Exploring Privacy Issues in Web Services Discovery Agencies

The increasing discussions concerning Web services privacy often neglect a key building block of the Web services architecture: discovery agencies. This overview of discovery agency privacy issues highlights the various challenges and proposes different technical approaches for addressing them.

Barbara Carminati and Elena Ferrari
University of Insubria at Como, Italy

Patrick C.K. Hung
University of Ontario Institute of Technology (UOIT), Canada

Web services let individuals and organizations do business over the Internet using standardized protocols to facilitate application-to-application interaction. This offers many benefits, including platform and vendor independence, faster time to production, and convergence of disparate business functionalities. However, Web services also raise significant privacy concerns over the confidentiality of business information.

Discussions of such concerns are increasing in the industry and research communities. Because information privacy is a key issue, these discussions often focus on Web services' privacy policies. Such policies express clear and concise goals for data protection mechanisms, including what the Web services requestor expects Web services to enforce. To enable privacy protection for Web services consumers across multiple domains and services, the World Wide Web Consortium's working draft *Web Services Architecture (WSA) Requirements* has defined specific privacy requirements for Web services.[1] However, these WSA requirements don't cover all the privacy issues that might arise in a real scenario. In particular, none of the requirements address privacy concerns related to discovery agencies.

Discovery agencies manage the registries that contain Web services descriptions, thereby helping service requestors find appropriate services. As such, discovery agencies are a primary building block of the WSA and have particular privacy challenges. Here, we discuss such privacy issues and propose different technical approaches to tackle privacy concerns relating to publishing service descriptions at different types of registries. Because privacy is a major requirement in any information system, this topic is relevant not only to those working with the Web services architecture, but to IT managers and security administrators as well.

Web services overview

There are three major entities in the Web services model:

- A *provider* is the person or organization that provides an appropriate Web service for a particular business purpose.
- A *requestor* is a person or organization that seeks to use a provider's Web service to meet business requirements.
- A *broker*, or discovery agency, acts as a matchmaker between the Web services provider and requestor.

The publish-find-bind model

Figure 1 shows how the Web services' entities interact using a publish-find-bind model. In the *publish* phase, the Web services provider uses the Web Services Description Language[2] to describe its service's technical details. A WSDL document describes the Web service's interface, such as which operations the Web service supports, which protocols to use, and how to pack the exchanged data. Eventually, this WSDL document will serve as a sort of contract between the Web service's provider and requestor. The provider publishes the WSDL document to a Web services broker via universal description, discovery, and integration registries.[3]

UDDI is like a "yellow pages" of WSDL documents. In the *find* phase, UDDI provides a standard means for organizations to describe their businesses and services and

publish them so requestors can discover them online. In this scenario, the Web services broker serves as a discovery agency—much like the Google and Yahoo search engines—to help requestors find Web services that match their specific requirements.

Once requestors find a Web service at the UDDI registries, they enter the *bind* phase, requesting the service's corresponding WSDL document so that they can attempt to bind with the service via a Simple Object Access Protocol[4] message. SOAP, an XML-based messaging protocol, is independent of the underlying transport protocol (HTTP, SMTP, FTP, and so on). Service requestors use SOAP messages to invoke Web services; Web services use SOAP messages to answer the requests. The Web service thus receives the input SOAP message from the requestor and generates an output SOAP message to the requestor.

Technical framework

As Figure 2 shows, Web services each have a unique Uniform Resource Identifier (URI) located at a Web server on the Internet. Services can be defined, described, and discovered using SOAP messages, which are typically HTTP binding. On the other side, the Web services clients can be any device: a computer, PDA, or even a cell phone. Different systems interact with the Web service using SOAP messages, in a manner prescribed by the service description.[5] Today, nearly all major computing companies, including Microsoft, IBM, Sun, Oracle, and Hewlett-Packard, provide Web services tools. Early Web services adopters include several industries, such as the financial sector, in which diverse trading partners work closely together over the Internet.

