Chapter II

Privacy Issues in the Web Services Architecture (WSA)

Barbara Carminati, University of Insubria at Como, Italy
Elena Ferrari, University of Insubria at Como, Italy
Patrick C. K. Hung, University of Ontario Institute of Technology (UOIT), Canada

Abstract

A Web service is a software system that supports interoperable application-to-application interactions over a network. Web services are based on a set of XML standards such as Universal Description, Discovery and Integration (UDDI), Web Services Description Language (WSDL), and Simple Object Access Protocol (SOAP). Recently, there have been increasing demands and discussions about Web services privacy technologies in the industry and research community. To enable privacy protection for Web service consumers across multiple domains and services, the World Wide Web Consortium (W3C) published a document called “Web Services Architecture (WSA) Requirements” that defines some fundamental privacy requirements for Web services. However, no comprehensive solutions to the various privacy issues have been so far defined. For these reasons, this chapter will focus on privacy technologies by first discussing the main privacy issues in WSA and related protocols. Then, this chapter illustrates the standardization efforts going on in the context of privacy for Web services and proposes different technical approaches to tackle the privacy issues.
Introduction

Privacy is a state or condition of limited access to a person (Schoeman, 1984). Information privacy relates to an individual’s right to determine how, when, and to what extent information about the self will be released to another person or to an organization (Leinon-Kilpi, Dassen, Gasull, Lemonidou, Scott, & Arndt, 2001). Threats to information privacy can come from insiders and from outsiders in each organization (Fischer-Hubner, 2001). In general, privacy policies describe what information an organization collects from individuals (e.g., consumers) and what (e.g., purposes) they do with it. Many studies show that good privacy protection is an important factor to generate a good business (Bennett, 1997).

In the U.S., the Privacy Act of 1974 (Fischer-Hubner, 2001) requires that federal agencies grant individuals access to their identifiable records that are maintained by the agency, ensure that existing information is accurate and timely, and limit the collection of unnecessary information and the disclosure of identifiable information to third parties. In Canada, the Personal Information Protection and Electronic Documents Act (PIPEDA) governs privacy issues related to collected data, including those collected via traditional Web-based applications (PIPEDA, 2005). On the reverse, the Europe Union Data Protection Directive (Steinke, 2002) contains two statements contradicting the U.S. one. The first statement requires that an organization must inform individuals about the purposes for which it collects and uses information about them, how to contact the organization, and the types of third parties to which it discloses the information. The second statement requires that personal data on EU citizens may only be transferred to countries outside the 15 nation block that adopt these rules or are deemed to provide “adequate protection” for the data. As a result, these statements imply that no information of any EU citizen can be transferred to the U.S. due to the conflicts between two privacy acts. Consequently this creates obstacles for conducting business activities between these two regions. To overcome the difficulties, the U.S. government already has a voluntary scheme called “Safe Harbour” to provide an adequate level of data protection which can safeguard transfers of personal data to the U.S. from EU. The U.S. companies doing business in the EU must certify to the Commerce Department that they will follow the regulations of the EU directive. Any violation would be subject to prosecution by the Federal Trade Commission (FTC) for deceptive business practices.

For instance, biometrics (Grippink, 2001) and healthcare applications (Cheng & Hung, 2005) have to seriously enforce privacy protection. Under the Health Insurance Portability and Accountability Act of 1996 (HIPAA) privacy rules (HL7, 2004) in the U.S., Protected Health Information (PHI) includes individually identifiable health information related to past, present, and future physical and mental health conditions, as well as the past, present, and future payment for the provisions of healthcare to an individual. HIPAA provides a set of standard policies that the healthcare providers must exercise in order to protect a patient’s privacy. HIPAA provides a standard set of electronic transaction formats and regulations to ensure the privacy and security of healthcare-related transactions. Similarly, the Personal Health Information Protection Act of 2004 (PHIPA) in Canada establishes rules for the collection, use, and disclosure of personal health information about individuals that protect the confidentiality of that information.
and the privacy of individuals with respect to that information, while facilitating the effective provision of healthcare (PHIPA, 2005). For example, applied research in e-health services shows that integrated views can result in different information formats being provided through the use of intelligent electronic health data access, analysis and visualization tools (Lacroix, 2002; Tan & Hung, 2005). However, e-health services that link e-patients’ health datasets to other sources of patient-specific data pose significant risks to the privacy of stored patient data. Indeed, these “micro-datasets” often contain identifiable and sensitive information such as genetic or demographic data about individuals, for example, name, age, sex, address, phone number, employment status, family composition, and DNA (Quantin, Allaert, & Dusserre, 2000). In this sense, not only will disclosure of sensitive information of particular individuals potentially create personal embarrassment, but it may also, very possibly, lead to social ostracism (France, 1996).

All of these facts state that privacy is a very important topic, while there are more and more business applications deployed on the Internet nowadays. As Web services are becoming more and more popular for supporting different business applications, there are also increasing demands and discussions about Web services privacy technologies in the industry and research community. The information exchange in such a Web services-based business environment must be protected by privacy-enhancing technologies, especially if the information may be sensitive (Senicar, Jerman-Blazic, & Klobucar, 2003). Thus, it is required to have a privacy framework for supporting the Web services-based businesses. To enable privacy protection for Web service consumers across multiple domains and services, the World Wide Web Consortium (W3C) published a document called “Web Services Architecture (WSA) Requirements” that defines some specific privacy requirements for Web services. However, no comprehensive solutions to the various privacy issues have been so far defined. For these reasons, this chapter will focus on privacy technologies by first discussing the main privacy issues in WSA and related protocols. Then, this chapter illustrates the standardization efforts going on in the context of privacy for Web services, and proposes different technical approaches to tackle the privacy issues.

