

Preserving Privacy in Web Services *

Abdelmounaam Rezgui, Mourad Ouzzani, Athman Bouguettaya, Brahim Medjahed

Department of Computer Science

Virginia Tech

7054 Haycock Road

Falls Church, VA 22043, USA

{rezgui,mourad,athman,brahim}@vt.edu

ABSTRACT

Web services are increasingly being adopted as a viable means to access Web-based applications. This has been enabled by the tremendous standardization effort to describe, advertise, discover, and invoke Web services. *Digital government* (DG) is a major application domain for Web services. It aims at improving government-citizen interactions using information and communication technologies. Government agencies collect, store, process, and share information about millions of citizens who have different preferences regarding their privacy. This naturally raises a number of legal and technical issues that must be addressed to preserve citizens' privacy through the control of the information flow amongst different entities (users, Web services, DBMSs). Solutions addressing this issue are still in their infancy. They consist, essentially, of enforcing privacy by law or by self-regulation. In this paper, we propose a new technical approach for preserving privacy in government Web services. Our design is based on *digital privacy credentials*, *data filters* and *mobile privacy preserving agents*. This work aims at establishing the feasibility and provable reliability of technology-based privacy preserving solutions for Web service infrastructures.

Categories and Subject Descriptors

H.2 [Database Management]: Systems, Database Administration; H.3.5 [Information Storage and Retrieval]: Online Information services—*Web-based Services*

General Terms

Management, Design, Security

Keywords

Digital Government, Privacy, Web Services, Mobile Agents

*This research is supported by the National Science Foundation under grant EIA-9983249 and by a grant from the Commonwealth Information Security Center (CISC).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WIDM'02, November 8, 2002, McLean, Virginia, USA.

Copyright 2002 ACM 1-58113-593-9/02/0011 ...\$5.00.

1. INTRODUCTION

Web technologies are driving a paradigm shift in several economic activities including business-to-customer, business-to-business, and government-to-citizen relationships. The current trend in Web technologies is to provide access to business and government applications through Web-based services (or, simply, Web services). A *Web service* is a functionality that can be programmatically accessible via the Web [26]. An impressive research is currently taking place to organize, query, monitor, compose, and optimize the discovery and delivery of Web services. A significant work has focused on developing a number of standards for Web services. Examples include standards for Web service specification (e.g., WSDL [31]), advertisement and discovery (e.g., UDDI [30]), and communication (e.g., SOAP [29]). In this work, we address the issue of *e-privacy*, i.e., the privacy guarantee that Web services offer to users. For example, consider a government Web service providing social and welfare benefits (e.g., health insurance, child support). A citizen accessing this service would typically provide sensitive private information. To preserve citizens' privacy, the Web service must support mechanisms that translate the citizen's right to conceal any personal information and decide what, when and to whom any of that information may be revealed.

We view e-privacy as a three-dimension feature of Web service infrastructures: *user privacy* (as perceived by the user of a Web service), *data privacy* (as supported at the data level), and *service privacy* (as exposed to users by a Web service). Enforcing the requirements of these three levels of privacy poses a number of challenges. For example, enforcing privacy at the service level may require a complex and dynamic handling of users' privacy requirements. Moreover, service privacy is further complicated by service *composition*. For example, if \mathcal{S} were a *composite* Web service, it might have to transparently interact with a number of other Web services to answer a citizen's request. As those services do not necessarily have a privacy policy that is compatible with the privacy policy of \mathcal{S} , the citizen's information might be revealed to parties that normally have no access rights to this information. The lack of technology-based solutions to the e-privacy problem in (simple) Web services and the *implicit* sharing of information that results from the use of composite Web services have brought to the fore legitimate concerns about privacy in Web services.