There are several key Web services properties:

- *Loosely coupled*. Web services can run independently of each other on entirely different implementation platforms and runtime environments.
- *Encapsulated*. The only visible part of a Web service is the public interface, such as WSDL and SOAP.
- *Standard protocols and data formats*. Interfaces are based on a set of standards, such as XML, UDDI, WSDL, and SOAP.
- *Invoked over an intranet or the Internet*. Web services can be executed within or outside a firewall.
- *Components*. Web services composition can enable business-to-business transactions or connect separate enterprise systems, such as those related to workflow.
- *Ontology*. All interacting entities must understand the functionality behind the data value computations.
- *Business-oriented*. Web services are not end-user software.

Privacy technologies

Strategies for ensuring privacy issues have been actively investigated both in the database and Web environments.

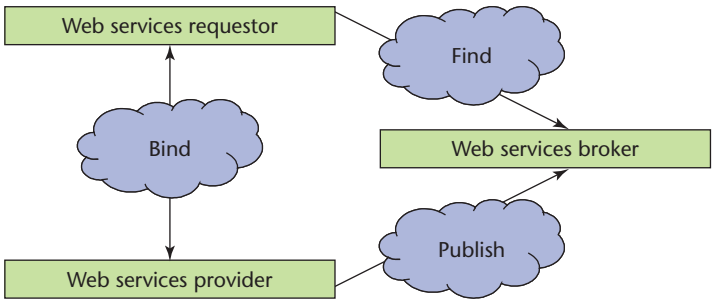

Figure 1. The publish-find-bind model. Web services providers publish their services through brokers, who act as matchmakers with requestors looking for services that meet specific business requirements.

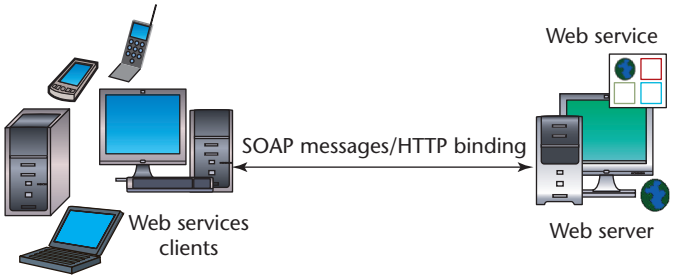

Figure 2. Web services' technical framework. An Internet Web server hosts each Web service's unique Uniform Resource Identifier. SOAP messages, which are typically HTTP binding, can be used to define, describe, and discover services.

Database technologies

The conventional database community generally interprets privacy as the confidentiality of the user's personal information. Access to database information is typically achieved through an *access control mechanism*, a software module (called the reference monitor) that regulates data accesses using access control policies. These policies are enforced through a set of authorizations: the subject identified by `sbj-id` can access the object identified by `obj-id` under the specified `access mode`. Thus, by using the appropriate languages (such as SQL for relational databases) the database administrator can specify authorizations to enforce various privacy policies.

Clearly, this scheme assumes that the reference monitor's code is trusted; barring this, there's no guarantee that the mechanism can meet policy requirements. The access control mechanism must therefore be hosted in a trusted component of the architecture. With the WSA, however, it's not always possible to ensure such a trusted party for UDDI registry management. We therefore must extend the conventional database solutions when we move to Web services.

Privacy and the Web

Obviously, researchers have been investigating privacy technologies for the Web environment for some time. One example is the efforts of the World Wide Web Consortium's Platform for Privacy Preferences (P3P) working group.[6] P3P user agents automatically inform users of a site's privacy practices and automate decision-making based on those practices. P3P also provides a language, P3P Preference Exchange Language 1.0 (APPEL1.0),[7] to express the user's preferences for making automated or semi-automated decisions regarding the acceptability of machine-readable privacy policies from P3P-enabled Web sites. Although not originally designed for tackling Web services privacy issues, the P3P framework can serve as a fundamental model for tackling privacy concerns.

