Web Service Security Management Using Semantic Web Techniques

Diego Zuquim Guimarães Garcia
Institute of Computing
University of Campinas
POB 6176 – Postal Code 13.084-971
Campinas, SP, Brazil
+55 19 3788 5842
diego.garcia@ic.unicamp.br

Maria Beatriz Felgar de Toledo
Institute of Computing
University of Campinas
POB 6176 – Postal Code 13.084-971
Campinas, SP, Brazil
+55 19 3788 5842
beatriz@ic.unicamp.br

ABSTRACT

The importance of the Web service technology for business, government, among other sectors, is growing. Its use in these sectors demands security concern. The Web Services Security standard is a step towards satisfying this demand. However, in the current security approach, the mechanism used for describing security properties of Web services restricts security policy specification and intersection. In environments that include loosely-coupled components, a rich description of components is needed to determine whether they can interact in a secure manner. The goal of this paper is to propose a security approach for Web services, which combines Web Services Policy Framework policies and a Web Ontology Language ontology to overcome the limitation of the current syntactic approach. The main contribution of this paper is an extended approach based on semantics-enriched security policies.

Categories and Subject Descriptors

H.3.5 [Information Storage and Retrieval]: Online Information Services – *Web-based services.*

General Terms

Security, Standardization, Languages.

Keywords

Web Service, Security, Policy, Semantic Web, Ontology, Web Services Security, Web Services Policy Framework, Web Ontology Language.

SAC'08, March 16-20, 2008, Fortaleza, Ceará, Brazil.

1. INTRODUCTION

Significant progress has been done towards making the Web service technology a suitable solution for areas such as e-business, e-government, among others. However, there are open issues that are hindering the wide scale deployment of Web services.

This paper focuses on security, which is one of these open issues. With the growing importance of the Web service technology, security becomes a critical factor for success.

A basic architecture for Web services is consolidated [1]. Extensions are being developed to include transaction processing, reliable messaging and security, among other aspects.

The current Web service security approach relies on a set of policy assertions [12] that can be used with the Web Services Policy Framework (WS-Policy) [3]. However, the policy intersection mechanism provided by WS-Policy restricts policy specification, the verification of policy compatibility and interactions among security interoperable Web services.

The goal of this paper is to propose a security approach for Web services. The proposed approach extends the current approach including Semantic Web techniques to overcome its limitation. This extension has the purpose of providing service providers and consumers the ability to interact in a secure manner.

In the approach, an ontology defined in Web Ontology Language (OWL) [15] is used to annotate Web service security policies. Security features defined in the Web Services Security (WS-Security) [13] standard are considered. Thus, policy specifications offer semantic information about security requirements and capabilities. This information can be used to verify policy compatibility and to guarantee interoperability among service participants, with respect to security aspects.

The paper major contribution is extending the Web service security approach with the inclusion of WS-Policy policies based on an OWL ontology.

The rest of the paper is organized as follows. Section 2 presents basic concepts. Section 3 discusses security in the context of Web services and motivates the security approach described in Section 4. Section 5 discusses related work. Section 6 closes the paper with contributions and future work.

2. BASIC CONCEPTS

2.1 Web Services and Policies

A Web service is an electronic service identified by a URI (Uniform Resource Identifier). XML (eXtensible Markup Language) standards are used to specify service interfaces and to invoke services through the Web. The Web service technology comprises three basic standards [1]:

- Web Services Description Language (WSDL): a format for describing the functionality of a service.
- Universal Description Discovery & Integration (UDDI): a registry that supports service publication and discovery.
- SOAP (formerly Simple Object Access Protocol): a protocol for message exchange among services.

Additional standards are under development. One example is WS-Policy [3]. It provides a model for expressing service properties as policies. Policies can be associated with XML elements, as defined in the Web Services Policy Attachment specification.

A policy is a collection of alternatives and each policy alternative is a collection of assertions. An assertion is defined as an individual requirement, capability or other property. Assertions specify characteristics that are critical to service selection and use, for instance, Quality of Service (QoS) attributes.