This work is part of our ongoing research in digital government. We are building a comprehensive Web-based DG infrastructure called *WebDG* [6, 7]. *WebDG* integrates a

Web Service	Function	Description	Registry	Database
<i>Women, Infant, Children (WIC)</i>	Provides high-quality nutritional care and food to needy citizens	WSDL	UDDI	Oracle
<i>Medicaid</i>	Provides health care to low-income citizens	WSDL	UDDI	Oracle
<i>Teen Outreach Pregnancy (TOP)</i>	Provides childbirth and postpartum educational support to pregnant teens	WSDL	UDDI	Oracle
<i>Temporary Assistance for Needy Families (TANF)</i>	Provides cash assistance and supportive services to low-income families	WSDL	UDDI	Informix
<i>Food Stamps (FS)</i>	Supplements low income households with food stamps	WSDL	UDDI	Informix
<i>Blind Registry (BR)</i>	Enables the registry of blind people	WSDL	UDDI	Informix
<i>Family Participation Day (FPD)</i>	Helps families of visually impaired citizens develop a realistic outlook towards blindness	E-speak	E-speak	Oracle
<i>Communication Skills (CS)</i>	Teaches communication techniques needed by a visually impaired person	E-speak	E-speak	Informix
<i>Job Placement (JP)</i>	Helps citizens find employment consistent with their disabilities	E-speak	E-speak	Informix
<i>Independent Living (IL)</i>	Maximizes the independence and integration of disabled citizens in community leadership	E-speak	E-speak	Oracle

Table 1: WebDG Services and Databases.

set of techniques that enable efficient access to government Web services. We present a new approach for preserving privacy in government Web services. The approach is based on *digital privacy credentials*, *data filters* and *mobile privacy enforcement agents*. *Privacy credentials* define the scope of access of an entity to another entity’s sensitive data. *Data filters* use digital privacy credentials to control the access of remote entities to local data. When an entity is authorized to deliver an information to another entity, *mobile privacy enforcement agents* guarantee that the remote entity does not violate the local entity’s privacy requirements. The proposed privacy preserving mechanisms are implemented within the *WebDG* system.

The paper is organized as follows. In Section 2, we provide an overview of *WebDG* and its functionalities. Section 3 presents our privacy model for Web services. In Section 4, we present the design approach and details of the proposed architecture. In Section 5, we describe the implementation of the approach within the existing *WebDG* system. Section 6 provides a discussion related to the security of the mobile agents used in our solution. In Section 7, we discuss some of the related work. We conclude the paper in Section 8.

2. WEBDG: A SYSTEM FOR GOVERNMENT WEB SERVICES

This section briefly outlines the architecture of *WebDG* (Web Digital Government)¹ [6, 7]. As a case study, we use social and welfare services within *Indiana’s Family and Social Services Administration* (FSSA). The FSSA is composed of dozens of autonomous and geographically distant departments. Each department provides several rehabilitation programs to help disadvantaged citizens. It also contains a myriad of databases that store government and citizens’ information. The *WebDG* prototype uses emerging Web service standards. These include (i) *WSDL* (*Web Service Description*

Language) [31] for describing operational features of e-government services (messages, operations, host and port numbers, etc), (ii) *UDDI* (*Universal Description, Discovery and Integration*) [30], a programmatic interface for publishing and discovering services, and (iii) *SOAP* (*Simple Object Access Protocol*) [29], a messaging framework for exchanging XML formatted data among services. The current prototype includes ten (10) Web services described using *WSDL* and *HP’s e-speak* [13]. These database-backed services are published in *UDDI* and *e-speak* registries. Government and citizens information is stored in *Informix* and *Oracle* databases. Table 1 summarizes the different services and databases implemented in *WebDG*.