The remainder of this chapter is organized as follows: Section iv discusses the motivation and background information of this chapter. Next, Section v gives a literature review. Section vi introduces privacy policy enforcement in WSA. Then, Section vii addresses privacy issues in Web services discovery agencies. Section viii discusses strategies for privacy enforcement in Web services discovery agencies. Section ix presents the future trends, whereas Section x discusses the conclusions.

**Motivation and Background**

In this chapter, we first give the motivations behind the chapter, and then we present some background information of security mechanisms that are needed to understand the solutions proposed in Section vii.
Motivation

Figure 1 depicts different privacy concerns existing in the context of Web services architecture (WSA). On the left hand side, the users interact with the Web services application via information exchanges. The information exchanges between the users and Web services application always contain different confidential and sensitive data. Referring to the publish/find/bind model in Web services (Mohen, 2002), one can imagine that Web services providers publish their Web services descriptions (e.g., WSDL documents) at registries (e.g., UDDI) for the public to access. Then, the users (Web service requestors) find the appropriate Web services at the registries. In many cases, there may have a mediator (i.e., a service locator) that helps to find appropriate Web services for requestors. This process is called matchmaking (Zhang & Zhang, 2002). Once the Web services are found, the Web services application is trying to bind to each Web service via SOAP messages.

From the user’s point of view, privacy concerns mainly raise in the registries and Web services. For example, the users may want the registries to protect their privacy such as their identities and what information they have retrieved from the registries. In addition, the users may also want to validate the privacy policies of business entities and services based on their privacy preferences (W3C, 2002a). It means that the Web services application may only bind to those Web services satisfying their privacy policies. From another point of view, the privacy policies defined in UDDI for specific business entities and services must be consistent with the privacy policies defined in the WSDL documents of Web services.

As what we mentioned, there is little research on addressing Web services privacy. Very little privacy standards exist beyond a principal statement made by IBM and Microsoft: “organizations creating, managing, and using Web services will often need to state their privacy policies and require that incoming requests make claims about the senders’ adherence to these policies” (IBM, 2002). For illustration, Figure 2 shows an e-health
A database application example involving three entities: Web services application, Web service, and e-health care database. The Web services application can be any healthcare application at a health institute that is connected to a Web service at another health institute over the Internet. You can assume that the Web service is used as an interface to receive the request (e.g., retrieve/store healthcare data) from the application and then communicate with the e-health care database at the backend (e.g., read/write data). Once the request is completed, the Web service returns a result (e.g., acknowledgment or health data) to the application. Let’s assume that this example is a healthcare scenario in the U.S. Thus, there is a HIPAA compliant privacy policy enforced at the Web service according to the law (Cheng & Hung, 2005). It means that every request must be checked and verified with the privacy policy. If the request is eligible according to the privacy rules set in the HIPAA, the Web service will handle the request and initiate the process. Otherwise, the Web service will deny the request. In general, there are six rights that HIPAA gives patients with regards to their PHI as follows (HSS, 2004a):

1. The right to view and make a copy of a patient’s own medical records.
2. The right to request the correction of inaccurate health information.
3. The right to find out where PHI has been shared for purposes other than care, payment, or healthcare operations.
4. The right to request special restrictions on the use or disclosure of PHI.
5. The right to request PHI to be shared with the patient in a particular way.
6. The right to file complaints.

The scenario described so far is good enough for describing and justifying the five “Web Services Architecture (WSA) Requirements” introduced by W3C (Ref: AC020) for enabling privacy protection for the consumer (user) of a Web service across multiple domains and services (W3C, 2002b):

- **AR020.1**: The WSA must enable privacy policy statements to be expressed about Web services;
- **AR020.2**: Advertised Web service privacy policies must be expressed in P3P (W3C, 2002c);
• **AR020.3**: The WSA must enable a consumer to access a Web service’s advertised privacy policy statement;

• **AR020.5**: The WSA must enable delegation and propagation of privacy policy; and

• **AR020.6**: Web services must not be precluded from supporting interactions where one or more parties of the interaction are anonymous.

The major purpose of these WSA requirements is to enforce privacy policies in the context of WSA, where AR020.6 requirement is strongly related to workflow/business process integration issues (Hung & Chiu, 2003). Applying privacy policies in the context of WSA is one of the first important steps to develop a technical framework for supporting Web services privacy policies. Especially, they recommend adopting P3P technologies to define privacy policies. However, these WSA requirements are not covering all the related issues to be investigated for the real scenario. For instance, one can imagine that vocabularies vary in different business applications. Thus, it is essential to have a vocabulary for an independent privacy meta-language for WSA. The WSA requirement AR020.5 points out another relevant issue: the privacy policies also have to be enforced in a delegation and propagation situation, as shown in Figure 1. Web services may delegate some sub-activities that are decomposed from the assigned activities to other Web services. This assignment process is also called delegation or propagation (IBM, 2002). Nevertheless, the other important area to be further investigated is the privacy concerns in the intermediaries that pass the SOAP messages between the Web services application and Web services/registries, as shown in Figure 1. In the following sections, we focus on two main aspects: privacy issues arising in Web services discovery agencies (Section vii) and the definition of a framework for privacy policy enforcement (Section vi), compliant with the WSA requirements introduced in this section.

### Background on Security Mechanisms

Ever since security has become an essential asset for all information systems, several security solutions for protecting data have been proposed, (see Stallings [2000] for an overview). In general, these solutions exploit access control mechanisms for enforcing data confidentiality and integrity and cryptography-based solution for assuring confidentiality and authenticity during data transmission. In what follows, we give a brief overview of these techniques which will be used in Section vii for privacy enforcement in the context of WSA.