2.2 The Semantic Web

The Semantic Web is described as a World Wide Web evolution in which information available on the Web includes machine-accessible semantics for increasing information processing automation and improving information system interoperability [16].

It combines features of several Web standards, especially XML, which allows the creation of user-defined tagging schemes, and the Resource Description Framework (RDF), which offers a flexible data representation approach.

Based on these standards, OWL [15] extends the RDF Schema (RDFS) and provides additional vocabulary along with formal semantics for increasing semantics expressiveness and allowing ontology development. An ontology is a common knowledge conceptualization about a domain. It represents the meaning of terms in vocabularies and their relationships.

3. WEB SERVICE SECURITY

This paper focuses on message security, which is an important security aspect for Web services. It is a critical concern due to the several threats regarding message exchange.

In the Web service area, mechanisms that enhance SOAP to protect messages are defined in the WS-Security [13] specification. These mechanisms include digital signature, to protect against inappropriate message alteration, and encryption, to deal with incorrect message disclosure.

Signature and encryption may be used in specific blocks of a SOAP message, including header and body blocks. Moreover, these mechanisms may be used in overlapping message parts.

To guarantee message integrity, the digital signature mechanism uses the XML Signature [4] standard along with security tokens. A security token is a collection of claims. A claim is a statement made by an entity, for instance an identity or capability statement.

To sign elements in a SOAP envelope, signatures compliant with the XML Signature standard may be included into a security header block within the SOAP envelope. Thus, message producers are able to sign the important message parts that might be altered. In addition to allow message receivers to determine if messages were altered during their transport, receivers can verify whether security token claims apply to the producer of a message.

Encryption is based on the XML Encryption [6] standard and security tokens to offer message confidentiality. Messages may be encrypted using symmetric or asymmetric keys. After encrypting, the producer must include information about the encryption into the security header block of the message. Thus, the receiver can identify the portions to be decrypted.

In addition to encrypt portions of a message body, portions of a header may be encrypted as well. Thus, the encryption mechanism must consider the SOAP processing guidelines to guarantee both the confidentiality and the processing of the message.

The digital signature and encryption mechanisms support multiple signature formats and encryption processes. Moreover, they can be extended to support additional formats and processes.

In the security approach, the Web Services Security Policy Language (WS-SecurityPolicy) [12] provides a WS-Policy assertion set to describe how services work in terms of security. Typically, security policies are complex. Therefore, the mechanism for expressing policies must allow precise policy specifications. However, information provided by WS-SecurityPolicy does not include explicit meaning.

The inclusion of semantics into the security description mechanism offers benefits. For instance, semantics-enriched policies facilitate service negotiation since heterogeneously specified assertions are associated with ontological concepts. Moreover, as domain knowledge allows the understanding of domain semantics, an approach based on semantics enables more accurate intersections than the syntactic approach [8].

4. ENHANCED WEB SERVICE SECURITY

In this section, a description mechanism for the Web service security approach is proposed. It is based on an OWL ontology that considers the WS-Security message security model.

Message security is controlled using policies. Policies are used during different phases of the Web service life cycle:

- At design time, providers may define policies describing security properties of their services.
- At runtime, consumers may define policies stating security properties that should be offered by services.

The provider and consumer policies are intersected to compute the effective security policy. This policy indicates the interoperability between the participants in terms of security. The ontology offers a base for reasoning over policy specifications. Therefore, it supports rich policy intersections to guarantee the establishment of partnerships with suitable security levels for the participants.

4.1 Security Ontology

The ontology includes concepts for protecting Web service message exchanges. It supports a high abstraction level for dealing with security goals. The ontology classes are created to be equivalent to some XML elements of the WS-Security [13], XML Signature [4] and XML Encryption [6] standards.

In the ontology, the top-level class is called *MessageSecurity*. This class has some properties, including *keyBearing* of the *KeyBearing* type and *securityGoal* of the *SecurityGoal* type.