WebDG provides three important features:

- **Querying e-government databases:** The large number of FSSA databases makes it difficult to query the available information space if an efficient organization is not available. To deal with this problem, we organize FSSA databases into *distributed ontologies*. Each ontology contains databases that share the same domain of interest (e.g., pregnancy, employment). The use of distributed ontologies accelerates the discovery of FSSA databases.
- **Discovering and invoking government Web services:** To satisfy citizens’ needs, FSSA case officers must manually execute several applications. These applications exist in large numbers and are distributed over different FSSA departments. Thus, locating the ones that best fit citizens’ needs is usually a tedious, frustrating, and cumbersome task. To address this problem, *WebDG* wraps applications with Web services. The use of Web services caters for the dynamic *discovery* and *invocation* of welfare programs.
- **Composing government Web services:** FSSA case officers deal with different situations that depend on the particular needs of each citizen (health, children,

¹<http://www.nvc.cs.vt.edu/~dgv>

etc). For each situation, they must: (i) determine those services that appropriately satisfy the citizen's needs, (ii) determine how to access and invoke each service, and (iii) combine the results returned by the different services. To deal with this problem, *WebDG* uses a *declarative* framework for defining *composite services*. A *composite service* aggregates multiple e-government services to provide a *value added* service [19]. The use of composite services provides one-stop social services that outsource from a variety of services located in geographically distant bureaus.

3. A PRIVACY MODEL FOR WEB SERVICES

A typical Web transaction (e.g., voting, tax payment, online shopping, hotel reservation) involves three components: *users*, *services* and *databases*. This naturally defines a privacy model with three different types of privacy: *user privacy*, *service privacy*, and *data privacy*.

3.1 User Privacy

Users of a Web service include persons (e.g., citizens and case officers), applications, and other Web services. In many cases, users interacting with a Web service are required to provide a significant amount of personal sensitive information (e.g., their social security number, credit card number, health information, and address). Users of Web services, however, may expect or require different levels of privacy according to their perception of the *information sensitivity*. For example, a user may have tighter privacy requirements regarding medical records than employment history. The user's perception of privacy also depends on the *information receiver* (i.e., who receives the information) and the *information usage* (i.e., the purposes for which the information is used) [1].

The set of privacy preferences applicable to a user's information is called *user privacy profile*. A user privacy profile is typically defined by the user but can also be uniformly set for a group of individuals. Privacy profiles are *dynamic*: users can create, view, update, or delete their privacy profiles. To provide support for resolving legal disputes over privacy violation, the underlying Web service architecture must trace all of these operations. We also define a user's *privacy credentials* as a signature that is typically appended to any request that the user submits to the Web service. They determine the *privacy scope* for the corresponding user. A privacy scope for a given user defines the information that a Web service can disclose to that user. For example, a case officer accessing a government Web service may have privacy credentials granting a privacy scope that includes information about citizens' employment, housing, etc. Privacy credentials may be assigned to users on an individual or group basis.

3.2 Service Privacy

A Web service generally has its own *privacy policy* that specifies a set of rules applicable to *all* users. Service privacy generally specifies three types of policy: *usage* policy, *storage* policy, and *disclosure* policy. The *usage* policy states the purposes for which the information collected can be used. For example, consider a government Web service *Medicaid* that provides healthcare coverage for low-income

citizens. *Medicaid* may state that the information collected from citizens will not be used for purposes other than those directly related to providing health services to citizens. The *storage* policy specifies whether and until when the information collected can be stored by the service. For example, *Medicaid* may state that the information it collects from citizens will remain stored in the underlying databases one year after they leave the welfare program. The *disclosure* policy states if and to whom the information collected from a given user can be revealed. This information may relate to individual persons or to *groups* of individuals. For example, the privacy policy of the Web service *Medicaid* may state that external users cannot access statistical information that reveals general characteristics of the beneficiaries (e.g., average income, racial background distribution, etc).

3.3 Data Privacy

A data object may be accessed by several Web services. For example, consider the US National Database for New Hires (NDNH) that contains information about over 200 millions hired employees. A record in this database can be accessed (using a government Web service) by an IRS officer to check the accuracy of an employee's tax form. It may also be accessed (using another government Web service) by an officer at a child support agency to check whether a parent is compliant with his child support obligations. This shows that different Web services may need different information from the same data object. Thus, data objects must be able to *expose* different views to different Web services. For each data object, we define a *data privacy profile* that specifies the access views that it exposes to the different Web services.