Within the Web services industry, WS-Privacy[8] is one possible solution for defining a subject's privacy preferences and an organization's privacy practice statements. The WS-Privacy idea was announced a few years ago, but at this point, the WS-Privacy specification has yet to be released. The Enterprise Privacy Authorization Language technical specification[9] formalizes privacy authorizations for actual enforcement for in-house and business-to-business privacy control. EPAL focuses on privacy authorization by abstracting data models and user authentication from all deployment details. However, like P3P, the EPAL framework doesn't consider privacy enforcement specifically within a WSA context.

Privacy requirements for Web services

The W3C working draft *Web Services Architecture (WSA) Requirements*[1] specifies five privacy requirements (AC020) for enabling privacy protection for Web services consumers across multiple domains and services:[1]

- *AR020.1*: The WSA must enable the expression of a Web service's privacy policy statements.
- *AR020.2*: Advertised Web service privacy policies must be expressed in P3P.[6]
- *AR020.3*: The WSA must give consumers access to a Web service's advertised privacy policy statement.
- *AR020.5*: The WSA must enable privacy policy delegation and propagation.
- *AR020.6*: Web services must be allowed to support interactions with anonymous parties.

The W3C P3P Beyond HTTP task force is investigating most of these requirements. The task force is also working to identify the requirements for adopting P3P into protocols and applications other than HTTP, such as XML applications, SOAP, and Web services. (A working draft of the task force report is available for public comment at www.w3.org/P3P/2003/p3p-beyond-http/Overview.html).

Obviously, these WSA requirements don't cover all the privacy issues that might arise in a real scenario. In particular, none of the AC020 requirements show any privacy concern regarding discovery agencies.

Privacy and Web services discovery agencies

In the Web service publish-find-bind model, discovery agencies support the description and discovery of

- businesses, organizations, and other Web services providers;
- their available Web services; and
- the technical interfaces to those Web services.[3]

Discovery agencies provide a searchable set of Web service descriptions in centralized or distributed UDDI registries. The agencies take service requestors' queries and search appropriate Web services to suit the specific query requirements, often providing requestors with value-added information such as performance evaluations and predictions.

Basic operations

As Figure 3 shows, the interaction between service providers and the discovery agency is called a *publish operation*. The simplest case is a direct publish, in which the Web services provider sends the service description directly to the service requestor through an email attachment, FTP site, or even a CD ROM. Direct publish usually occurs after two businesses have agreed on the terms of doing business over the Internet. In this case, the Web services requestor maintains a local copy of the service description.

Next, Web services requestors find the appropriate Web services to meet their specific requirements such as cost, presence on an approved-partners list, binding support, historical performance, quality-of-service classifications, and proximity. The Web services requestors can retrieve a Web services description directly from Web services providers or they can query discovery agencies. Once requestors locate a service, they can

- retrieve the service's interface descriptions at design time for application development, or
- retrieve the service's binding and location description for invocation at runtime.

Example discovery agency

To give a sense of how discovery agencies operate, we'll use a travel reservations example. Let's say the International Air Transport Association acts as a discovery agency for its members, which include airlines, travel agents, freight forwarders, and industry suppliers. Every IATA member acts as both a Web service provider and requestor and can publish and find Web services via IATA registries. Members might use Web services to find flight schedules or book air tickets, for example.

Within the IATA, different members have various goals and privacy concerns. Airlines might form various alliances, for example, such as the Star Alliance and One World. To protect confidential business information, some airlines might prefer to provide Web services only to those IATA members in their alliance. Thus, each airline might want its Web services listings at IATA available only to certain trading partners. In addition, airlines within an alliance are also competitors, and thus might want to prevent their trading partners from knowing which Web services they've found at the IATA registries.

Privacy concerns

Figure 3 illustrates different privacy concerns related to Web services discovery agencies, organized according to the publish-find-bind model.

