

# SECURING THE USE OF SENSITIVE DATA ON REMOTE DEVICES USING A HARDWARE-SOFTWARE ARCHITECTURE

Jeffrey Scott Dwoskin

Adviser: Ruby B. Lee

Readers: Edward Felten, Margaret Martonosi

June 2010

## **Abstract**

Many corporations, private organizations, and government agencies maintain sensitive data that must be accessed remotely by their employees using portable devices. The organizations have a responsibility to secure the data to ensure that it does not get used inappropriately or get disseminated beyond these trusted users. We have designed a computer architecture for these devices, combining new hardware and software, that allows trust to be placed in the devices even when they are not under the organization's physical control.

We have designed, implemented, and tested the Authority-mode Secret-Protection Architecture, which places roots of trust in hardware in the processor chip. It provides new hardware mechanisms based on these roots of trust to protect the execution of trusted software and to provide that software with master secrets. The software uses the master secrets to secure the sensitive data and to communicate securely over the network. The user interacts with this software, which enforces security policies while giving access to data.

The organization designates a central authority that will manage the software on the devices, set security policies, communicate with the devices, and control access to data. Our new hardware mechanisms bind together the device's on-chip roots of trust with the authority's data and trusted software, such that the authority can be assured that the security policies will always be enforced.

To show how our design can be adapted to other platforms, we provide a modified architecture for embedded devices. We additionally demonstrate how the full architecture can be integrated with trustworthy system software in a mandatory access control system.

Finally, we have built a testing framework that can help designers validate new security architectures like ours. The framework allows new architectures to be modeled in a virtualization environment, where a separate testing system has complete controllability and observability over hardware and software. It is used to test the effects of various security attacks and to assist in the development of trusted software for the new architecture. We use the framework to test the prototype hardware and software of our architecture.

## Outline

### Chapter 1: Introduction

In the introduction, we provide motivation for the challenge of securing data on portable devices. We consider how a remote authority can enforce access control policies and prevent attacks on sensitive data, even when the devices are used remotely. We discuss our threat model and a variety of usage scenarios, including medical practitioners, crisis response, and government agencies.

### Chapter 2: Related Work

We discuss related work in a number of areas. For hardware architectures, we look at secure processors and secure co-processors and how they are used for protecting data, preventing software piracy, controlling information flow, and preventing attacks on software. We then survey secure software systems, including operating systems and hypervisors, with a focus on building a trusted computing base and providing isolation and least privilege separation. Next, we consider methods used to evaluate security and test security architectures. Finally, we broadly evaluate methodologies for controlling access to data, including digital rights management, mandatory access control policies, capabilities, and dedicated embedded devices.

### Chapter 3: Authority-mode SP Architecture

In this chapter, we present a design for a new hardware-software security architecture, Authority-mode SP, for portable devices that provides remote and transient trust for sensitive data. The architecture allows a central authority to disseminate data to remote devices, which through a combination of hardware mechanisms and trusted software are able to protect the confidentiality and integrity of the data, while enforcing access control policies that determine how the data can be shared and used on the device. We develop a usage scenario for crisis response and emergency management, demonstrating remote trust for managing access to data, multiple complementary methods of revocation of data on the remote devices, the secure initialization of the devices, and the use of data by a local user.

### Chapter 4: SP on Embedded Devices

In order to more fully explore the usefulness of the architectural concepts inherent to Authority-mode SP, we have created a compact version for embedded devices, described in this chapter. In this architecture, Embedded SP, we have extended and applied the Authority-mode SP mechanisms to limited platforms, such as sensor nodes, which have more restricted hardware capabilities and distinct threat models from the personal computing devices for which SP was originally designed. We design a reduced hardware architecture that provides similar remote trust capabilities for the embedded platform, and demonstrate how it can be applied to securing key-distribution protocols in mobile ad-hoc sensor networks.

### Chapter 5: A Framework for Testing Hardware-Software Security Architectures

Next, we developed and implemented a framework for testing new hardware-software security architectures. It can verify the security properties of such an architecture during the design-phase, before real hardware can be built. The platform emulates new security mechanisms that would exist in hardware and software in the real architecture, allowing actual software applications to be designed, developed, and thoroughly tested on the platform. The framework provides an environment where the full state of the system under test is available to be actively observed and dynamically controlled, allowing attacks to be performed that test and stress the security mechanisms, while observing the effects of these attacks beyond just their success or failure.