Furthermore, data objects with similar *data* privacy profiles form a *privacy cluster*. A major motivation of data clustering is that legal regulations and self-defined policies enforcing privacy are typically applicable to large segments of populations (e.g., residents of a state). A privacy cluster has one single global privacy profile. Overlapping privacy clusters may exist. For example, the administrator of a government database that contains information about citizens may partition the database into two clusters C_1 and C_2 . The information in C_1 is accessible to local, state, and federal Web services, and information in C_2 is accessible only to local and state Web services.

4. DESIGN APPROACH AND FUNCTIONAL ARCHITECTURE

We now describe our privacy preserving DG architecture (Figure 1). We present the role of each component through the description of a scenario in which a user submits a service request to a government Web service (GWS). In this paper, we focus on privacy for simple (non composite) Web services. To access GWS, a user must first obtain a *digital privacy credential* (DPC) from GWS. If granted, the DPC defines the scope of the information that can be delivered by GWS to that user. A DPC encompasses information such as the user's id, the DPC's expiration date, and the set of valid operations that the user can execute through GWS. Subsequent service requests are submitted to GWS along with the user's DPC.

When the service GWS receives a service request, it first checks that the requester has the necessary credentials to access the requested service (i.e., processing that request does

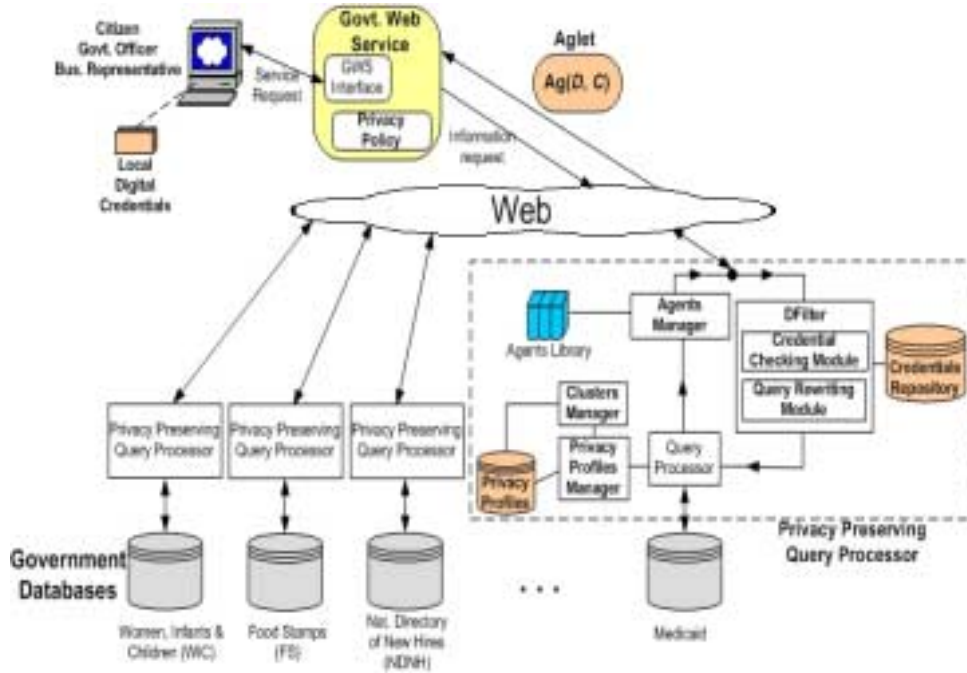


Figure 1: The functional architecture of the privacy preserving DG infrastructure.