Bind. Web services requestors can use their own privacy preferences[7] to validate a Web service's privacy policy, binding only to those Web services whose privacy policies are consistent with their own constraints or preferences. In our IATA scenario, a travel agency might want to bind only with airlines that are its trading partners. UDDI registries' privacy policies for specific business entities and services must be consistent with privacy policies defined in the service descriptions (WSDL documents). Recently, the W3C P3P Beyond HTTP task force recommended that registries associate a privacy policy with UDDI entries. When Web services provide optional associations, however, they must ensure that the multiple associations don't conflict with each other in different UDDI entries.

Publish. Web services providers sometimes give discovery agencies sensitive registration information on how they handle business transactions. In the IATA scenario, for example, a freight forwarder agency might not want to release its trade-off model between quality and cost of service to its competitors, who could then use the agency's sensitive information to take business advantages. Also, a service provider in this case might be a self-employed travel agent who would not want identifying information—such as mailing address, phone number, or social security number—released to unauthorized or unaffiliated parties.

Find. In the find phase, Web services requestors might want discovery agencies to protect certain information—such as their identity and query details or patterns—from unauthorized or unaffiliated parties. Examples here include competitors or other companies who might pursue such information for marketing promotion or other purposes. In our IATA example, an industry supplier might want to prevent its competitors from knowing which airlines it retrieved from UDDI to prevent possible price competition.

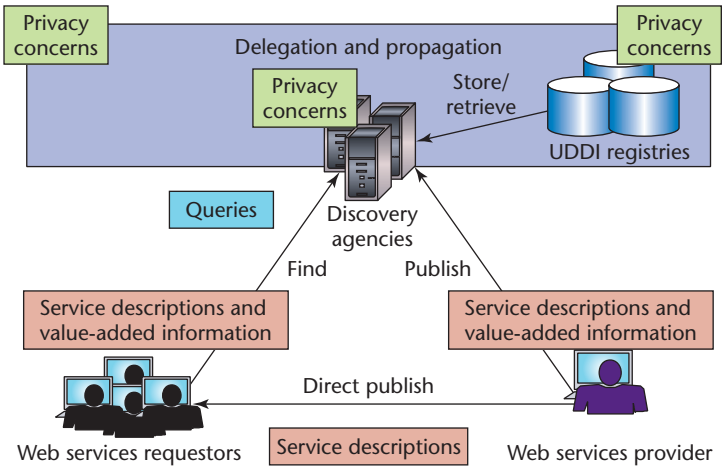

Figure 3. Privacy concerns in Web services discovery agencies. Additional privacy issues arise with universal, description, discovery, and integration (UDDI) registries, as well as with service description delegation and propagation.

Other issues

Any party in the WSA can act in one or multiple roles, which raises various privacy issues. Web services requestors might have privacy concerns when discovery agencies also act as service providers, for example, because it's doubtful that they'll properly protect requestors' information privacy.

Discovery agencies can also delegate tasks to other services. This delegation and propagation process[8] raises concerns among Web services providers and requestors as to whether the discovery agencies will protect their sensitive information. They might also be concerned that discovery agencies could delegate and propagate their sensitive information to third parties without their consent.

Privacy requirements

Given these scenarios, the first requirement is that discovery agencies have their own privacy policies that govern the use of collected data. Such a policy should include two properties:

- *Identifying purposes.* All information collected from Web services providers and requestors will be used only to perform 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, except with the consent of the subject or as required by law. Further, Web services providers' and requestors' information must be retained only as long as necessary to perform publish and find operations, respectively.

Having established the basic components of the privacy policy, we now need suitable mechanisms for privacy enforcement.

Privacy enforcement strategies

With respect to the publish and find operations, the WSA acts as a third-party architecture: information owners (Web service providers) are distinct from the entities (discovery agencies) that manage information descriptions and answer queries. In a third-party architecture, it's not always possible to adopt the traditional database security techniques because they rely on the existence of a trusted reference monitor to implement the access control mechanism. We thus need to devise strategies that don't require a trusted third party. Moreover, UDDI registries are published in clear text, so we must have a certain degree of trust in the UDDI registry itself.

To provide a comprehensive privacy solution we must therefore account for such things as the type of WSA, data sensitivity, and the trade-off between efficiency and privacy assurance. To this end, we propose three types of solutions and show how various UDDI registries can apply them.