#### Access Control Mechanisms

The task of an access control mechanism is to avoid unauthorized operations on the managed data. The access control mechanism (or reference monitor) is a software module

Copyright © 2006, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.
that intercepts each access request submitted by a user to the system and determines whether the access should be partially or totally authorized or whether it should be denied. In order to decide what is authorized for a user, the access control mechanism consults a set of authorizations, which state for each user the rights he/she can exercise on the objects managed by the system. In its basic format, an authorization consists of three main components: a subject, who is the entity to which the authorization is granted; an object, which is the resource to which the access is granted; and an access mode, specifying the action that the subject can exercise on the object.

The authorizations are specified according to the access control policies adopted by the system, that is, the high level security rules stating how the information should be managed. More precisely, the access control policies are translated in authorizations by means of an access control model, which specifies the characteristics of authorization basic components (subjects, objects, and access modes) and states how access control should take place. In the past years, several access control models have been proposed, for example, mandatory (Bell & LaPadula, 1975), discretionary (Gollmann, 1999), role-based (Ferraiolo, Sandhu, Gavrila, Kuhn, & Chandramouli, 2001), and credential-based (Winslett, Ching, Jones, & Slepchin, 1997), which mainly differ on the subject and object specification and in the way they perform access control.

**Encryption Algorithms**

In general, the main role covered by encryption primitives in security mechanisms is to obscure the data, thus making it inaccessible by users not supplied with the appropriate decryption keys. Encryption algorithms can be grouped into two main classes: symmetric and asymmetric algorithms. In symmetric encryption (e.g., DES, Blowfish, RC5, AES), a unique key, called secret key, is used to both encrypt and decrypt the data. Thus, according to symmetric encryption, if user A wants to send confidential data to user B, A has to encrypt the data with the secret key that it shares with B. The main drawback of this approach is that it requires a secure channel for the secret key exchange.

By contrast, asymmetric encryption (e.g., RSA, ElGamal) exploits two distinguished keys: a public key that can be published and distributed, and the private key, which must be kept secret. In general, the asymmetric algorithms are defined in such a way that data encrypted with the public key can be only decrypted using the corresponding private key. This property makes the asymmetric encryption able to ensure data confidentiality. Indeed, according to asymmetric encryption, with each user is associated a pair of private and public keys. Thus, if user A wants to send confidential data to user B, A has to encrypt the data with the public key of B. Since B is the only one that has the corresponding private key, he/she is the only one able to decrypt the data sent by A. Some asymmetric algorithms (e.g., RSA) can also be used to verify the source of the information. In this case, if user A wants to send user B data and makes it able to verify its authenticity, A has to encrypt the data with its private key. If B is able to decrypt the data with the public key that corresponds to the claimed sender, it is ensured about the source authenticity.
Hash Functions

Hash functions (e.g., MD5, SHA-1) cover an important role in several authentication mechanisms (like authentication protocols, digital signatures). A function H() is a hash function if and only if it satisfies the following properties: (1) taking as input a message of arbitrary length, H always returns a fixed-length output; (2) given a hash value h, it must be computationally infeasible to find a value x such that H(x)=h (one-way property); (3) it must be computationally infeasible to find a pair x, y such that H(y)=H(x) (strong collision resistance property). Given the above-mentioned properties, the hash function can be exploited to produce a fingerprint of the message to be authenticated. Indeed, one of the most well-known usages of hash function is in digital signatures. The digital signature of a message M consists of the encryption of the hash value of M, called digest, with the private key of the user. Thus, when user A wants to send a message M to user B and make it able to verify its authenticity, A computes the digest of M and encrypts it with its private key (i.e., it digitally signs the message). When B receives the message and its digital signature, it decrypts the digital signature with the public key of A, obtaining thus the hash value of the original message. Then, B computes the hash value of the received message and compares this value with the decrypted hash value. Due to the properties of hash functions and asymmetric encryption, if the hash values match, B is assured that the message has not been modified and has been effectively created by A.

Literature Review

In the recent years, there are increasing demands and discussions about privacy technologies for supporting Web services-based applications. For example, WS-Policy describes the business policies to be enforced on intermediaries and endpoints (IBM, 2002). The business policies contain certain requirements such as required security tokens, supported encryption algorithms, and privacy rules. The WS-Policy is represented by a policy expression, that is, an XML Infoset representation of one or more policy statements. The WS-Policy includes a set of general messaging-related assertions defined in WS-PolicyAssertions (IBM, 2002) and a set of security policy assertions related to supporting the WS-Security specification defined in WS-SecurityPolicy (IBM, 2002). In particular, WS-Security describes how to attach security tokens such as X.509 certificates to SOAP messages (IBM, 2002). However, the current WS-Policy specification does not discuss the privacy rules in detail. Even though WS-Privacy is mentioned to describe a model for defining subject privacy preferences and organizational privacy practice statements, WS-Privacy has not been developed yet (IBM, 2002).

Next, the Platform for Privacy Preferences Project (P3P) working group at W3C develops the P3P specification for enabling Web sites to express their privacy practices (W3C, 2002). On the other hand, P3P user agents allow users to automatically be informed of site practices and to automate decision making based on the Web sites’ privacy practices. Thus, P3P also provides a language called P3P Preference Exchange Language.
1.0 (APPEL1.0), to be used to express users’ preferences for making automated or semi-automated decisions regarding the acceptability of machine-readable privacy policies from P3P enabled Web sites (W3C, 2002a).

