending on the attacker's location and level of access, attacks can be launched from the perimeter, network, host, or even the application itself. One should note that it may be impossible and very expensive to thwart every threat. Therefore, the focus should be on minimizing and spreading the risk.

Table 4-1: Defense Against Threats

Threat	Defense
Perimeter	Firewall, VPN, Intrusion Detection
Network	Access Control List (ACL), Encryption, Intrusion Detection
Host	Host-based Firewall, Patch, Antivirus, Configuration
Application	Configuration, Security Model, Authentication
Data or Resource	ACL, Auditing, Digital Rights Management

Web services are present at the application layer. This means it is important that perimeter, network, and host are well secured to reduce threats. In short, more security is needed in addition to existing security mechanisms.

Chapter 5

Web Services Security Specification Development and Standardization

Many standards bodies — such as the World Wide Web Consortium (W3C), Organization for the Advancement of Structured Information Standards (OASIS), the Liberty Alliance, and others — are developing horizontal and vertical Web services infrastructure standards and specifications to allow enterprises to overcome challenges associated with traditional security technologies. Some prominent security standards and specifications that are of interest include:

- **XML Signature** – One of the most prevalent XML-based security standards. This provides an XML-compliant syntax for representing the signature of Web resources and portions of protocol messages (anything that can be referenced by a Universal Resource Identifier or URI) and procedures for computing and verifying such signatures.
- **XML Encryption** – This specifies a process for encrypting data and representing the result in XML. The data may be arbitrary data (including an XML document), XML element, or XML element content. The result of encrypting data is an XML Encryption element that contains or references the cipher data. The standard also specifies an XML Signature decryption transform that enables XML Signature applications to distinguish between those XML Encryption structures that were encrypted before signing (and must not be decrypted), and those that were encrypted after signing (and must be decrypted) for the signature to validate.

- **Simple Object Access Protocol (SOAP)** – As of version 1.2, SOAP is also referred to as XML Protocol. SOAP is a lightweight, XML-based messaging protocol framework for building and exchanging distributed, structured information in a decentralized and distributed environment.
- **SOAP Message Security (Also known as Web Services Security, or WSS)** – Supports security mechanisms of several types, each using implementation and language-neutral XML formats defined by XML schema. The security mechanisms include use of XML signature to provide SOAP message integrity; use of XML encryption to provide SOAP message confidentiality; attaching and/or referencing security tokens in headers of SOAP messages, carrying security information for potentially multiple, designated actors; and associating signatures with security tokens.
- **XML Key Management Specification (XKMS)** – An XML protocol that allows a simple client to obtain key information (value, certificate, management, or trust data) from a Web service. It also describes protocols for distributing and registering public keys, suitable for use in conjunction with the standards for XML Signature and XML Encryption. XKMS helps overcome PKI complexity by allowing Web services to become clients of a key management service.
- **Extensible Access Control Markup Language (XACML)** – Describes both an access control policy language and a request/response language. The policy language is used to express access control policies (who can do what, and when). The request/response language expresses queries about whether a particular access should be allowed (request) and describes answers to those queries (responses).
Lately, a new specification called the Web Services Policy Language (WSPL) is being developed as a generic language to express policy information. This is based on the XACML work.
- **Extensible Rights Markup Language (XrML)** – XrML provides a universal method for securely specifying and managing rights and conditions associated with all kinds of resources, including digital content and services.
- **Web Services Description Language (WSDL)** – An XML language for describing Web services; it defines the core language that can be used to describe Web services, based on an abstract model of what the services offer. Technically, WSDL describes network services as a set of end points operating on messages containing either document-oriented or procedure-oriented information. It also describes the sequence, direction, and cardinality of abstract messages sent or received by an operation.
- **Security Assertion Markup Language (SAML)** – SAML defines a protocol by which clients can request assertions from SAML authorities and receive responses from them (exchange of security information). This protocol, consisting of XML-based request/response message formats, can be bound to many different underlying communications and transport protocols. The security information is expressed in the form of assertions about subjects. A subject is an entity that has an identity in some security domain. Assertions can convey information about authentication acts performed by subjects, attributes of subjects, and authorization decisions about whether subjects are allowed to access certain resources.
- **Liberty Alliance** – A consortium of commercial and noncommercial organizations created to support the development, deployment, and evolution of an open, interoperable standard for federated network identity. The vision of the Liberty Alliance is to enable a networked world in which individuals and businesses can more easily conduct transactions, while protecting the privacy and security of vital identity information. The specifications created by this alliance support and include other open industry standards such as SAML, SOAP, Wireless Application Protocol (WAP), Web Services Security (WS-Security), and XML. Also, some of the components

of the published specification have been presented to the SAML working group to be incorporated as extensions to SAML.