Access-control-based solution

This solution requires that the WSA contain an access control mechanism to serve as a trusted party in charge of managing and specifying the access control policies. Such a mechanism can regulate discovery agencies' access to the UDDI registry and can thereby enforce access control policies and ensure privacy for both Web service requestors and providers.

As Figure 4 (Solution 1) shows, this solution requires a specifying policy that states which Web services consumers or providers can access which UDDI entries (or portions of them) under which access mode and conditions. Developers have defined several access control models (including mandatory, discretionary, role-based, and credential-based models) that could be exploited in the WSA context, depending on the type of UDDI registry.

An access-control-based solution works as follows. 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. It might also prune some portions of the answer to suit the requesting subject's authorization level. It's therefore possible to address some privacy issues simply by stating the right access control policies.

Consider, for example, privacy concerns related to publishing with discovery agencies. With the access-control-based solution, Web services providers could define an access control policy stating that a Web service's registration information must be available only to the Web services provider who published it.

The access-control-based solution also makes it possible to enforce the Web services requestors' privacy preferences. To do this, discovery agencies would simply define an access control policy that lets Web services requestors access a UDDI entry only if it validates the Web services requestors' privacy preferences.

The drawback of this solution is that it requires a trusted party—such as a particular organization or industry consortium—to host the access control mechanism.

Cryptographic-based solution

This solution also requires an access control mechanism managed by a trusted third party. It doesn't, however, require a trusted UDDI registry because we insert an encryption module that makes sensitive information unusable by UDDI registries. This therefore prevents them from using the data maliciously (such as tracing requestor queries).

This solution exploits the W3C standard for XML encryption[10] and is similar to the solution proposed for secure, selective XML document dissemination.[11] As Figure 4 (Solution 2) shows, the encryption module encrypts different portions of the same UDDI entry with different encryption keys according to the specified access control policies. It then publishes the encrypted copy of the entry to the UDDI registry. When a Web services provider publishes its service descriptions, the access control module marks such data with the applicable access control policies; the encryption module then 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 this scenario, we assume that the trusted party covers the key management task: supplying the right keys to the right Web services requestors according to the specified access control policies. The Web services requestor submits an encrypted query stating the search conditions. The UDDI registry requires no decryption keys because it can simply compare encrypted values. Obviously, this approach works only for queries based on equality conditions. However, the range of applicable queries could be extended using an encryption scheme that exploits polymorphic cryptography techniques to permit basic comparison operations on encrypted data.

Hash-based solution

When it's not possible or desirable to use a trusted third party, we can use hashing techniques to protect Web services requestors' query information (see Figure 4, Solution 3).

With this approach, Web services providers publish hashed service descriptions in an untrusted discovery agency. The published version contains all Web services

providers' contact information as clear text, but all other information (basically, the Web services' properties) are 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. It then submits it to the untrusted discovery agency, which cannot infer the search criteria.

The discovery agency can perform the hashed query on the hashed description and send the requestor contact information for Web services that match its requirements, since this information is in clear text. The requestor can then contact the Web services provider for further interactions.

Proposed solutions: Five UDDI scenarios

There are five major types of UDDI registries for the WSA:[5] internal enterprise application, portal, partner catalog, e-Marketplace, and UDDI Business Registry.

Internal enterprise application

In this scenario, several Web services exist within the same organization, which acts as both service requestor and provider. The UDDI registries are therefore behind the firewall.

The basic assumption here is that all entities accessing the UDDI registries (as publishers or requestors) belong to the organization hosting the UDDI. Given that all Web services are thus well known and trusted, we can assume that the organization specifies which Web services requestor can access a UDDI entry (or portions thereof) and under which access modes and conditions.

Applying the access-control-based solution to the internal UDDI scenario has a further benefit: organizations can use the access control mechanism to enforce workflow rules, which usually express precedence relationships among Web services' execution. To achieve this, organizations define proper access control rules that limit a requestor's access to only those Web services dictated by the business rules.

