Proxy-Based Security Protocols in Networked Mobile Resources

Matthew Burnside, Dwaine Clarke, Todd Mills, Andrew Maywah, Srinivas Devadas, Ronald Rivest

Introduction: The goals of ubiquitous and pervasive computing [1, 2] are becoming more and more feasible as the number of computing resources[1] in the world increases rapidly. However, there are still significant hurdles to overcome when integrating wearable and embedded resources into a ubiquitous computing environment. These hurdles include designing resources smart enough to collaborate with each other, increasing ease-of-use, and enabling enhanced connectivity between the different resources. Security in the system is extremely important; resources must only allow access to authorized users and must also keep the communication secure when transmitting or receiving personal or private information.

Approach: To allow our architecture to use a public-key security model on the network while keeping the resources themselves simple, we create a software proxy for each resource. All objects in the system, e.g., appliances, wearable gadgets, software agents, and users have associated trusted software proxies that either run on an embedded processor on the appliance, or on a trusted computer. In the case of the proxy running on an embedded processor on the appliance, we assume that resource-proxy communication is inherently secure.[2] If the resource has minimal computational power,[3] and communicates to its proxy through a wired or wireless network, we force the communication to adhere to a resource-proxy protocol. Proxies communicate with each other using a secure proxy-proxy protocol based on SPKI/SDSI (Simple Public Key Infrastructure / Simple Distributed Security Infrastructure). With two different protocols, we are allowed to run a computationally-inexpensive security protocol on impoverished resources, and a sophisticated protocol for authorization and communication on more powerful resources.

The resource-proxy protocol varies for different types of resources. In particular, we consider lightweight resources with low-bandwidth wireless network connections and slow CPUs, and heavyweight resources with higher-bandwidth connections and faster CPUs. We assume that heavyweight resources are capable of running proxy software locally (i.e., the proxy for a printer could run on the printer's CPU). With a local proxy, a sophisticated protocol for secure resource-proxy communication is unnecessary, assuming critical parts of the resource are tamper resistant. For lightweight resources, the proxy must run elsewhere. An example of a resource-proxy protocol for a lightweight resource is one in which the resource and its proxy share symmetric keys with which they encrypt and authenticate their communication.

For the proxy-proxy protocol, we have adopted a client-server architecture. When a particular principal, acting on behalf of a resource or user, makes a request via one proxy to a resource represented by another proxy, the first proxy acts like a client, and the second as a server. Services on the server are either public or protected by SPKI/SDSI access control lists (ACLs). To gain access to a service protected by an ACL, a client must send a signed copy of its request, and a chain of SPKI/SDSI certificates demonstrating that it is a member of a group in an entry on the ACL.[4]

The proxy-proxy protocol layers SPKI/SDSI access control over an application protocol, which in turn is layered over a key-exchange protocol. This allows us to deal with a variety of application protocols which may be implemented across wired or wireless links in a heterogeneous network.

Using the SPKI/SDSI framework, ACLs associated with resources can be created once and rarely need to be modified. User access rights are modified by issuing certificates based on group membership; rights can be revoked through a variety of mechanisms such as online checks. In addition, SPKI/SDSI features an elegant model for delegation of authority, allowing for the partitioning of responsibilities. The user maintaining an ACL on a resource could, but need not be, the same user that authorizes others to access the resource. This significantly eases the burden of system administration.

Progress: Both the resource-proxy and proxy-proxy protocols have been implemented [5], and hardware to support the resource-proxy protocol has been built [6]. We are currently testing the system for scalability and performance.

[1] A *resource* refers to any type of shared network entity, either hardware or software.

[2] For example, in a video camera, the software that controls various actuators runs on a powerful processor, and the proxy for the camera can also run on the embedded processor.

[3] This is typically the case for lightweight resources such as remote controls, active badges, etc.

[4] For examples of SPKI/SDSI ACLs and certificates, see [3] or [4].

Future: In the proxy-proxy protocol, the client proxy authenticates a resource and its attributes before making requests for its services. Designing simple, flexible, and scalable schemes for authenticating resource attributes is a topic of ongoing research (examples of a resource's attributes are its IP address, its geographical location, and its current service load).

Research Support: This research was supported by Acer Inc., Delta Electronics Inc., HP Corp., NTT Inc., Nokia Research Center, and Philips Research under the MIT Project Oxygen partnership, and by DARPA through the Office of Naval Research under contract number N66001-99-2-891702.

References:

[1] M. Dertouzos, "The Future of Computing," *Scientific American*, August 1999.

[2] G. Banavar, J. Beck, E. Gluzberg, J. Munson, J. Sussman, and D. Zukowski, "Challenges: An Application Model for Pervasive Computing," in *Proc. ACM MOBICOM*, August 2000.

[3] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen, "Simple Public Key Certificate," *The Internet Society*, July 1999, See http://world.std.com/∼cme/spki.txt.

[4] D. Clarke, "SPKI/SDSI HTTP Server / Certificate Chain Discovery in SPKI/SDSI," M.S. thesis, Massachusetts Institute of Technology, 2001.

[5] M. Burnside, D. Clarke, T. Mills, A. Maywah, S. Devadas, and R. Rivest, "Proxy-Based Security Protocols in Networked Mobile Devices," in *Proc. ACM Symposium on Applied Computing*, March 2002.

[6] T. Mills, "An Architecture and Implementation of Secure Device Communication in Oxygen," M.S. thesis, Massachusetts Institute of Technology, 2001.