Referring to the illustrative e-health database application example, let’s assume that the Web services application sends a request to store the health data (e.g., Hospital, Treatment, and Pharmacy) to the Web service. Before the application submits the health data to the Web service, the application must check the P3P privacy policy (Figure 3) posted on the Web service with its APPEL privacy preference (Figure 4). The privacy policy at the Web service states that “The Web service is the only recipient collected the data (Hospital, Treatment, and Pharmacy) for healthcare purposes.” Referring to the P3P privacy policy shown in Figure 3, the assertion <STATEMENT></STATEMENT> describes the data practices as applied to data elements. The assertion <RECIPIENT></RECIPIENT> describes the legal entity, or domain, beyond the service provider and its agents where data may be distributed. In this case, the recipient is <ours/> that means only the Web service collected the data. The assertion <PURPOSE></PURPOSE> describes the reason(s) for data collection and use. In this case, the purpose is for healthcare only. Then, the assertion <DATA-GROUP></DATA-GROUP> describes the data to be transferred or inferred. This case covers the data in the hospital, treatment, and pharmacy. On the Web services application side, the privacy preference states that “The Web services must be the only recipient to collect the data (Hospital, Treatment, and Pharmacy) for health care purposes.” Referring to the APPEL privacy preference shown in Figure 4, there are two statements (rules) defined as follows. Only the Web service <ours/> can be the recipient of the data of hospital, treatment, and pharmacy, and the purpose must be only for <healthcare/>. Based on this example, the P3P privacy policy and the Web service will submit the health data to the Web service; the Web service will write the data into the e-health care database shown in Figure 2.

Figure 3. An illustrative P3P privacy policy

```xml
<POLICY>
  ...
  <!-- evidence (abbreviated) -->
  ...
  <STATEMENT>
    <RECIPIENT><ours/></RECIPIENT>
    <PURPOSE><healthcare/></PURPOSE>
    <DATA-GROUP>
      <DATA ref="#hospital.*"/>
      <DATA ref="#treatment.*"/>
      <DATA ref="#pharmacy.*"/>
    </DATA-GROUP>
  </STATEMENT>
  ...
  <!-- evidence (abbreviated) -->
  ...
</POLICY>
```
Furthermore, Lategan and Olivier (2002) propose a conceptual model for enhancing the decision making at the user agents by using the Chinese Wall security policy based on the P3P framework. Though the P3P framework is not mainly designed for supporting Web services privacy policies, the P3P working group is currently studying the feasibility of applying a revised version of P3P into Web services privacy policy framework.

Next, the Enterprise Privacy Authorization Language (EPAL) technical specification is used to formalize privacy authorization for actual enforcement within an intra- or inter-enterprise for business-to-business privacy control (IBM, 2003). EPAL concentrates on the privacy authorization by abstracting data models and user-authentication from all deployment details. The goal behind EPAL is to enable an enterprise to encode its privacy-related data-handling policies and practices in XML for facilitating privacy-enforcement in enterprise information systems. Its recent emergence as a fine-grained, privacy-related language standard is in response to the irreversible trend of having more

Figure 4. An illustrative APPEL privacy preference

```xml
<appel:RULE behavior="request.store">
    ...
    <-- evidence (abbreviated) -->
    ...
    <POLICY>
        <STATEMENT>
            <RECIPIENT appel:connective="or-exact">
                <ours/>
            </RECIPIENT>
            <DATA-GROUP appel:connective="or-exact">
                <DATA ref="#hospital.*"/>
                <DATA ref="#treatment.*"/>
                <DATA ref="#pharmacy.*"/>
            </DATA-GROUP>
        </STATEMENT>
        <STATEMENT>
            <PURPOSE appel:connective="or-exact">
                <healthcare/>
            </PURPOSE>
            <DATA-GROUP>
                <DATA>
                    <CATEGORIES appel:connective="or-exact">
                        <DATA ref="#hospital.*"/>
                        <DATA ref="#treatment.*"/>
                        <DATA ref="#pharmacy.*"/>
                    </CATEGORIES>
                </DATA>
            </DATA-GROUP>
        </STATEMENT>
    </POLICY>
    <-- evidence (abbreviated) -->
    ...
</appel:RULE>
```
and more dynamically formed and evolving federations of organizations in this e-business era. The EPAL vocabulary includes lists of hierarchies of data-categories, user-categories, and purposes, and also sets of actions, obligations, and conditions. Data-categories are used to define different categories of collected data handled differently from a privacy perspective such as financial data. User-categories are used to describe the users or groups accessing collected data such as investors. Purposes are used to model the intended service for which data is used such as an investment. Actions are used to model how the data is used such as buy and sell. Obligations are used to define actions that must be taken by the environment of EPAL such as, “No personal data will be released to any unauthorized party.” In particular, conditions are Boolean expressions such as, “all sellers must have signed the confidential agreement form.” A vocabulary may be shared by more than one enterprise. On the other hand, the EPAL policy defines the privacy authorization rules that allow or deny actions on data-categories by user-categories for certain purposes under certain conditions while mandating certain obligations. EPAL work together with the access control markup language XACML (OASIS, 2003a) as well as the recent development access control in Semantic Web (Agarwal, Sprick, & Wortmann, 2004) to achieve privacy data protections.

Referring to the illustrative e-health database application example, let’s assume that the Web services application (healthcare system) sends a request to retrieve health data to the Web service for healthcare purposes. Moreover, let’s assume that the Web service adopts the EPAL privacy policy shown in Figure 5. In particular, this privacy policy contains a privacy authorization rule that allows healthcare systems (i.e., <user-category/>) to “retrieve” (i.e., <action/>) the health data from Hospital, Treatment, and

![Figure 5. An illustrative EPAL policy](image-url)
Pharmacy (i.e., <data-category/>)) for healthcare purposes (i.e., <purpose/>). In this case, the Web service will read the health data from the e-health care database and return the health data to the application.