In Figure 1, the main ontology classes and their relationships related to message security goals are presented.

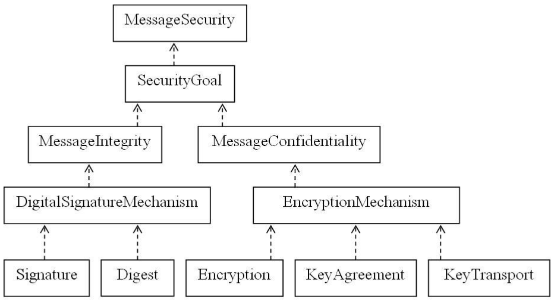

Figure 1. Security ontology: security goals.

The *SecurityGoal* class represents message security goals, including message integrity and confidentiality. These goals are captured in the ontology by defining two *SecurityGoal* subclasses: *MessageIntegrity* and *MessageConfidentiality*.

The digital signature mechanism is associated with the message integrity goal as a technique for achieving it. This mechanism is represented by the *DigitalSignatureMechanism* class, which includes properties of the following types:

- *Signature*: signature algorithms are represented by instances of this class, including DSA-SHA1 (Digital Signature Algorithm - Secure Hash Algorithm) and RSA-SHA1 (Rivest Shamir Adleman - SHA).
- *Digest*: digest algorithms are captured by this class, which includes instances such as SHA1, SHA256 and SHA512.

The encryption mechanism is associated with the message confidentiality goal. The *EncryptionMechanism* class represents this mechanism and includes properties of the following types:

- *Encryption*: this class specifies encryption algorithms. Two specializations are defined to represent block and stream encryption algorithms. Thus, in addition to have specific properties, they inherit the *Encryption* properties. The *BlockEncryption* class includes some instances, such as 3DES (Triple Data Encryption Standard), AES-128 (Advanced Encryption Standard), AES-192 and AES-256.
- *KeyTransport*: key transport algorithms are represented by instances of this class, including RSA-v1.5 and RSA-OAEP (RSA - Optimal Asymmetric Encryption Padding).
- *KeyAgreement*: this class defines key agreement algorithms. It includes the Diffie-Hellman instance.

Figure 2 presents the main ontology classes related to key bearing.

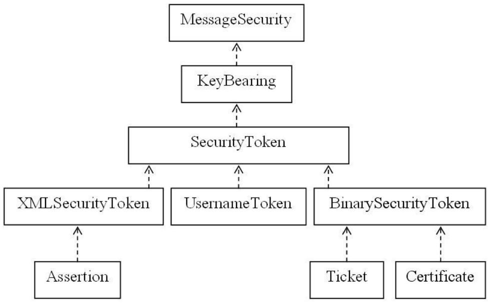

Figure 2. Security ontology: key bearing.

The *KeyBearing* class represents mechanisms for bearing security keys. An example of a key bearing mechanism is defined by the *SecurityToken* class, which is a subclass of the *KeyBearing* class.

Signature and encryption use security keys. Tokens are used to hold keys within or outside messages. There are different types of tokens with different manners of attaching them to messages. The *SecurityToken* class includes three token type subclasses:

- *UsernameToken*: username tokens offer a means of providing usernames to use Web services.
- *BinarySecurityToken*: this token type includes binary-formatted security tokens.
- *XMLSecurityToken*: this token type includes XML-based security tokens.

The *BinarySecurityToken* class has an *encodingFormat* property that indicates the token encoding format. For instance, the base64 encoding format is represented by the *Base64* instance.

Two classes are specified for binary tokens: *Certificate* and *Ticket*, which define the certificate and ticket concepts, respectively. Moreover, specializations are defined, including *X.509Certificate* for the *Certificate* class and *KerberosTicket* for the *Ticket* class.

