# Designing and Implementing Access Control for Impromptu Collaboration

William R Claycomb and Dongwan Shin
Department of Computer Science
New Mexico Tech
Socorro, NM 87801
Email: {billc, doshin}@nmt.edu

## I. Introduction

Impromptu collaboration, often characterized as being opportunistic, spontaneous, proximity-based, and transient, is fast becoming a common way of interactions in mobile and pervasive computing. Access control for impromptu collaboration is a complex issue, with many aspects relating to the establishment, management, and enforcement of methods and policies that allow mobile devices to share resources with each other. Further, communication between mobile devices can arise spontaneously, involve the sharing of few resources between heterogeneous platforms, and only need to be maintained for a short time. Additionally, the devices often communicate with each other a single time, and have no pre-shared or *a priori* knowledge of the other device or its capabilities. Due to the nature of pervasive computing, and the possibility that shared resources could contain sensitive information, it is important to be able to secure communication between devices, while still maintaining flexibility and ease of use.

With this work, our objective is to design and implement a secure solution for providing controlled access to local resources of mobile devices. Specifically we aim to incorporates a method of demonstrative identification of mobile devices, key-based capability delegation, and two-dimensional visual barcode technology to provide a simple access control solution with low maintenance costs. The solution is particularly for one-time-only communication situations between two previously unknown mobile devices. In this abstract, we will briefly describe background topics, the approach we have taken, the work that has been completed to date, and our future plans.

## II. Background

Two important areas to understand when approaching this problem are controlling access to resources between mobile devices and the method used to establish secure communication in the first place. Several works address access control in pervasive computing environments, but few specifically address access control between two previously unintroduced devices. Many of the works focus on a method of access control known as *distributed trust*, in which some or all of the existing network devices control access to shared resources. Other methods employ the use of a central authority to bootstrap access control restrictions on both client and host devices, but do not assume that a connection to that central authority will be present at the time the client seeks to access the host. Our approach will build on this method, but with several enhancements and modifications that apply to pervasive computing environments.

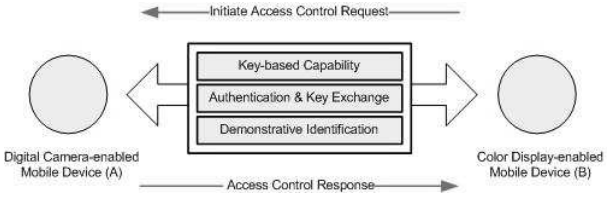


Fig. 1. Our Approach to Access Control in Impromptu Collaboration

Establishing a secure connection between mobile devices has also been previously explored and several methods have been proposed. We build on one particular approach, called *Ubicolor* [3] for that purpose.

## III. Our Approach

Our approach to designing and implementing access control for impromptu collaboration in pervasive computing consists of three distinct components, as shown in Figure 1.

- Demonstrative identification of devices
- Authentication and key exchange
- Key-based capability delegation/revocation

Each of the components can handle different questions such as how to verify the identity of a remote device, how to prevent unintended access by malicious entities, how to restrict access to specific resources, and how to revoke permissions once granted.

### A. Device Identification

Device identification concerns demonstratively identifying a remote device before access control can occur. Our approach leverages one particular method, *Ubicolor*. Ubicolor utilizes two key components commonly found in many mobile devices, specifically their colorized displays and integrated digital cameras, to identify the devices and establish a secure communication channel between them.

### B. Authentication and Key Exchange

Once the public key is verified and the binding information between the physical address and the public key of the client

device is visually tested, there is a need to determine if the remote entity really holds a corresponding private key. In addition, a short-term session key is necessary to keep the communication between the two devices confidential. The shared secret key can be generated by the host device and passed to the client device as encrypted by the client device's public key.

### C. Key-based Capability Delegation

Capability lists used for controlling access to shared resources in the host device are contructed by the attributes established in the previous two components. Specifically they include the following:

- A shared secret key
- A verified public key

With these attributes established, the component of key-based capability list concerns issuing and revoking capability credentials, which contain information identifying the remote client, the remote client's verified public key, which resources the remote client has authorization to access, and what level of access that remote client has with respect to those resources.

## IV. Design and Implementation

Designing and implementing this type of access control mechanism may not seem very technically challenging, but some considerations must be noted about design and implementation details. Mobile devices, particularly those with color displays, digital cameras, and operating systems capable of supporting Ubicolor, are more than capable of supporting the encryption and secure communication required by the access control mechanism described in this abstract.

We believe our access control system needs to contain the following components to be fully functional:

- A graphical user interface
- A connection/certificate management engine
- A connection/certificate association table
- A policy decision and enforcement engine
- A resource sharing engine
- A resource discovery engine

### A. Current Status

We currently have a prototype implementation capable of establishing a secure channel between two devices, issuing self-signed X.509 certificates, transmitting certificates between devices, and discovering remote resources based on those certificates. A screen shot of the GUI is shown in Figure 2.

While not optimized for performance yet, we have done some limited performance testing on key components of this system. In doing so, we considered two distinct parts. The first is to establish a secure communication channel. The results of this are discussed in detail in [3]. On average, a secure connection can be made between two mobile devices in about six seconds, using Ubicolor. The second component of performance measure is the time and resources necessary to establish access control restrictions for the secure connection. Based on initial testing, we are able to generate self-signed X.509v3 certificates on an mobile device in approximately 1 second[1].


Fig. 2. Selecting Resources to Share

## V. Future Plans

Our future plans includes adding actual resource sharing to the system. This will begin with simple resources, such as files and contact information, but will hopefully extend to include more complex resources, such as GPS receiver information.

Of course, performance analysis is a key component to the success of such a system. We intend to closely monitor not only the time factor involved in establishing this type of access control, but also the resources used by the mobile device in doing so. Several costly steps are involved in establishing and using both public/private key pairs and X.509 certificates, in terms of processing capacity and power consumption. We intend to identify the cost of these steps and evaluate the overall affect to the device based on expected use.

A related area we also intend to explore is finding ways to reduce the computational cost and power consumption of the overall system, while maintaining the security features our model provides. We would like to see secure connections established and access control defined with the least use of resource-intensive procedures. By doing so, we feel the model will not only be better suited for extended use in mobile devices, but also will be a candidate for secure resource sharing among less capable devices.

## References

[1] S. Berger, C. Binding, C. Hoertnagl, S. McFaddin, and A. Ranganathan. Towards pluggable discovery frameworks for mobile and pervasive ... In *Proceedings of the 2004 IEEE International Conference on Mobile Data Management, 2004*, 2004.

[2] R. Campbell, J. Al-Muhtadi, P. Naldurg, G. Sampemane, and M. D. Mickunas. Towards security and privacy for pervasive computing. In *Proceedings of International Symposium on Software Security*, Tokyo, Japan, 2002.

[3] W. R. Claycomb and D. Shin. Using a two dimensional colorized barcode solution for authentication in pervasive computing. In *Proceedings of the IEEE International Conference on Pervasive Services 2006*, Lyon, France, June 2006.

[1]Tested on an HP iPAQ rx3715 PDA, with a Samsung S3C2440 processor and 64 MB of RAM. The operating system of this device was Microsoft Windows Mobile 2003, Second Edition.