Portal

A portal UDDI registry is useful when organizations must distinguish between their internal services and those offered to external partners. In this scenario, the UDDI registry is located in the service provider's environment outside the firewall or in a demilitarized zone between firewalls.

In this case, the UDDI manages Web services descriptions belonging to the same organization, but it's queried only by external partners. Thus, because the Web services requestors' identities are unknown at design time, it might be impossible to exploit an identity-based mode to qualify the subjects a policy applies to (as in internal enterprise application UDDI registries). A possible solution here is to use an access control model based on subject credentials. According to such models, users subject to an access control policy are determined by exploiting the notion of *credentials*. In this context, credentials are a set of Web services requestor properties to be used for access control purposes. (These properties can be inferred by the Web services requestor description.) Thus, the organization can state conditions on the Web services requestors' properties, under which the requestors can access UDDI registry information.

Moreover, in this scenario, the UDDI registry cannot be located behind the firewall, and thus could be considered untrusted. A possible solution is to apply the encryption module to make information confidential.

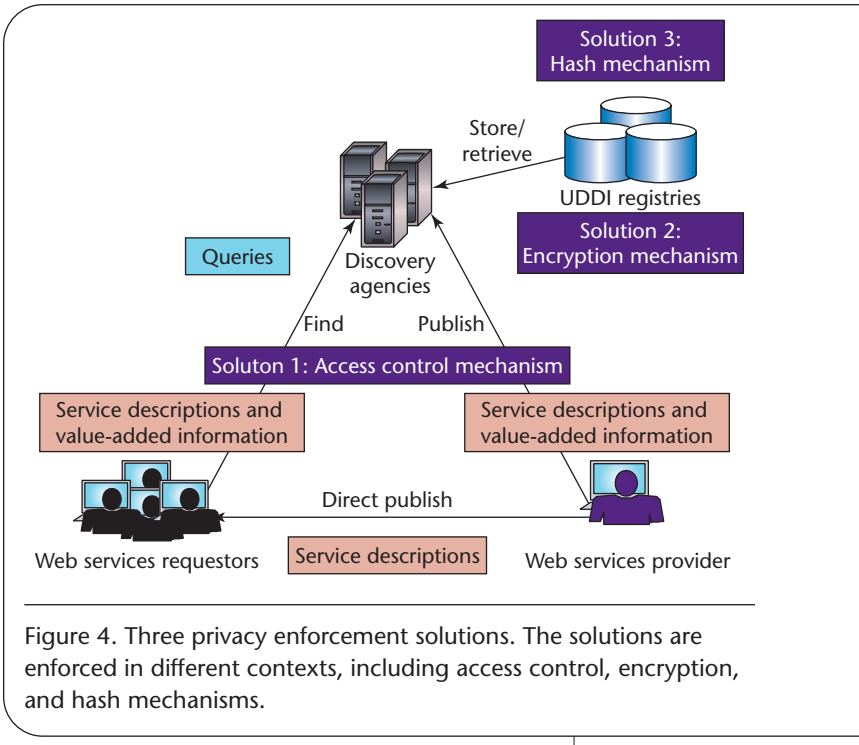

Figure 4. Three privacy enforcement solutions. The solutions are enforced in different contexts, including access control, encryption, and hash mechanisms.

Partner catalog

This UDDI registry publishes Web services descriptions for use by a particular company, providing a private UDDI registry behind the firewall. This type of registry lists only approved, tested, and valid Web services descriptions from legitimate business partners. Given this, all the considerations for the internal UDDI registries hold for this type of registry.

e-Marketplace

e-Marketplace UDDI registries publish service descriptions related to Web services for which a service provider intends to compete for requestors' business. e-Marketplace UDDI registries are typically hosted by an industry

Privacy policy enforcement

Privacy is a state or condition of having access to a person limited. Information privacy in particular relates to a person's right to determine how, when, and to what extent his or her personal information will be released to other people or organizations.