There are a few number of research works related to Web services privacy policies. For example, Langheinrich (2002) discusses a privacy awareness system targeted at ubiquitous computing environments. In the privacy awareness system, privacy proxies, which are implemented as a set of SOAP services, handle privacy relevant interactions between data subjects and data collectors but also provide access to specific user control capabilities disclosed in the privacy policy. Though this work is not mainly target on the context of Web services, it provides a basic framework for implementing Web service privacy-enhancing technologies in the future. Further, Rezgui et al. (2002) view privacy in Web services from the aspects of user privacy, data privacy, and service privacy. In particular, service privacy includes three types of policies: the usage policy stating the purposes for which the information collected can be used; the storage policy specifying whether and until when the information collected can be stored by the Web service; the disclosure policy stating if and to whom the information collected from a given user can be revealed. In addition, they have also applied their model into a digital government architecture that aims at preserving citizens’ privacy (Rezgui et al., 2002). Also, Tumer, Dogac, and Toroslu (2003) present a semantic-based privacy framework for Web services by using DAML-S.

**Privacy Policy Enforcement in WSA**

Referring to the WSA requirements introduced in Section iv, enabling privacy protection for the consumer of a Web service across multiple domains and services only defines the guidelines according to which privacy enhancing technologies for Web services should be designed (Hung, Ferrari, & Carminati, 2004). The first important point is that from the language point of view, different business applications certainly will wish to adapt the privacy policies to their own circumstances. Here we propose the idea of having a vocabulary independent framework, able to adapt to different Web services applications. Figure 6 shows the concept of domain specific vocabularies for supporting different types of business applications in the proposed privacy authorization language framework. For example, one can imagine that there exists a financial or medical application specific vocabulary (Webmethod, 2002). The vocabulary can be described by using DAML-S (DAML, 2003) or the OWL Web Ontology Language (W3C, 2003a). DAML-S defines an upper ontology for describing the semantics of Web services. OWL is an XML language proposed by W3C for defining Web ontology. OWL ontology includes descriptions of classes, properties, and their instances, as well as formal semantics for deriving logical consequences in entailments (Carminati, Ferrari, & Bertino, 2005).

The second point is defining a protocol for policy enforcement. As far as AC020 is concerned, P3P is proposed to be the privacy authorization language in WSA. An example of such a protocol is presented in (Hung et al., 2004), and discussed below assuming that privacy policies and preference exchange rules are specified using P3P.
Parties involved in the protocol are: 

- **Web services (provider)** — *A*, **Web services consumer** — *B*, **Discovery agency** — *C*, **SOAP intermediary** — *D*, and **Web services partner** — *E* (see Figure 7). The interactions are described as follows:

1. **A→C**: Request discovery agency’s privacy policy in P3P.
2. **A**: The Web services provider matches its privacy preferences in APPEL with discovery agency’s privacy policy.
3. **A→C**: If they match, the Web services provider publishes service in WSDL and related privacy policies in P3P. Otherwise, the provider can decide what to do.
4. **B→C**: Find an appropriate Web service via UDDI.
5. **B**: The Web Service Consumer matches discovery agency’s and service provider’s privacy policies in P3P with its privacy preferences in APPEL.
6. **B→D**: If they match, the Web services consumer attempts to bind to the Web service via SOAP message, by attaching a P3P privacy policy in the SOAP header for enforcing SOAP intermediaries to obey.

7. **D→A**: If the SOAP intermediaries all obey the privacy policy, the SOAP message will be passed to the Web service.

8. **A→B**: Request consent from the Web services consumer for propagation and delegation of information if there is a need.

9. **A→E**: Once the consent is given from the Web services consumer, the Web services provider will pass the information to the Web services partner if and only if the privacy policy at the partner is also compatible with the provider’s.

It is our belief that such issues, independent from the used privacy language, are the basic steps towards the standardization of Web services privacy technologies in the coming years.

### Privacy Issues in Web Services

#### Discovery Agencies

In this section, we focus on a particular class of privacy issues arising in the WSA, that is, those referring to discovery agencies. Discovery agencies provide a searchable set of service descriptions in centralized or distributed UDDI registries. Discovery agencies take service requestors’ queries and then search appropriate Web services to suit the specific requirements in the queries. In the current practices, a UDDI entry is optional to Web services in that the service provider can also send the service description directly to the service requestor. However, this usually occurs only after two business partners have agreed on terms of doing e-business over the Internet. For this reason, in the following we focus on the more general case in which a discovery agency acts as service locators that help to find appropriate Web services for requestors.

The interaction between the service providers and the discovery agency is called a **publish** operation (see Figure 8). Next, Web services requestors **find** the appropriate Web services by querying the discovery agencies. In addition to a list of Web services, the discovery agencies return some value-added information, such as performance evaluations and predictions. This process is always called **matchmaking** (Zhang & Zhang, 2002). Once a suitable Web service has been selected, the Web services requestor gets the correspondent WSDL document and tries to **bind** with the Web service via SOAP.

In this scenario, there may be several privacy concerns (see Figure 8) that we classify according to the publish-find-bind model. Let us first consider the publish operation. In this case, the service provider may not want the discovery agency to access some of its personal information. For example, the Web services providers may have to provide some registration information to the discovery agencies. The registration information may...
contain some sensitive and confidential information for handling the business transactions, such as the trade-off model between quality and cost of service, which must be provided to the Web service requestor. Furthermore, the service provider may be an individual; it may not like to release identifiable information to unauthorized or unaffiliated parties, such as mailing address, phone number, and social security number (SSN).