not violate the service privacy policy). If so, GWS translates the user’s service request into an equivalent data query that is submitted to the appropriate government DBMS. When the query is received by that DBMS, it is first processed by a privacy preserving data filter (DFilter) and then submitted to the local Query Processor. The DFilter is a module that sits between a database query engine and the communication middleware. It is composed of two modules: the Credential Checking Module (CCM) and the Query Rewriting Module (QRM). The CCM uses the credential received with the query to determine whether the sending entity is authorized to access the requested information. If the credential authorizes access to only part of the requested information, the QRM *redacts* the query so that the local privacy policy of the agency and overall privacy policy of the DG infrastructure are not violated. For example, assume that a user issues a service request that is translated into the SQL query “select name, age, salary from Medicaid.enrollees”. If the request’s credential does not authorize access to enrollees’ salaries, the QRM will delete the **salary** field from the query before it is submitted to the Query Processor (QP). Figure 2 summarizes the previous steps.

When the Query Processor receives a query, it cooperates with the Privacy Profiles Manager (PPM) to generate the final result of the query. The PPM is responsible for enforcing privacy at a finer granularity than that enforced by the CCM. For example, the local CCM may decide that a given organization can have access to local information regarding a group of citizens’ health records. However, a subset of that group of citizens may explicitly request that parts of their records should not be made available to third-party entities. In this case, the local PPM will systematically instruct the query processor to discard those parts from the generated result. The PPM is a translation of the consent-based privacy model in that it implements the privacy preferences

of *individual* citizens. It maintains a repository of privacy profiles that stores individual privacy preferences. Requests made by citizens to update their privacy profiles are also handled by the Privacy Profiles Manager.

The Clusters Manager (CM) maintains data clusters. When the PPM receives a profile update request that requires a reorganization of the current configuration of clusters, it instructs the CM to compute the new configuration and appropriately update the relevant data structures.

The components described so far guarantee that privacy requirements are locally enforced. However, they cannot guarantee that the local privacy requirements are also met once the information is delivered to the querying entities. Appropriate mechanisms must be provided to enforce requirements such as preventing remote storage (at the client side) of the delivered information or forwarding of that information to third-party entities.

To address this problem, we introduce *mobile privacy preserving agents* (MPPAs). Our approach uses *mobile agents* to enforce privacy at remote sites. Mobile agents are entities consisting of code, data and control information [12]. When the query is evaluated and the result is generated, the Agents Manager (AM) generates an agent $Ag(\mathcal{D}, \mathcal{C})$ that has a data component \mathcal{D} and a code component \mathcal{C} . The data component is the result of the query. The code part is a lightweight code that is responsible for preserving the local privacy requirements on the site of the remote entity. The final result of the query is then delivered to the querying entity in the form of a two-component mobile agent $Ag(\mathcal{D}, \mathcal{C})$. This agent provides a *controlled* access interface to the query result \mathcal{D} .

5. IMPLEMENTATION

The architecture described in the previous section is implemented within the *WebDG* system [6, 7]. In the current

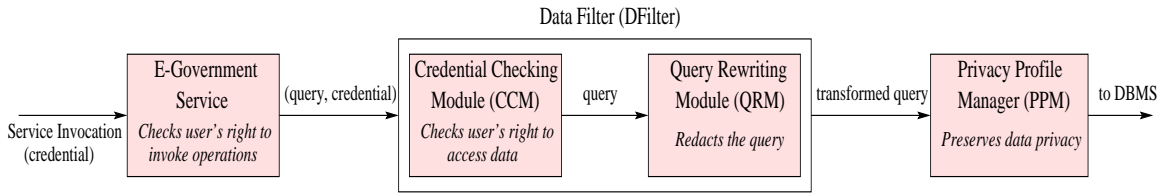


Figure 2: The data filtering mechanism.

Java-based implementation of the system, we used JDBC to access the underlying databases. A key idea in our design is the use of mobile agents to preserve privacy. We used the Aglets Software Development Kit (Version 2.0.2) as the core deployment technology. The Aglets SDK is a Java-based mobile agent system developed by IBM Tokyo Research Laboratory [14, 9]. An aglet is a mobile code (Java object) that can move from one Internet host to another along with its data and state information.