The *X.509Certificate* class includes several instances, which represent X.509 versions: X.509 Version 3, X.509 PKCS7 (Public-Key Cryptography Standards), X.509 PKI (Public-Key Infrastructure) Path Version 1 and X.509 Version 1. Some *KerberosTicket* instances are also defined, including instances that represent Kerberos Version 5 AP-REQ (Application Request) and GSS (Generic Security Service) Kerberos Version 5 AP-REQ.

The same scheme is used for XML tokens. The *Assertion* class is a *XMLSecurityToken* subclass. It represents security assertions. *SAMLAssertion* is defined as a specialization for this class. *SAML-v1.1* and *SAML-v2.0* are specified as instances of the *SAMLAssertion* class. These instances represent different versions of the Security Assertion Markup Language (SAML), including SAML Version 1.1 and SAML Version 2.0.

The ontology offers a flexible approach to support interoperability, which is a requirement in Web service environments. It can be extended with additional message security techniques and technologies by including new classes and properties.

4.2 Security Policy

In the approach, consumers and providers specify security requirements and capabilities of services using policies. The basic structure of policies is compliant with the WS-Policy normal form, which is shown in Figure 3.

```
<wsp:Policy
   xmlns:wsp = "…/ws/2004/09/policy" ...>
   <wsp:ExactlyOne>
      ( <wsp:All>
         ( <Assertion ...> ... </Assertion> )*
      </wsp:All> )*
   </wsp:ExactlyOne>
</wsp:Policy>
```

Figure 3. Basic policy structure.

In Figure 3, *wsp* is a prefix for the WS-Policy namespace URI. In addition to the components included into the normal form, other general-purpose components, also specified by WS-Policy, can facilitate policy manipulation. A policy includes the following components:

- *Policy*: the root element that indicates a policy.
- *Name*, *Id*: two kinds of policy identification may be used. Either the policy is associated with an absolute URI, using the *Name* attribute, or it is associated with a reference within the enclosing document, using the *Id* attribute.
- *PolicyReference*: the *PolicyReference* element may be used to include the content of a policy into another policy.
- *Service*: a provider policy includes a *Service* element to describe details of the service implementation for which the policy has been specified. A consumer policy includes this element to specify details of the service type to which the policy applies.
- Operators: in a policy, policy alternatives are grouped into an *ExactlyOne* operator. The *All* operator represents a policy alternative and groups the alternative assertions. Policy operators may be recursively nested.
- Assertions: policy assertions are elements that describe security requirements and capabilities of Web services. A policy assertion may contain nested assertions and a nested policy.
- *Optional*: the *Optional* attribute may be used to indicate that a policy assertion is optional.

It is in the assertion components that a policy is specialized. Assertions use concepts from the security ontology in opposition to the current approach, which uses assertions specified in WS-Policy supplementary specifications.

Policy assertions may be specified in more general or specific manners. For instance, a policy may be defined for a service consumer that demands services capable of processing security certificates; or a policy can be used to indicate that a Web service requires a specific certificate format, issued by a specific certification authority.

Following, examples of assertions extracted from a policy are shown. They illustrate a policy that defines properties of a service.

Figure 4 presents a token assertion. It indicates that the service uses the X.509 Version 3 token (Line 01) with the base64 format (Line 03). The *Id* attribute (Line 02) specifies the local identification of the token element. The *sec* and *wsu* prefixes are associated with the namespace URIs of the security ontology and the WS-Security-Utility XML Schema Definition, respectively.

```
01 | <sec:X.509-v3
02 |    wsu:Id = "X.509Token"
03 |    EncodingFormat = "sec:Base64"/>
```

Figure 4. Token assertion example.

The encryption assertion in Figure 5 indicates that the body of SOAP messages (Line 06) sent by the service is encrypted using the 3DES (Line 01) algorithm. Line 03 shows that the encryption mechanism uses the token defined in Figure 4.

```
01 | <sec:3DES>
02 |    <sec:Token>
03 |       <sec:Reference URI = "#X.509Token"/>
04 |    </sec:Token>
05 |    <sec:EncryptedParts>
06 |       <sec:Body/>
07 |    </sec:EncryptedParts>
08 | </sec:3DES>
```

Figure 5. Encryption assertion example.