- **Digital Signature Standard (DSS)** – This is an upcoming specification. The goal is to support processing of digital signatures as Web services, define a protocol for a centralized digital signature verification Web service that can verify signatures in relation to a given policy set, and define a protocol to produce cryptographic time stamps that can be used for determining whether a signature was created within the associated key's validity period or before revocation.
- **Electronic Business XML (ebXML)** – An initiative between OASIS and the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT). ebXML provides a technical framework that will enable XML to be utilized in a consistent manner for the exchange of all electronic business data. The ebXML Messaging Service (ebMS) is an extension of ebXML that was created to address the implicit security requirements associated with transferring data via the Web.

There are many other specifications that have been proposed for securing Web services. These include WS-Trust, WS-Federation, WS-SecurityPolicy, and WS-SecureConversation. These are not listed above, as they have not been contributed to any standards body for formal standardization.

Why Standards?

Open standards have been the focus of Sun since its inception. After all, the “N” in Sun stands for network, which relies upon known, interwoven, and reliable connections. Open standards are essential to this network concept.

Standards are both initiators and guardians of technical innovation, and their value cannot be underestimated. Standardization enables agreement and interoperability at a particular technical level, beyond or upon which proprietary and differentiating ideas can be added. For vendors, standardization enables more innovation: If the industry comes to agreement upon a particular solution, efforts can then be turned to improving that solution and looking to the next area of innovation. This ultimately benefits consumers. Consumers also win through an ability to choose between competing but interoperable implementations of these standards.

Sun has a very specific definition of open standards. By open, we mean specifications that are reliable, free from the threat of legal encumbrances, able to work across development and deployment environments, and subject to peer review and input throughout most of their life cycles, as well as aligned with general industry and customer needs. By standard, we mean a specification that is developed in recognized, standards-setting organizations.

For the developer, the promise of open standards is dependability, predictability, usefulness, and a solid, long-term technology investment. Developers must know that the standards they rely upon to architect or build a business solution will follow a life cycle that can be viewed and is open to wide participation. Developers need to plan for updates, and want to know what the future of that standard will be, and what it might entail for their work.

Sun's definition of open standards does not include specifications developed by one, two, or even three companies, published on their Web sites, and left there for months or years, with no legal terms for their use — while the companies involved develop products based on the specifications which they proclaim to be standards.

Because of how the Web works, only software which faithfully implements open standards — with minimal or no additional features — can form the foundation for long-term software development.

Specification Development, Standards, and Sun

Sun participates in numerous specification development efforts, from industry organizations to commercial ventures, consortia and myriad standards-developing organizations. The following, a relevant although not exhaustive list, highlights some of the major working groups and technical committees related to security of which Sun is a participant.

Note – Some of the working groups mentioned below have already produced standards and may also be closed.

Table 5-1: Current Sun Participation in Standards Activities Related to Security

Standards Organization	Major Working Groups/Technical Committees
W3C	Web Services Architecture, XML Protocol, XML Digital Signature (in association with IETF), XML Encryption, XKMS
OASIS	ebXML (in association with UN/CEFACT), XML-based Security Services, SAML, XACML, ebXML Implementation - Interoperability and Conformance, Business Transactions, WSS, Web Services Reliable Messaging (WS-RM), PKI
Open Mobile Alliance (OMA)	Mobile Web Services - Data Synchronization, Requirements, Architecture, Interoperability
Internet Engineering Task Force (IETF)	Long-standing and ongoing participation in Applications, General, Internet, Operations and Management, Routing, Security, SubIP, Transport, and User Services
Object Management Group (OMG)	Middleware and Related Services (MRS), Platform Task Force
Java™ Community Process (JCP)	Long-standing and ongoing participation in most of the committees
Java Card Forum	Long-standing and ongoing participation in most of the committees
Global Platform	Card Committee and its subworking groups
European Telecommunications Standards Institute (ETSI)	Smart Card Platform (SCP), Third Generation Partnership Project (3GPP)
Liberty Alliance	Long-standing and ongoing participation in most of the committees
Web Services Interoperability Organization (WS-I)	Basic Security, Sample Applications, and others
Global Grid Forum (GGF)	Open Grid Services Architecture (OGSA), Open Grid Service Infrastructure (OGSI), Web Services Distributed Management (WSDM), OGSA-Security, Common Management Model, Distributed Resource Management (DRM)