We designed the Web server (called *DGWebServer*) to run as an aglet at the server side. It receives HTTP requests destined to *WebDG* and routes them to a privacy preserving query server (called *PPQueryServer*) that also runs as an aglet. These two aglets run in the context of IBM's Tahiti aglets manager. The Tahiti aglet server is also responsible of dispatching aglets that carry query results to their recipients.

When *DGWebServer* receives a service request, it translates it into a data query and sends it to *PPQueryServer*. The query is then processed by the *DFilter* (as depicted in Figure 2). When the final query result is generated, an aglets generator (called *AgletsGenerator*) encodes it within a mobile agent (called *MobPrivAg*) that is then dispatched to the client (i.e., the host that has initiated the HTTP request). When it arrives at the client side, this agent delivers the query result to the user and permanently enforces the source privacy requirements at the client host.

6. SECURITY OF MOBILE PRIVACY PRESERVING AGENTS

The security issue is a major problem in distributed computing infrastructures based on mobile agents [23, 15, 21]. In our case, mobile privacy preserving agents must be able to access and use the resources of remote hosts whose local security policies and/or performance limitations may require a restricted access to those resources. Also, servers must be guaranteed that their agents run remotely as expected and are not compromised by tampering and reverse engineering. The challenge is then to solve the mutual security problem [24]: protecting hosts from malicious agents (also called *host security* [25]) and protecting agents against attacks from malicious hosts (also called *code security* [25]).

Most of the research in agent-based computing has focused on the first aspect of the problem [33]. Solutions include techniques such as proof-carrying code [17, 20, 4], the BSD packet filter [18], or authentication and authorization mechanisms [5]. Addressing the second aspect of the problem requires dealing with attacks (by malicious hosts) on mobile agents. An *agent executor* (i.e., host that executes a mobile agent) can [32, 15]:

- reverse engineer the agent's code
- analyze the agent's data

- arbitrarily change the agent's code and/or data
- experiment with the agent (e.g., feeding it with arbitrary data in order to observe its reactions)
- take actions that mislead the agent and result in changing its normal behavior.

This aspect of the security problem has been rated as not solvable by software means [12]. Some partial solutions have been proposed in [2, 23, 12, 32, 25]. In [2], the authors present an approach that assumes the existence of a minimally trusted third party called *secure computation service*. This independent entity performs some operations on behalf of the mobile application without learning anything about its encrypted computation. The solution presented in [23] addresses the scenario where multiple hosts on an agent's itinerary collaborate in attacking that agent. The solution's basic idea is that a proper selection of itineraries can minimize the risk of coalitions of malicious hosts being formed. The mechanism is based on the idea of *co-operating* agents, i.e., mobile agents that co-operate to establish a distributed virtual shelter inside the (untrusted) open network. Another solution was presented in [12]. It is based on the idea of creating a blackbox out of an original agent. A blackbox is an agent that performs the same work as the original agent but has a different structure. The approach presented in [32] relies on a trusted and tamper-resistant hardware device that provides the mobile agent with the means to protect itself. The solution presented in [25] involves encrypting mobile agent code using conventional symmetric key techniques to produce cipher text that is neither executable nor directly interpretable. The solution subdivides the original code of a mobile agent into several components that are encrypted/decrypted using different symmetric keys. These keys are distributed (by the agent owner) to the various host platforms using mobile agents called *Keylets*. Encryption-based approaches, however, have been shown to be limited to cases where only the code originator learns the result of the computation of the mobile code [2]. Our current investigations on the security issue also include research that has specifically addressed aglets-based distributed environments [16, 9].

7. RELATED WORK

Little attention was devoted to e-privacy in Web service standards. For instance, the two standards *WSDL* and *UDDI* provide little or no support for privacy enforcement. To the best of our knowledge, this work is one of the first in approaching privacy in Web services from a pure technical standpoint. Moreover, our approach aims at building Web service infrastructures in which privacy preserving is an integrated part similarly to other functionalities such as discovery and composition.