Further, there also exist privacy concerns in the find operation to the discovery agencies. For example, the Web services requestors may want the discovery agencies to protect their privacy such as their identities. Additionally, the service requestor may not want the discovery agencies to release the details of their queries or even the patterns to any unauthorized or unaffiliated party. It is because the Web services requestors may have concerns about the discovery agencies to release the information to competitors or for other purposes such as marketing promotions. Finally, let us consider the bind operation. A Web service requestor may want to validate the privacy policies of business entities and services based on their privacy preferences (W3C, 2002c), before binding to the Web service. This means that the Web services requestors may only bind to those Web services whose privacy policies are matched with their privacy constraints or preferences. From another point of view, the privacy policies defined in UDDI registries for specific business entities and services must be consistent with the privacy policies defined in the service descriptions (WSDL documents). Recently, the W3C P3P Beyond HTTP task force (W3C, 2003b) recommended that associating a privacy policy with the UDDI entries is one of the technical approaches to tackle this concern. However, when optional associations are provided, the Web services providers must ensure that multiple associations do not conflict with each other in different UDDI entries.

In addition, discovery agencies may delegate some tasks to other services. There also exist privacy concerns in the delegation and propagation, in that the Web services...
providers and requestors not only concern how the discovery agencies protect their confidential and sensitive information but also whether the discovery agencies may delegate and propagate their information to other third parties without getting their consents.

Based on all the scenarios just discussed, a first requirement is that the discovery agencies should have their own privacy policies that govern the use of collected data, with the following two properties (W3C, 2003c):

- **Identifying purposes**: All the information collected from Web services providers and requestors will only be used for performing publish and find operations respectively.

- **Limiting use, disclosure, and retention**: Providers’ and requestors’ information must not be used or disclosed for purposes other than performing the publish and find operations respectively for which it was collected, except with the consent of the subject or as required by law. Web services providers’ and requestors’ information must be retained only as long as necessary for the fulfillment of performing publish and find operations, respectively.

Then, we need to devise suitable mechanisms for privacy enforcement. Some of them are described in Section viii.

### Strategies for Privacy Enforcement in Web Services Discovery Agencies

With respect to the publish and find operations, the WSA can be considered as a third-party architecture, in that the owner of the information (i.e., Web service providers) is distinct from the entities (i.e., the discovery agencies) responsible for managing information descriptions and for answering queries. In a third-party architecture, it is not always possible to adopt the traditional techniques developed for database protection (i.e., those relying on the existence of a trusted reference monitor), since they require the presence of a trusted party implementing the access control mechanism. Thus, we need to devise strategies for ensuring privacy in discovery agencies, which do not always rely on the availability of a trusted third party. Therefore, it is not possible to devise a single solution for privacy enforcement that fits in all the environments, since the right solution depends on many factors, such as the trust on the discovery agency, the sensitivity of the data, and the trade-off between efficiency and the degree of privacy assurance. Therefore, in the following, we describe three different kinds of solutions for privacy enforcement, whose applicability depend on the above-mentioned characteristics (Carminati, Ferrari, & Hung, 2005).
Access Control-Based Solution

The first kind of proposed solution relies on the presence of a trusted party inside the WSA, in charge of managing an access control mechanism and specifying access control policies (see Background on Security Mechanisms). Access control mechanisms regulate the access to UDDI registries through a set of access control policies, ensuring privacy for both Web service requestors and providers. Such trusted party can be either the discovery agency or a third party. Exploiting an access control-based solution implies that when a Web services requestor submits a query to the discovery agency, the access control mechanism filters the query answer according to the specified access control policies, and possibly prunes some portions of the answer if the requesting subject does not have proper authorizations for it. A key component of this solution (see Figure 9, solution 1) is the availability of an access control model according to which access control policies can be specified. An access control policy states which Web services consumers/providers can access which UDDI entries (or portions of them) and under which access mode and conditions. To this purpose, several access control models have been proposed (see Background on Security Mechanisms). Also, some XML-based languages for encoding authorization rules are today available (OASIS, 2003b; Bertino, Carminati, & Ferrari, 2001). The choice of the right access control model and language mainly depends on the sensitivity of the information in UDDI registries and the kinds of privacy constraints we would like to enforce. Therefore, in the following, we use an abstract notation for authorization rules, which does not make any assumption on the policy language and the underlying model. More precisely, an authorization rule is

Figure 9. Three solutions for enforcing privacy (Carminati, Ferrari, & Hung, 2005)
represented as a tuple: \((\text{subj\_spec}, \text{obj\_spec}, \text{acc\_mode}, \text{constraints})\), where \text{subj\_spec} denotes the subjects to which the rule applies (e.g., user IDs, roles, conditions on user credentials); \text{obj\_spec} denotes the protection objects to which the rule applies (e.g., a whole UDDI registry, portions of it, registration information); \text{acc\_mode} is the access mode granted by the rule; and \text{constraints} are the conditions under which the rule applies.

With this solution, it is, thus, possible to address some of the privacy issues presented in Section vi, by simply stating the right access control policies. For instance, consider the privacy concerns related to the publish operation to the discovery agencies, that is, the need of ensuring the privacy of the registration information submitted by Web services providers to UDDI registries. According to the access control-based solution, to avoid the discovery agency releasing this information to competitors, the trusted party specifies an access control policy stating that the registration information of a Web service must be made available only to the Web service provider that has published it.

Another interesting use of the access control-based solution is the possibility of enforcing Web services requestors’ privacy preferences. Indeed, as remarked in Section vi, a user may want that the discovery agency returns him/her only those services validating his/her privacy preferences. This can be easily obtained if the third party defines an access control policy, which allows a Web services requestor to access an UDDI entry only if its privacy preferences are validated by the corresponding Web service privacy practices.

**Example 1**

The following are examples of authorization rules enforcing the above-mentioned privacy requirements (for simplicity, we assume that all authorization rules refer to the same discovery agency):

\[ R_1 = (\text{service\_req1, business\_entity, read, match\_privacy\_preferences}) \]

\[ R_2 = (\text{service\_provider, registration\_info, registration\_info.id = service\_provider.id}) \]

The first authorization rule authorizes the service requestor \text{service\_req1} to see the business entities only of those services whose privacy policies match its privacy preferences. By contrast, the second rule makes registration information available only to the service providers to which they refer.