We have implemented the framework using a virtualization platform and are using it to test the Authority-mode SP architecture. To do this, we have created an emulation module that implements many of the features of the base SP architecture as well as all of the relevant mechanisms for the Authority-mode features. Thus we are able to write trusted software for Authority-mode SP that runs on the platform and can interact with other applications. We then test a variety of interactions on the platform both with and without attacks taking place.

#### Chapter 6: SecureCore Platform

Finally, we study how to use Authority-mode SP as part of the SecureCore platform, as a means to more thoroughly evaluate the software model of our architecture and its interactions with other trusted and untrusted software. We have integrated Authority-mode SP with a secure hypervisor and trusted operating system to produce a full virtualized platform that can enforce Mandatory Access Control (MAC) policies and control the dissemination and use of sensitive data. We use this platform to implement an emergency device that supports Multi-Level Security (MLS) with data at varying security levels separated into different partitions, while still offering the remote trust and revocation capabilities of Authority-mode SP. We pay particular attention to ensuring that the combined architecture is safe from covert channels. This emergency device demonstrates a method to protect sensitive data during display to the user with trusted I/O paths, while preventing information flow off of the device in digital form. We also solve the problem of how to integrate SP's trusted software into both the application and operating system layers of the system.

#### Chapter 7: Virtualization of SP and Dual-mode SP Devices

We discuss some additional enhancements to the SP architecture with designs for virtualizing the SP architecture and for combining user-mode and authority-mode SP in a single device. We further provide a summary of all SP instructions and states that are needed to implement the combined features of all of the enhancements.

#### Chapter 8: Conclusion

This chapter provides a summary of the work in the dissertation and considers areas for future work related to protecting trusted software, expanding and improving the SP architecture, protecting system software, securing interactions with the user, and further developing the testing framework.

#### Appendix A: TSM Best Practices

This appendix provides some best practices that we have developed for writing TSMs securely using SP mechanisms.

#### Appendix B: Testing Framework Implementation Details

This appendix provides the detailed specification for the hypercall interface in the VMM that implements the Event & Attack Module API from Chapter 5.

#### Appendix C: Acronyms

This appendix provides a list of acronyms that are used in the dissertation.

## A. PUBLICATIONS RELATED TO DISSERTATION

Timothy Levin, **Jeffrey S. Dvoskin**, Ganesha Bhaskara, Thuy Nguyen, Paul Clark, Ruby B. Lee, Cynthia Irvine, Terry Benzel. "Securing the Dissemination of Emergency Response Data with an Integrated Hardware-Software Architecture," *Proceedings of the 2nd International Conference on Trusted Computing (TRUST 2009)*, pp. 133-152, Oxford, U.K., April 2009.

**Jeffrey S. Dvoskin**, Mahadevan Gomathisankaran, Ruby B. Lee. "A Framework for Testing Hardware-Software Security Architectures", *Princeton University Department of Electrical Engineering Technical Report CE-L2009-001*, Feb 04, 2009, updated June 2009. *To be re-submitted*.

**Jeffrey Dvoskin**, Dahai Xu, Jianwei Huang, Mung Chiang, and Ruby B. Lee, "Secure Key Management Architecture Against Sensor-node Fabrication Attacks." *IEEE Global Telecommunications Conference 2007 (GLOBECOM'07)*, pp. 166-171, Washington, DC, November 2007.

**Jeffrey S. Dvoskin** and Ruby B. Lee, "Hardware-rooted Trust for Secure Key Management and Transient Trust," *Proc. of the 14th ACM Conference on Computer and Communications Security (CCS 2007)*, pp. 389-400, October 2007.

## B. PUBLICATIONS PERIPHERALLY RELATED TO DISSERTATION

Dahai Xu, **Jeffrey Dvoskin**, Jianwei Huang, Tian Lan, Ruby Lee, Mung Chiang. "Key Management in Sensor Networks". Book chapter, *Theoretical Aspects of Distributed Computing in Sensor Networks*, Springer Verlag, November, 2009. *In submission*.

Dahai Xu, Jianwei Huang, **Jeffrey Dvoskin**, Mung Chiang, Ruby Lee, "On Secure Key Management in Mobile Ad Hoc Networks," 2008. *Journal paper, in submission to IEEE Transactions on Mobile Computing*.