A notorious standardization effort that focused on specifying e-privacy is W3C's Platform for Privacy Preferences Project (P3P). The motivation behind P3P is to develop an industry standard that "enables Web sites to express their privacy practices in a standard format that can be retrieved automatically and interpreted easily by user agents"[28]. P3P assists users in understanding the privacy practices of the Web sites they visit before they release personal information. However, it provides no technical mechanisms that guarantee that those Web sites actually implement their stated privacy policy. It only provides preventive measures to preserve e-privacy. Moreover, P3P is proposed as a standard to specify the privacy of Web sites and *not* Web services; it only automates the process of checking that users' privacy will not be violated when they access applications through a P3P-enabled Web browser. In our approach, we preserve privacy when applications (i.e., Web services) are accessed by other applications (not necessarily via a Web browser).

Other techniques that also addressed e-privacy include Crowds [22], anonymizing tools (e.g., Anonymizer [3], cryptographic techniques [8], onion routing [11]), and aliases (personae) generators for Web users [10]. However, these techniques focus on hiding the *real* identity of users when they access Web-based applications. In our work, e-privacy is addressed from a multi-dimensional standpoint, i.e., users, services, and data.

8. CONCLUSION

The issue of preserving e-privacy is driving a significant amount of research. The objective is to establish the viability and reliability of technology-based solutions. In this paper, we present a DG architecture that provides a set of mechanisms that cooperate to preserve citizen's privacy. Our solution relies fundamentally on the key idea of combining *digital privacy credentials*, *data filters*, and *mobile privacy preserving agents* to enforce e-privacy. Data filters are used to control access to information stored in the different government agencies' databases. A local private information is delivered to a remote entity by a *mobile privacy preserving agent*. This agent runs at the querying host and enforces the local and global privacy policies when the associated information is accessed at the remote entity.

Although our solution's performance is yet to be evaluated, we anticipate that three major factors will determine its overall performance: the performance of the data filtering mechanism, the size of the mobile privacy preserving agents and the related cost of dispatching these agents.

Finally, on a practical note, the aglets-based implementation of our approach makes the assumption that *all* users of Web services are able to correctly set the right (local) environment supporting the execution of aglets on their hosts. To address this issue, we are investigating methods to enable a fully *transparent* aglets management at the client side.

9. REFERENCES

- [1] A. Adams. User's Perception of Privacy in Multimedia Environment. *PhD thesis, School of Psychology, University College London*, 2001.
- [2] J. Algesheimer, C. Cachin, J. Camenisch, and G. Karjoth. Cryptographic security for mobile code. Technical Report RZ 3302 (# 93348), IBM Research, 2000.
- [3] Anonymizer. <http://www.Anonymizer.com>, 2002.
- [4] A. W. Appel. Foundational proof-carrying code. In *Proc. 16th Annual IEEE Symposium on Logic in Computer Science (LICS '01)*, June 2001.
- [5] S. Berkovitz, J. D. Guttman, and V. Swarup. Authentication for mobile agents. *Lecture Notes in Computer Science*, 1419, 1998.
- [6] A. Bouguettaya, A. Elmagarmid, B. Medjahed, and M. Ouzzani. Ontology-based Support for Digital Government. In *Proc. of VLDB 2001*, pages 633–636, Roma, Italy, September 2001.
- [7] A. Bouguettaya, M. Ouzzani, B. Medjahed, and J. Cameron. Managing Government Databases. *IEEE Computer*, 34(2):56–64, February 2001.
- [8] L. F. Cranor. Electronic Voting. *ACM Crossroads Student Magazine*, January 1996.
- [9] S. Fischmeister, G. Vigna, and R. A. Kemmerer. Evaluating the security of three java-based mobile agent systems. In G. P. Picco, editor, *MA*, volume 2240 of *Lecture Notes in Computer Science*. Springer, 2001.
- [10] E. Gabber, P. B. Gibbons, D. M. Kristol, Y. Matias, and A. Mayer. Consistent, Yet Anonymous, Web Access with LPWA. *Communication of the ACM*, 42(2), February 1999.
- [11] D. Goldschlag, M. Reed, and P. Syverson. Onion Routing. *Communication of the ACM*, 42(2), February 1999.
- [12] F. Hohl. Time limited blackbox security: Protecting mobile agents from malicious hosts. In G. Vigna, editor, *Mobile Agents and Security*, volume 1419 of *Lecture Notes in Computer Science*, pages 92–113. Springer-Verlag, Berlin, 1998.
- [13] HP. E-speak developer site. <http://www.e-speak.net>, 2002.
- [14] IBM. Aglet Software Development Kit. *Online at: http://www.trl.ibm.com/aglets*, 2000.
- [15] W. A. Jansen. Countermeasures for mobile agent security. *Computer Communications: Special Issue on Advances in Research and Applications of Network Security*, November 2000.
- [16] G. Karjoth, D. Lange, and M. Oshima. A security model for aglets. *IEEE Internet Computing*, 1(4):68–77, 1997.
- [17] P. Lee and G. Necula. Research on proof-carrying code for mobile-code security. *DARPA Workshop on Foundation for Secure Mobile Code*, March 26–28 1997.
- [18] S. McCanne and V. Jacobson. The BSD packet filter: A new architecture for user-level packet capture. In *USENIX Winter*, pages 259–270, 1993.
- [19] B. Medjahed, M. Ouzzani, and A. Bouguettaya. Using web services in e-government applications. In *Proc. of the National Conference on Digital Government*