Note that the overhead required to implement the access-control-based solution is similar to the one we have in conventional database management systems (DBMSs) for enforcing access control. Similar to the proposed solution, in a standard DBMS each query is intercepted by the reference monitor, which verifies whether the access request can be authorized or not, according to the specified access control rules. In relational DBMSs access rules are stored into system catalogues (i.e., relational tables). Therefore, access control is very efficient since it can be performed by issuing few SQL queries on the system catalogues.
Cryptographic-Based Solution

According to the access control-based solution, data contained in the UDDI registry are published in clear text; therefore, this solution imposes a certain degree of trust also in the discovery agency that it does not release unauthorized information to other parties. To relax this assumption, we propose an alternative solution that always relies on a trusted party but that does not require discovery agencies to be trusted. To prevent discovery agencies from maliciously using the data they manage, we insert an additional module, with the goal of making such information unusable by discovery agencies. More precisely, we propose a solution that is similar to that proposed in Carminati, Ferrari, and Hung (2005). This solution relies on an encryption module (see Figure 9, solution 2), which encrypts different portions of the same UDDI entry with different encryption keys, according to the specified access control policies, and then publishes the encrypted copy of the entry to the UDDI registry. When a Web services provider publishes its services descriptions, the access control module marks such data with the applicable access control policies, and then the encryption module encrypts it with one or more keys, depending on the result of the marking. Finally, the encrypted copy of the UDDI entry is submitted to the UDDI registry.

In such a scenario, one can assume that the trusted party covers the task of key management by supplying the right keys to the right Web services requestors, according to the specified access control policies. When a Web services requestor needs to perform a query, it submits an encrypted query, that is, a query stating the conditions of the search in an encrypted form. This has the further benefit of avoiding that the discovery agency can trace the requestor queries. Clearly, the discovery agency must be able to evaluate queries over encrypted data without having the decryption keys. To this purpose, a method similar to the one proposed in Carminati, Ferrari, and Hung (2005) can be adopted.

Example 2

Suppose that the service provider MyBank does not want its competitors to access the details of the new home banking service it offers to its clients, whereas such details can be seen by all the other customers accessing the UDDI registry. According to the cryptographic-based solution, the UDDI entry corresponding to MyBank is encrypted with two different keys, $k_1$ and $k_2$, where: one, say $k_1$, is used to encrypt the business service element corresponding to the new home banking service, whereas the other is used to encrypt all the remaining portions of the business entity element associated with the considered service provider. Then, MyBank's competitors will receive only $k_2$, whereas all the other requestors will receive both $k_1$ and $k_2$.

The performance overhead implied by the use of a cryptographic-based solution is mainly related to two factors: encryption generation and management, and the overhead implied by querying encrypted data. Let us first consider encryption generation and management. Clearly, the cost of encryption generation depends on the number of encryption keys used to generate the encryption, which strictly depends on the number of specified authorization rules, and the size of the ciphered data. However, encryption...
generation is done once, when the UDDI entry is submitted to the discovery agency, and therefore such cost does not impact the overall performance very much. By contrast, the cost of update management should be carefully considered. Indeed, each time an authorization rule, or a portion of an UDDI entry is modified, this may require the update of the entry encryption. To limit the overhead implied by such operation, an incremental approach can be used (similar to the one presented in Carminati and Ferrari [2003]), which modifies only those portions of the encryption that are really affected by the update, without the need of regenerating the whole encryption from scratch each time an update occurs. As far as query processing is concerned, the development of efficient techniques for querying encrypted data is still an open research issue. However, some work has been done both in the context of relational (Hacigumus, Iyer, Li, & Mehrotra, 2002) and XML data (Carminati, Ferrari, & Hung, 2005), which can be applied also to UDDI registries. Finally, it is important to note that one of the main drawbacks of any encryption-based scheme is that it requires costly key management procedures (e.g., safe key storage, key recovery, and key delivery procedures), whose cost depends on the number of keys to be managed. Therefore, one of the key issues is that of devising a key generation method able to minimize the number of encryption keys that need to be managed. By applying a naïve solution according to which a different key is associated with each different set of authorization rules applied to a portion of an UDDI entry, such number is, in the worst case, equal to \(2^{N_{authrules}}\), where \(N_{authrules}\) is the number of specified authorization rules. However, more sophisticated key assignment schemes can be devised (such as the one proposed in Bertino, Carminati, and Ferrari [2002]), which greatly reduces the number of keys that need to be managed. More precisely, according to the approach proposed in Bertino, Carminati, and Ferrari (2002), the number of keys to be generated is linear in the number of specified authorization rules.

**Hash-Based Solution**

Both the solutions discussed so far are suitable for a scenario where there exists a trusted third party managing the access control policies and the encryption keys. However, there can be cases in which it is not possible to rely on this assumption. For this reason, in Carminati, Ferrari, and Hung (2005) has been proposed an additional approach (Figure 9, solution 3), which exploits hashing techniques and does not rely on a trusted third party. Such approach provides a solution to privacy concerns of Web services requestors that do not want to release their query’s details or even pattern to any unauthorized or unaffiliated party. According to this approach, the Web services providers publish hashed service descriptions in an untrusted discovery agency. More precisely, the published version contains all the information regarding how to contact the Web services providers as clear text, whereas all the other information is hashed using a standard hash function. Thus, when a Web services requestor looks for a service with certain properties, it generates a query specifying all the conditions on the properties as hashed values, and then it submits it to the untrusted discovery agency, which cannot infer the searching criteria. The discovery agency is able to perform the hashed query on the hashed description and to return to the service requestor the information for contacting the Web services that match its requirements, since this information is in clear
Having this information, the Web services requestor is then able to contact the Web services provider for further interactions.