Ruby B. Lee, Peter C. S. Kwan, John Patrick McGregor, **Jeffrey Dvoskin**, and Zhenghong Wang, "Architecture for Protecting Critical Secrets in Microprocessors," *Proceedings of the 32nd International Symposium on Computer Architecture (ISCA 2005)*, pp. 2-13, Madison, Wisconsin, June 2005.

### C. OTHER PUBLICATIONS NOT IN DISSERTATION

Xiaoxin Chen, Tal Garfinkel, E. Christopher Lewis, Pratap Subrahmanyam, Carl A. Waldspurger, Dan Boneh, **Jeffrey S. Dvoskin**, Dan R. K. Ports, "Overshadow: A Virtualization-Based Approach to Retrofitting Protection in Commodity Operating Systems," *Proc. of the Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pp. 2-13, March 2008.

Dahai Xu, Jianwei Huang, **Jeffrey Dvoskin**, Mung Chiang, and Ruby Lee, "Re-examining Probabilistic Versus Deterministic Key Management," *IEEE International Symposium on Information Theory (ISIT'07)*, pp. 2586-2590, Nice, France. 2007.

**Jeffrey Dvoskin**, Sujoy Basu, Vanish Talwar, Raj Kumar, Fred Kitson, and Ruby Lee, "Scoping Security Issues for Interactive Grids," *Proceedings of the 37th Asilomar Conference on Signals, Systems, and Computers*, pp. 367-373, November 9-12, 2003.

### D. US PATENT APPLICATIONS

Ruby B. Lee, **Jeffrey S. Dvoskin**. "Hardware Trust Anchors in SP-Enabled Processors", U.S. Patent Application – filed August 14, 2009.

### E. OTHER TECHNICAL REPORTS

Yu-Yuan Chen, **Jeffrey S. Dvoskin**, Mahadevan Gomathisankaran, Ruby B. Lee. "Making Security Validation as Easy as Performance Evaluation", *Princeton University Department of Electrical Engineering Technical Report CE-L2009-005*, November, 2009.

**Jeffrey Dvoskin**, Mahadevan Gomathisankaran, David Champagne, Ruby B. Lee, "SP Reference Manual Addendum – Secure Stacks for TSMs and Emulation of SP Interrupt Protection," *Princeton University Department of Electrical Engineering Technical Report CE-L2009-006*. August, 2009.

**Jeffrey S. Dvoskin**, Mahadevan Gomathisankaran, Ruby B. Lee. "Framework for Design Validation of Security Architectures," *Princeton University Department of Electrical Engineering Technical Report CE-L2008-013*, November 2008.

**Jeffrey Dvoskin**, Ganesha Bhaskara, Thuy D. Nguyen, Ruby Lee, "SecureCore Prototype/Demo Manual," Version 1.0. *Princeton University Department of Electrical Engineering Technical Report CE-L2008-009*, 8/11/2008.

**Jeffrey Dvoskin**, Ruby B. Lee, "SP Processor Architecture Reference Manual," *Princeton University Department of Electrical Engineering Technical Report CE-L2008-008*, 8/11/2008. (Previous version: CE-L2007-009. Version 0.7, 11/21/2007)

**Jeffrey Dvoskin**, Ruby B. Lee, "Processor Architecture for Remote, Transient, Policy-controlled Secrets," *Princeton University Department of Electrical Engineering Technical Report CE-L2006-007*, November 2006.

Ganesha Bhaskara, Timothy E. Levin, Thuy D. Nguyen, Cynthia E. Irvine, Terry V. Benzel, **Jeffrey Dvoskin**, Ruby B. Lee, "Virtualization of Secure Processor Key Management within a Separation Kernel Architecture," *Princeton University Department of Electrical Engineering Technical Report CE-L2006-006*, November 2006.

Ruby B. Lee, **Jeffrey Dvoskin**, David Champagne, "Fundamental Architectural Features in SP Processors for Protecting Sensitive Information," 2006. *Unpublished*.

**Jeffrey Dvoskin**, Ruby B. Lee, "Enabling Transient Access to Protected Information for Crisis Response," *Princeton University Department of Electrical Engineering Technical Report CE-L2006-001*, May 2006.