Chapter 6

Requirements for Securing Web Services

Web services providers must assure their customers that the integrity, confidentiality, and availability of information they collect, maintain, use, or transmit is protected. The confidentiality of information is threatened not only by the risk of improper access to stored information, but also by the risk of interception during transmission. Table 3 lists a set of requirements that help protect the confidentiality, integrity, and availability of information for both data at rest and in motion.

Federated Network Identity is proving to be extremely important in maintaining privacy and identity control. It offers single sign-on (SSO) functionality to users who demand ease of use and rapid access to resources.

There is also a requirement area for Intellectual Property Rights (IPR). This is becoming very important, with a high volume of open source software development and usage. Also, implementing standards may require licenses from vendors (cost factor). Of late, there are a large number of vendor-driven specifications in the field of Web services. Many of these specifications have illegible or confusing copyrights and licensing terms. Implementations may also be available free of charge for developmental work, but they are not clear on commercial usage.

Table 6-1: Requirements and Implementation Matrix

Requirement Area	Implementation
Administrative Planning and Management	<ol style="list-style-type: none"> 1. Contingency plan: <ul style="list-style-type: none"> • Applications and data criticality analysis • Data backup plan • Disaster recovery plan • Emergency mode operation plan • Testing and revision 2. Chain of trust partner agreement 3. Formal mechanism for processing records 4. Information access control: <ul style="list-style-type: none"> • Access authorization • Access establishment • Access modification 5. Internal audit 6. Security configuration management: <ul style="list-style-type: none"> • Documentation • Hardware/software installation, maintenance review and testing for security features • Inventory • Security Testing • Virus checking 7. Security incident procedures — Request and response 8. Security management process: <ul style="list-style-type: none"> • Risk analysis • Risk management • Sanction policy • Security policy 9. Training: <ul style="list-style-type: none"> • Awareness training for all personnel (including management) • Periodic security reminders • User education concerning virus protection • User education in importance of monitoring login success/failure, and how to report discrepancies • User education in password management

Requirement Area	Implementation
Physical Security	<ol style="list-style-type: none">1. Personnel security:<ul style="list-style-type: none">• Assure supervision of maintenance personnel by authorized, knowledgeable person• Maintain record of access authorizations• Personnel clearance procedure• Personnel security policy/procedure• System users, including maintenance personnel, trained in security2. Termination procedures:<ul style="list-style-type: none">• Combination locks changed• Removal from access lists• Removal of user account(s)• Turn in key, token, or card that allows access3. Media controls:<ul style="list-style-type: none">• Access control• Accountability (tracking mechanism)• Data backup• Data storage• Disposal4. Physical access controls:<ul style="list-style-type: none">• Disaster recovery• Emergency mode operation• Equipment control (into and out of site)• Facility security plan• Procedures for verifying access authorizations prior to physical access• Maintenance records• Need-to-know procedures for personnel access• Sign in for visitors and escort, if appropriate• Testing and revision5. Policy/guideline on work station use6. Secure work station location7. Security awareness training