The overhead implied by this solution is related to the generation of the hashed entries and answering a query containing hash values. The implied overhead is, however, less than that implied by cryptographic-based solutions in that the hashing of the UDDI entry and the query generation are not driven by authorization rules. As such, the cost of such operations mainly depends on the size of the input data and the selected hash function. Moreover, this approach does not have any overhead due to key management.

**Proposed Solutions for the Five UDDI Scenarios**

In this section, we show how the solutions presented so far can be effectively applied to discovery agencies. The WSA (W3C, 2002b) defines five major types of UDDI registries, described in what follows. For each of them, we discuss the applicable solutions.

1. **Internal Enterprise Application UDDI Registry**

   This UDDI registry is for companies’ internal use. Therefore, all the entities accessing the UDDI registries (either as publishers or requestors) belong to the same organization in which the UDDI resides, and the UDDI registry is placed behind the firewall. In such a scenario characterized by the fact that all the Web services are well known and trusted within the organization, there is no need to apply cryptographic or hash-based solutions. By contrast, the access control-based solution is more appropriate, where the task of specifying access control policies and managing the access control module can be carried on by the organization itself. Applying the access control-based solution to the Internal UDDI scenario has a further benefit, in that the access control mechanism can be also used to enforce workflow rules within the organization, which usually express precedence relationships among the execution of Web services. This feature can be achieved by defining proper access control rules that limit the access of a Web services requestor to the UDDI entries of only those Web services that the requestor can access according to the business rules in place at the organization.

2. **Portal UDDI Registry**

   A portal UDDI registry is preferred each time there is the need for a distinction between the services offered for external partners and those for internal use. In this scenario, the UDDI registry is located in the service provider’s environment, outside the firewall or in a demilitarized zone (DMZ) between firewalls. In this case, the UDDI manages Web services descriptions all belonging to the same organization, but it is queried also by external partners. Even in this case, an access control-based solution can be used; however, the underlying access control model must support additional ways of qualifying the subjects to which a policy applies, wrt traditional identity-based mechanisms,
since the identity of external requestors cannot always be easily verified. A possible solution is the use of an access control model based on subject credentials (Jones, Ching, & Winslett, 1995). According to such models, subjects to which an access control policy applies are determined by exploiting the notion of credential, which represents in this context, a set of Web services requestor properties to be used for access control purposes. Moreover, in this scenario, the UDDI registry could not be located behind the firewall, and thus could be untrusted. In this case, a possible solution is to apply the cryptographic-based approach.

3. Partner Catalog UDDI Registry

This type of UDDI registry publishes Web services descriptions to be used by a particular company. A partner catalog UDDI registry acts like a private UDDI registry that sits behind the firewall. This kind of private UDDI registry contains only approved, tested, and valid Web services descriptions from legitimate business partners. Therefore, all the considerations made for the internal UDDI registries are still valid for the partner catalogue UDDI registries.

4. E-Marketplace UDDI Registry

E-Marketplace UDDI registries are used to publish service descriptions related to the Web services for which a service provider intends to compete for requestors’ business. These kinds of UDDI registries are usually hosted by an industrial consortium, with the goal of managing the descriptions of the Web services providers intending to integrate with other providers for requestor’s business purposes. In this scenario, applying the access control-based solution implies that the party hosting the UDDI registry must specify the access control policies and manage the access control module. Even in this case, a credential-based paradigm is the most appropriate one for performing access control. Moreover, similar to the Portal UDDI scenario, the UDDI registry could be untrusted. In that case, the cryptographic-based solution can be adopted.

5. UDDI Business Registry

Web services may also wish to publish to the UDDI Business Registry in some other public registries that may be discovered by new potential business partners or service users. This scenario is characterized by the absence of a trusted third party or a trusted UDDI registry. Therefore, only the hash-based solution is applicable.

Table 1 summarizes what we have discussed so far, by showing the applicability of each of the proposed solutions in all the considered UDDI scenarios. From Table 1, it is clear that access control-based solution can be applied only in those scenarios where it is possible to determine a trusted third party. In addition, when the UDDI is not behind the firewall, it is necessary to apply together with the access control-based solution also the cryptographic one.
Future Trends

One of the potential users for applying such a privacy framework is the healthcare industry (also shown in Figure 2). With the increase in digitalizing health information such as Electronic Medical Records (EMR), one can imagine that the demand for privacy enhancing technologies for healthcare applications, especially based on Web services, is ever increasing. In the context of Web services, the traditional view of access control model for healthcare applications should be extended with an enterprise wide privacy policy for the management and enforcement of individual privacy preferences (Powers, Ashley, & Schunter, 2002). For illustration, an access control model must be extended to fit in the privacy rules in different countries such as HIPAA. According to the U.S. Department of Health and Human Services (HSS), HIPAA is a set of rules to be followed by health plans, doctors, hospitals, and other healthcare providers in the U.S. (HSS, 2004b). HIPAA privacy rules create national standards to protect individuals’ health information. It is therefore necessary to be standardized in Web services. Therefore, a conceptual layered architecture is needed for facilitating the design and implementation of privacy act-compliant Web services-based applications.

Conclusions

In the past few years, Web services privacy issues are attracting more and more attention from the industry and research community. While the number of Web services-based business applications is increasing, one can imagine that the demands for privacy-enhancing technologies for Web services will also be increased in the future. This chapter provided an overview of privacy issues in WSA, surveyed related technologies, and proposed solutions for some of these issues. In particular, we have presented a privacy enforcement framework, compliant with the privacy requirements defined in the “Web Services Architecture Requirements” document. Furthermore, we have proposed a suite of strategies for privacy enforcement in Web service discovery agencies.
References


Copyright © 2006, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.