- Research*, Los Angeles, CA, USA, May 19-22, 2002.
- [20] G. C. Necula and P. Lee. Safe, untrusted agents using proof-carrying code. *Lecture Notes in Computer Science*, 1419, 1998.
- [21] A. Orso, G. Vigna, and M. Harrold. MASSA: Mobile Agents Security through Static/Dynamic Analysis. In *Proc. the ICSE Workshop on Software Engineering and Mobility*, Ontario, Canada, May 2001.
- [22] M. K. Reiter and A. D. Rubin. Anonymous Web Transactions with Crowds. *Communication of the ACM*, 42(2), February 1999.
- [23] V. Roth. Mutual protection of co-operating agents. In Vitek and Jensen [27], pages 277–287.
- [24] T. Sander and C. F. Tschudin. Protecting mobile agents against malicious hosts. *Lecture Notes in Computer Science*, 1419, 1998.
- [25] H. K. Tan and L. Moreau. Mobile Code for Key Propagation. In K. Fischer and D. Hutter, editors, *First International Workshop on Security of Mobile MultiAgent Systems (SEMAS'2001)*, Montreal, Canada, May 2001.
- [26] S. Tsur, S. Abiteboul, R. Agrawal, U. Dayal, J. Klein, and G. Weikum. Are Web Services the Next Revolution in e-Commerce? (Panel). In *Proc. of VLDB 2001*, Roma, Italy, September 2001.
- [27] J. Vitek and C. Jensen, editors. *Secure Internet Programming: Security Issues for Mobile and Distributed Objects*, volume 1603 of *Lecture Notes in Computer Science*. Springer-Verlag, New York, NY, USA, 1999.
- [28] W3C. *The Platform for Privacy Preferences 1.0 (P3P1.0) Specification*, April 2002.
- [29] W3C. *SOAP: Simple Object Access Protocol*, <http://www.w3.org/TR/soap>, 2002.
- [30] W3C. *UDDI: Universal Description, Discovery, and Integration*, <http://www.uddi.org>, 2002.
- [31] W3C. *WSDL: Web Services Description Language*, <http://www.w3.org/TR/wsdl>, 2002.
- [32] U. G. Wilhelm, S. Staamann, and L. Buttyan. Introducing trusted third parties to the mobile agent paradigm. In Vitek and Jensen [27], pages 469–489.
- [33] B. Yee. A sanctuary for mobile agents. In Vitek and Jensen [27], pages 263–275.