Organization-level controls

To disclose their data practices, organizations often use privacy policies that outline what information they collect from individuals (such as consumers) and what they do with that information. One of the most significant objectives of privacy policy enforcement is to protect personal identifiable information.

Threats to information privacy can come both from inside and outside an organization. Consumers and organizations often indicate that privacy of their sensitive information is their foremost concern in e-commerce activities. Among the common concerns are giving out credit-card information, how organizations use personal information, and who has access to that information behind the firewall.

National and international policies

Privacy control measures are concerned with what happens to data after individuals have released it to organizations for particular purposes. In the US, the Privacy Act of 1974 requires federal agencies to

- grant individuals access to their identifiable records,
- ensure that existing information is accurate and timely, and
- limit both unnecessary information collection and the disclosure of identifiable information to third parties.

The US thus relies mostly on self-regulation and limited legislation. Federal agencies often circumvent these constraints, however, by subscribing to commercial surrogates who collect and store the same data with no constraints. A *Washington Post* article, "Web Firms Choose Profit Over Privacy" (1 July 2003), stated that almost all companies promise not to sell consumer data. What they do not mention, the article said, was that such information is often *rented*: the list owner won't release the data to outside marketers, but will send messages to the list on the outsider's behalf.

The Europe Union Data Protection Directive[1] contains two statements that contradict this US policy. First, the directive requires that an organization inform individuals about

- why it collects personal data and how it uses the information,
- 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 only be transferred to countries outside the 15 EU signatory nations if those countries either adopt the same directive or are deemed to provide "adequate protection" for the data. The implied result is that EU organizations cannot transfer information on any EU citizen to the US due to privacy act conflicts. This creates obvious obstacles for conducting business activities between the two regions.

To ease such difficulties, the US government's voluntary "safe harbor" scheme aims to provide adequate data protection to safeguard personal data transfers from EU organizations. US companies doing business in the EU must certify to the US Department of Commerce that they'll follow the EU directive's regulations. Any violation is subject to US Federal Trade Commission prosecution for deceptive business practices.

Other issues

In a recent survey, US bank officers said that they had ongoing concerns—mostly procedural ones—about how to handle the anticipated privacy regulations of the US Gramm-Leach-Bliley Act requiring financial institutions to regularly communicate privacy policies to customers and provide adequate opportunities for "opting-out" of personal information disclosure to non-affiliated third parties.

Clearly, privacy is an important topic today and each information security system should enforce an organization's stated privacy policy. Organizations should also embed privacy-enhancing technologies in their information security mechanisms. By integrating privacy concepts into security mechanisms, organizations will be better able to earn and maintain public confidence and trust in their services.

Reference

1. G. Steinke, "Data Privacy Approaches from US and EU Perspectives," *Telematics and Informatics*, vol. 19, no. 2, 2002, pp. 193–200.

standards organization. These registries often provide services for the data they manage, such as filtering the functionalities of illegitimate entries or quality-of-service guarantees.

In this scenario, applying the access-control-based solution implies that the host organization must specify the access control policies. As with the portal UDDI registry, requestors' identities are unknown at design time, so credentials might be used for access control. Also, because the UDDI registry might be untrusted, an encryption module is needed.

UDDI Business Registry

Web services can publish to the UDDI Business Registry or other public registries where new potential business partners or service users might discover them. This scenario lacks a trusted third party and a trusted UDDI registry. Thus, only the hash-based solution is applicable.

Table 1. Applicability of the proposed solutions.

UDDI REGISTRY SCENARIO	ACCESS CONTROL	ENCRYPTION	HASHING TECHNIQUES
Internal enterprise application	Required	Applicable	Applicable
Portal	Required	Required	Applicable
Partner catalog	Required	Applicable	Applicable
e-Marketplace	Required	Required	Applicable
UDDI Business Registry	Not applicable	Not applicable	Required

Discussion

Table 1 summarizes the proposed solutions and the applicability of each in the UDDI scenarios. The key consideration with each of these solutions is the presence or absence of a trusted third party within the WSA, regardless of whether the discovery agency itself is trusted. In the first four UDDI scenarios, we can assume that the organization or the industry consortium acts as the trusted third party. Moreover, only in the first and third scenarios can we assume that the UDDI registries are trusted because they're located behind the firewall.