The example in Figure 6 shows the description of the message integrity protection offered by the service. It asserts that the service uses the RSA-SHA1 signature algorithm (Line 01). This algorithm is employed to sign the body (Line 06) and the timestamp header (Line 07) of SOAP messages. The token defined in Figure 4 is used by the signature mechanism (Line 03).

```
01 | <sec:RSA-SHA1>
02 |    <sec:Token>
03 |       <sec:Reference URI = "#X.509Token"/>
04 |    </sec:Token>
05 |    <sec:SignedParts>
06 |       <sec:Body/>
07 |       <sec:TimestampHeader/>
08 |    </sec:SignedParts>
09 | </sec:RSA-SHA1>
```

Figure 6. Signature assertion example.

Policy operations defined in the WS-Policy specification may be used for processing security policies. For example, the intersection operation is used to determine providers whose security policies are suitable for a given consumer policy.

The intersection operation matches consumer and provider policies. Policies are compatible if there is at least one pair of compatible alternatives between a consumer policy and a provider policy. Policy alternatives are compatible if the capability assertions of one alternative satisfy the requirement assertions of the other alternative. The compatibility between assertions is determined by using OWL-based operators, such as "subclass of" and "instance of".

For instance, if a consumer requirement list includes the *Message-Integrity* concept, then the service required by the consumer must offer message integrity protection. Thus, the service for which the policy assertion in Figure 6 was defined is considered a suitable partner for this consumer. The domain knowledge captured by the ontology allows determining that the service supports message integrity, as required by the consumer.

5. RELATED WORK

Some studies in the area of Web service security employ the Semantic Web technology. These studies focus on authorization and do not deal with message security. They are discussed below, including work in the general area of Web service QoS [10].

Web service QoS ontologies are described in [18, 5]. The DAML-QoS ontology [18] is realized using the OWL predecessor. The QoSOnt ontology [5] is realized using OWL.

These contributions do not consider policies as a mechanism for specifying non-functional characteristics of Web services. Policies for distributed systems offer a way for defining capabilities and requirements. Thus, they are a suitable choice for specifying QoS attributes of Web services. Moreover, policy intersection may be used to improve service selection.

Maximilien and Singh [9] propose a QoS ontology and a policy language. Kagal et al [7] use the Semantic Web technology to handle authorization and privacy. Shields et al [17] propose an approach for the specification of access control policies using an ontology. In [14], Web Services Agreement Specification (WS-Agreement) agreements are enriched with semantics.

These contributions do not support interoperability and full integration into the Web service architecture. Some important standards are not used, including WS-Policy and OWL:

- WS-Policy gives a flexible open-standard that matches with fundamental requirements of Web services: simplicity and extensive support in industry [1]. Currently, WS-Policy is a popular standard to be aggregated into the Web service architecture.

- OWL is a World Wide Web Consortium (W3C) recommendation. Moreover, the use of OWL allows the integration of the ontologies with the widely accepted Web Ontology Language for Services (OWL-S) ontology.

Problems concerning the use of WS-Policy are discussed in [11, 2]. Assertions of different domains may have dependencies that invalidate a policy. The WS-Policy model relies on the design of new schema components with associated specifications for each domain. This increases the complexity of policy maintenance. In this paper, WS-Policy is used and OWL is included into the proposed approach for addressing these problems.

6. CONCLUSIONS

A mechanism that combines WS-Policy and OWL was introduced in this paper to support security information management. Policies are used to describe security requirements and capabilities of service consumers and providers. The approach enables the specification of services that implement mechanisms compliant with WS-Security. An ontology helps specifying semantics-enriched policies and reasoning about them during policy intersection.

The main contribution of this paper is extending the Web service security approach with the use of semantic security policies to enable service participants to conduct secure interactions.

Future work includes the consideration of other QoS attributes and facilities for mobile Web services.