Requirement Area	Implementation
Network Perimeter Security	<ol style="list-style-type: none"> 1. Access control: <ul style="list-style-type: none"> • Context-based access • Encryption • Procedure for emergency access • Role-based access • User-based access 2. Audit controls 3. Authorization control — Role- and user-based access 4. Data authentication 5. Entity authentication: <ul style="list-style-type: none"> • Automatic logoff • Biometrics • Password • Personal Identification Number (PIN) • Telephone callback • Token • Unique user identification 6. Communication/network control: <ul style="list-style-type: none"> • Access control • Alarm • Audit trail • Encryption • Entity authentication • Event reporting • Integrity control • Message authentication
Application and Data Security	<ol style="list-style-type: none"> 1. Digital signature: <ul style="list-style-type: none"> • Ability to add attribute • Continuity of signature capability • Countersignature • Independent verifiability • Interoperability • Message integrity • Multiple signatures • Non repudiation • Transportability • User authentication
Web Services Creation Tools	<ol style="list-style-type: none"> 1. Rapid prototype 2. Debugging environment 3. Develop custom component

Requirement Area	Implementation
Web Services Execution Environment	<ol style="list-style-type: none"> 1. Application server 2. Web container 3. Servlet container 4. Operating system
Web Services Application Environment	<ol style="list-style-type: none"> 1. Directory service 2. Identity service 3. Portal service 4. Internationalization 5. Database 6. Application registry 7. Mail server 8. Messaging or integration service
Privacy	<ol style="list-style-type: none"> 1. Employee privacy 2. Record privacy 3. Legislation demand <p>These requirements may impair implementation of certain security measures</p>
Performance	<ol style="list-style-type: none"> 1. Hardware acceleration 2. Upgrade existing hardware 3. Tune current applications and operating systems 4. Network-attached encryption 5. Scalability 6. Reliability
IIPR Policy and Licensing Terms of Standards and Specifications	<ol style="list-style-type: none"> 1. Royalty-free (RF) license 2. Reasonable and Nondiscriminatory (RAND) license 3. RAND Zero, RANDz license 4. Berkeley Software Distribution (BSD) or FreeBSD License 5. General Public License (GPL) 6. Lesser General Public License (LGPL) 7. Java Community Process (JCP) license 8. Standard versus specification 9. Open source implementation <p>(See References for choosing a license)</p>

Table 6-2: Mapping Requirements, Technology and Standards

Requirement	Technology	Standard
Authentication	Username/password, key-based digital signing and signature verification, challenge/response, biometrics, smart cards, and more	XML Signature, XKMS, SAML, Java Card, Java 2 Platform, Standard Edition (J2SE™), Kerberos, Generic Security Service (GSS), DSS, Federated Network Identity (Liberty)
Authorization	Application of policy, access control, digital rights management	Java Authentication and Authorization Service (JAAS), XACML, XrML, DSS
Integrity	Message digest itself authenticated with digital signature	J2SE (Secure Hash Algorithm (SHA), MD5, and more), XML Signature, XKMS, XML Encryption
Non repudiation	Key-based signing and signature verification, message reliability	J2SE, Cert Path, XML Signature, XKMS, DSS
Confidentiality	Key-based digital encryption and decryption	J2SE (GSS, Java Cryptography Extension or JCE), XML Encryption, XKMS
Audit	Various forms of logging, themselves secured to avoid tampering	XML Signature, XKMS, J2SE (Logging)
Trust	Key-based signing and signature verification	J2SE (Cert Path, GSS), XML Signature, XKMS, DSS

Chapter 7

Conclusion

Securing Web services is complex and possibly overwhelming. Security must be incorporated into the planned requirements from the very beginning. Addressing a breach in security could be more expensive than implementing security measures in advance (cost of liability, public relations, loss of business, and so on). Also, security should be enforced throughout the infrastructure, both electronically and physically. Standards related to security are being developed by many standards organizations. Some of these standards are mature enough to be incorporated into your Web services applications today. Security for Web services is a necessity and can be deployed.

Chapter 8

Next Steps

The second paper planned in this series, *Securing Web Services - Architect, Deploy, and Manage*, is about architecting, implementing, and managing a Web services infrastructure. It provides an in-depth look into Java platform and Web services architecture, and some off-the-shelf products that could be incorporated to provide for scalability and security. It also provides insight into managing this infrastructure.

