Access-Controlled Resource Discovery

Matthew Burnside, Dwaine Clarke, Sanjay Raman, Srinivas Devadas, and Ronald Rivest

Introduction: Resource discovery is one of the fundamental challenges that must be faced in the context of pervasive computing. The dynamic nature of pervasive networks makes it difficult for users and applications to know exactly which resources are available at any given time. Furthermore, pervasive computing environments typically handle a diverse and heterogeneous set of users and resources, including computationally-limited devcies, posing new and different security challenges. Communication channels between many disparate devices must be secure and access control must be granted to resources in order to regulate their usage. While several systems propose resource discovery solutions for dynamic environments, they do not consider how the integration of security protocols influence scalability and performance. Here we describe a resource discovery system that provides access-controlled resource discovery, using the Intentional Naming System (INS). INS [1] is a naming system that enables applications to describe *what they are looking for* not *where to find it*. The access-controlled resource discovery system is part of a larger security infrastructure based strongly on proxy-based SPKI/SDSI [2] to provide a distributed security framework for pervasive networks of devices and computers.

Approach: Our approach to integrating access control with INS is based on the assumption that the interface between a resource discovery system and security infrastructure should not be hard. It is inefficient if a user (requestor) has to repeatedly iterate through lists of resources that he is prohibited from using while he searches for the most-optimal accessible resource. One only has to consider a scenario where a user is in an environment with several resources that are inaccesible to him to see how scalability becomes a major issue when adding security to resource discovery. If the resource discovery system has no knowledge about which resources the requestor can access, it could take tremendous computational effort to find an accessible resource.

A better approach would be to give the resource discovery system knowledge about the access control lists (ACLs) that protect each of the resources and the access-control groups and capabilities of the requestor. A resource discovery system already knows about the service and performance characteristics of each resource it represents; an access control list is nothing more than an additional characteristic that defines the resource. The idea here is not to leave security entirely up to the resource discovery system, but, instead, to provide the discovery system with "hints" regarding the accessiblity of resources. Security is still enforced by an end-to-end proxy protocol (this protocol is described in our previous work [2]), but the sharing of access information enables the resource discovery system to find resources that are guaranteed to be accessible. Figure 1 shows a system level diagram of our entire security infrastructure. This summary focuses on integrating acess control into INS.

INS provides users with a layer of abstraction so that applications do not need to know the availability or exact name of the resource for which they are looking. We extend INS to provide access-controlled resource discovery in two main ways:

- **Dynamic maintenance of ACLs in INS**. We implement a real-time maintenance of access control lists (ACLs) in the INS name resolvers in order to give INS knowledge about how each resource should be protected.

- **User authorization**. We introduce a certificate-based authorization step during the resolution of an INS request in order for INS to know the identity and privileges of a requestor.

To maintain ACLs in INS, we addressed two issues: 1) how to represent ACLs in INS search trees, and 2) how to traverse INS trees and make access control decisions. As a resource's ACL is like the other attributes that define a resource, we store ACLs as attribute-value pairs in INS, just like any other searchable property. Mapping access-control entries to attribute-value pairs is advantageous because we do not change the manner in which INS stores data.

INS uses a `LOOKUP-NAME` algorithm to retrieve resource addresses for a given query. It uses an internal tree that hierarchically organizes resources based on their attributes. For example, a printer in the system can be defined by attributes such as `dpi`, `location`, and `service`. The `LOOKUP-NAME` algorithm prunes branches that don't meet the criteria specified in the query. In order to implement access checks, we want INS to eliminate potential branches that cannot be accessed by the user. To do this, we use the logical OR ($\vee$) operator to compute intermediate ACLs at each parent node. The ACLs are computed in this manner all the way up the tree (see Figure 2). If a user is not in an
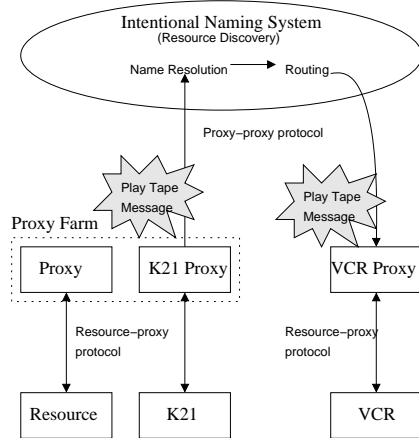
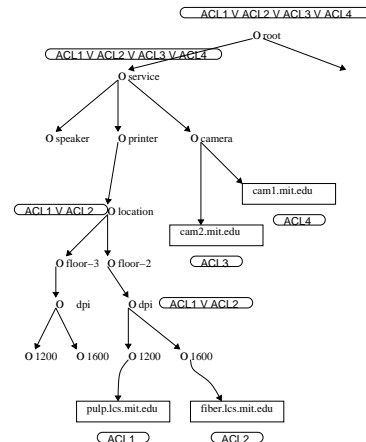Figure 1: Resource Discovery System Overview

Figure 2: Propagation of Access Control Lists

intermediate computed ACL, there is no need to continue searching down that particular branch. ACLs can be updated like any other resource property by secure messages between a resource proxy and an INS router. The K21 proxy for each user computes a (finite) transitive closure of the user's SPKI/SDSI certificates and extracts rules summarizing the user's authorizations. These rules state the access-control groups to which the user belongs and the operations that he can perform. The K21 proxy presents the user's authorization rules to INS with the user's query. INS uses the rules and the OR ($\vee$)'ed ACLs to help prune branches as it searches for the most-optimal accessible resource for the user. By using the modifications we describe, we develop a *secure and scalable* resource discovery system that is ideal for dynamic networks.

Progress: We are currently working on implementing the modifications to INS that have been proposed in this paper. There are several open issues of active research that still need to be addressed. We need to handle the authenticity of update messages that resouce proxies send to INS name routers. We also need to investigate the privacy and authenticity of requests from K21 proxies to INS name routers. While these problems appear to be straightforward, different options must be evaluated with a concern for how they impact the overall scalability and performance of the system.

Future: We plan to complete the implementation of the system we have described here and then integrate it with the existing SPKI/SDSI security system that has been developed. We also plan to quantitatively evalute the performance advantages of our system over alternative methods and compare that to a theoretical evaluation of our system that we have developed.

References:

[1] W. Adjie-Winoto, E. Schwartz, H. Balakrishnan, and J. Lilley, "The Design and Implementation of an Intentional Naming System," *Operating Systems Review*, vol. 34(5):186-301, December 1999.

[2] M. Burnside, D. Clarke, T. Mills, A. Maywah, S. Devadas, and R. Rivest, "Proxy-based security protocols in networked mobile devices," in *Proc. ACM SAC02*, March 2002.