Acknowledgements

This project is supported by FAPESP.

7. REFERENCES

[1] G. Alonso, F. Casati, H. Kuno, and V. Machiraju. *Web Services: Concepts, Architectures and Applications*. Springer, 2004.

[2] A. H. Anderson. Domain-independent, composable Web services policy assertions. In *Proc. of the IEEE Int'l Ws. on Policies for Distrib. Syst. and Net.*, pages 149–152. IEEE, 2006.

[3] S. Bajaj, et al. Web Services Policy 1.2 - Framework. W3C, Apr. 2006. http://www.w3.org/Submission/2006/SUBM-WS-Policy-20060425/, accessed on 08/2007.

[4] M. Bartel, J. Boyer, B. Fox, B. LaMacchia, and E. Simon. XML Signature. W3C, Feb. 2002. http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/, accessed on 08/2007.

[5] G. Dobson, R. Lock, and I. Sommerville. QoSOnt: a QoS ontology for service-centric systems. In *Proc. of the EUROMICRO Conf. on Soft. Eng. and Adv. Appl.*, pages 80–87. IEEE, 2005.

[6] T. Imamura, B. Dillaway, and E. Simon. XML Encryption. W3C, Dec. 2002. http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/, accessed on 08/2007.

[7] L. Kagal, M. Paolucci, N. Srinivasan, G. Denker, T. Finin, and K. Sycara. Authorization and privacy for semantic Web services. *IEEE Intelligent Systems*, 19(4):50–56, 2004.

[8] V. Kolovski, B. Parsia, Y. Katz, and J. A. Hendler. Representing Web service policies in OWL-DL. In Y. Gil, E. Motta, V. R. Benjamins, and M. A. Musen, editors, *Proc. of the 4th Int'l Semantic Web Conference*, volume 3729 of *LNCS*, pages 461–475. Springer, 2005.

[9] E. M. Maximilien and M. P. Singh. A framework and ontology for dynamic Web services selection. *IEEE Internet Computing*, 8(5):84–93, 2004.

[10] D. A. Menascé. QoS issues in Web services. *IEEE Internet Computing*, 6(6):72–75, 2002.

[11] N. K. Mukhi and P. Plebani. Supporting policy-driven behaviors in Web services: experiences and issues. In *Proc. of the 2nd Int'l Conference on SOC*, pages 322–328. ACM, 2004.

[12] A. Nadalin, M. Goodner, M. Gudgin, A. Barbir, and H. Granqvist. WS-SecurityPolicy 1.2. OASIS, Jul. 2007. docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/ws-securitypolicy-1.2-spec-os.pdf, accessed on 08/2007.

[13] A. Nadalin, C. Kaler, R. Monzillo, and P. Hallam-Baker. Web Services Security 1.1. OASIS, Feb. 2006. http://oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf, accessed on 08/2007.

[14] N. Oldham, K. Verma, A. Sheth, and F. Hakimpour. Semantic WS-Agreement partner selection. In *Proc. of the 15th Int'l WWW Conference*, pages 697–706. ACM, 2006.

[15] P. F. Patel-Schneider, P. Hayes, and I. Horrocks. OWL Web Ontology Language Semantics and Abstract Syntax. W3C, Feb. 2004. w3.org/TR/owl-semantics/, accessed on 08/2007.

[16] N. Shadbolt, W. Hall, and T. Berners-Lee. The Semantic Web Revisited. *IEEE Intelligent Syst.*, 21(3):96-101, 2006.

[17] B. Shields, O. Molloy, G. Lyons, and J. Duggan. Using semantic rules to determine access control for Web services. In *Proc. of the 15th Int'l WWW Conference*, pages 913–914. ACM, 2006.

[18] C. Zhou, L.-T. Chia, and B.-S. Lee. DAML-QoS ontology for Web services. In *Proc. of the IEEE Int'l Conference on Web Services*, page 472–479. IEEE CS, 2004.