Chapter 9

References

Web Sites

Health Insurance Reform: Security Standards – http://www.ihs.gov/generalweb/webcomponents/misc/ihs_disclaimer.cfm?link_out=http://aspe.hhs.gov/admsimp/nprm/seclist.htm

Sun™ BluePrints – sun.com/solutions/blueprints

Choose a License – java.net/choose_license.html

Open Source Licensing – opensource.org/licenses

XML Cover Pages – xml.coverpages.org

Project Liberty – projectliberty.org

Chapter 10

About the Author

Shivaram Mysore is a software architect at Sun Microsystems's Java Software Division. His current focus is in the areas of Cryptography and Security for the Java Card product, and emerging technologies for securing Web services, payment systems, and digital rights management. He has also worked in similar capacities on products such as the Java Message Queue, Java Web Server, Java Electronic Commerce Framework, and others. He is also an active participant in the W3C standards organization on its XML Encryption and Key Management working groups. Additionally, he cochairs the XML Key Management working group.

The author would like to thank Sai Allavarpu, Rafat Alvi, Gary Ellison, Eduardo Gutentag, Ed Julson, Komal Lahiri, Eve Maler, Vijay Rajvaidya, and Susy Struble — all from Sun Microsystems — for providing input, content, careful review, and feedback on this paper. Special thanks goes to Thomas Pfaeffle for review and formatting this document.

SUN™ Copyright 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Java, J2SE, Sun BluePrints, and Sun Inner Circle are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a).

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

SUN™ Copyright 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, Californie 95054 Etats-Unis. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, Java, J2SE, Sun BluePrints, et Sun Inner Circle sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REpondre A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



Please
Recycle



Adobe PostScript

Learn More

Get the inside story on the trends and technologies shaping the future of computing by signing up for the Sun Inner Circle program. You'll receive a monthly newsletter packed with information on the latest innovations, plus access to a wealth of resources. Register today to join the Sun Inner Circle Program at sun.com/joinic.

To receive additional information on Sun software, products, programs, and solutions, visit sun.com/software.

Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 USA Phone 800 786-7638 or +1 512 434-1577 Web sun.com



Sun Worldwide Sales Offices: Africa (North, West and Central) +33-13-067-4680, Argentina +5411-4317-5600, Australia +61-2-9844-5000, Austria +43-1-60563-0, Belgium +32-2-704-8000, Brazil +55-11-5187-2100, Canada +905-477-6745, Chile +56-2-3724500, Colombia +571-629-2323, Commonwealth of Independent States +7-502-935-8411, Czech Republic +420-2-3300-9311, Denmark +45 4556 5000, Egypt +202-570-9442, Estonia +372-6-308-900, Finland +358-9-525-561, France +33-134-03-00-00, Germany +49-89-46008-0, Greece +30-1-618-8111, Hungary +36-1-489-8900, Iceland +354-563-3010, India-Bangalore +91-80-2298989/2295454; New Delhi +91-11-6106000; Mumbai +91-22-697-8111, Ireland +353-1-8055-666, Israel +972-9-9710500, Italy +39-02-641511, Japan +81-3-5717-5000, Kazakhstan +7-3272-466774, Korea +822-2193-5114, Latvia +371-750-3700, Lithuania +370-729-8468, Luxembourg +352-49 11 33 1, Malaysia +603-21161888, Mexico +52-5-258-6100, The Netherlands +00-31-33-45-15-000, New Zealand-Auckland +64-9-976-6800; Wellington +64-4-462-0780, Norway +47 23 36 96 00, People's Republic of China-Beijing +86-10-6803-5588; Chengdu +86-28-619-9333; Guangzhou +86-20-8755-5900; Shanghai +86-21-6466-1228; Hong Kong +852-2202-6688, Poland +48-22-8747800, Portugal +351-21-4134000, Russia +7-502-935-8411, Singapore +65-6438-1888, Slovak Republic +421-2-4342-94-85, South Africa +27 11 256-6300, Spain +34-91-596-9900, Sweden +46-8-631-10-00, Switzerland-German 41-1-908-90-00; French 41-22-999-0444, Taiwan +886-2-8732-9933, Thailand +662-344-6888, Turkey +90-212-335-22-00, United Arab Emirates +9714-3366333, United Kingdom +44-1-276-20444, United States +1-800-555-9SUN or +1-650-960-1300, Venezuela +58-2-905-3800 10/03 R1.0