Over the past few years, the number of Web-services-based e-business applications has increased. Beyond the urgency of regulatory compliance, businesses are realizing that effective privacy management is essential for earning and maintaining public confidence and trust in their applications. We can therefore expect that the demands for privacy-enhancing technologies for Web services will also increase.

The scenarios we discuss here are not intended to be a completely accurate representation of real-world requirements, nor do we necessarily provide a comprehensive solution to tackling the privacy issues. Nonetheless, we believe this article is sufficiently representative in both introducing relevant research issues and serving as a starting point for future development. □

References

1. D. Austin et al., *Web Services Architecture Requirements*, Internet draft, work in progress, Nov. 2002.
2. R. Chinnici et al., *Web Services Description Language (WSDL) Version 1.2*, Internet draft, work in progress, July 2002.
3. T. Bellwood et al., *UDDI Version 3.0, UDDI Spec Technical Committee Specification*, Organization for the Advancement of Structured Information Standards (Oasis), July 2002; uddi.org/pubs/uddi-v3.00-published-20020719.htm.
4. M. Gudgin et al., *SOAP Version 1.2 Part 1: Messaging Framework*, World Wide Web Consortium (W3C) proposed recommendation, May 2003; www.w3c.org/TR/2003/PR-soap12-part1-20030507.
5. M. Champion et al., *Web Services Architecture*, Internet draft, work in progress, Nov. 2002.
6. L. Cranor et al., *The Platform for Privacy Preferences 1.0 (P3P1.0) Specification*, World Wide Web Consortium (W3C) recommendation, Apr. 2002; www.w3.org/TR/P3P.
7. L. Cranor, M. Langheinrich, and M. Marchiori, *A P3P Preference Exchange Language 1.0 (APPEL1.0)*, Internet draft, work in progress, Apr. 2002.
8. M. Hondo, D. Melgar, and A. Nadalin, *Web Services Security: Moving up the Stack*, white paper, IBM Corp.; www-106.ibm.com/developerworks/library/ws-secroad.
9. *Enterprise Privacy Authorization Language (EPAL)*, research report, IBM, 2003; www.zurich.ibm.com/security/enterprise-privacy/epal/Specification/index.html.
10. T. Imamura, B. Dillaway, and E. Simon, *XML Encryption Syntax and Processing*, World Wide Web Consortium (W3C) proposed recommendation, Dec. 2002; www.w3.org/TR/2002/REC-xmlenc-core-20021210.
11. E. Bertino et al., "Selective and Authentic Third-Party Distribution of XML Documents," *IEEE Trans. Knowledge and Data Eng.*, vol. 16, no. 10, 2004, pp. 1263–1278.

Barbara Carminati *is an assistant professor of computer science at the University of Insubria at Como, Italy, and a lecturer at the University of Milano's Computer Science School. Her research interests include database and Web security, XML, secure information dissemination, and publishing. Carminati has a PhD in computer science from the University of Milano. Contact her at barbara.carminati@uninsubria.it.*

Elena Ferrari *is professor of database systems at the University of Insubria at Como, Italy. Her research interests include database and Web security, and temporal and multimedia databases. She is on the editorial board of the* VLDB Journal *and the* International Journal of Information Technology. *Ferrari has a PhD in computer science from the University of Milano, and is a member of the ACM and the IEEE Computer Society. Contact her at elena.ferrari@uninsubria.it.*

Patrick C.K. Hung *is an assistant professor of business and information technology at the University of Ontario Institute of Technology. His research interests include security and privacy, business process integration, and electronic negotiation and contracting. He is on the editorial board of* International Journal of Web Services Research *and* International Journal of Business Process Integration Management. *Hung has a PhD in computer science from the Hong Kong University of Science and Technology. Contact him at patrick.hung@uoit.ca.*