

Michael Kaminsky

MIT Computer Science and Artificial Intelligence Laboratory
77 Massachusetts Ave, 32-G994
Cambridge, MA 02139
617-253-7328 (W)
617-577-5864 (H)

kaminsky@csail.mit.edu
<http://www.pdos.lcs.mit.edu/~kaminsky/>

Education

Massachusetts Institute of Technology

GPA: 5.0/5.0

Ph.D. Department of Electrical Engineering and Computer Science, Spring/Summer 2004 (expected)
Dissertation: User Authentication and Remote Execution Across Administrative Domains
Advisers: Frans Kaashoek, David Mazières (NYU)

S.M. Department of Electrical Engineering and Computer Science, May 2000

Thesis: Flexible Key Management with SFS Agents
Advisers: Frans Kaashoek, David Mazières (NYU)

University of California at Berkeley

GPA: 4.0/4.0

B.S. Department of Electrical Engineering and Computer Science, May 1998 (Highest Honors)

Professional Experience

MIT, Computer Science and Artificial Intelligence Laboratory

Advisers: Frans Kaashoek, David Mazières (NYU)

Research Assistant, August 1998–present

Co-designed and co-built the Self-Certifying File System (SFS); develop and maintain various pieces of the SFS project. SFS is a global file system that does not rely on centralized trust. It enables file sharing over insecure networks and across administrative domains. The SFS computing environment includes a file system, remote execution tool, user agent, and authentication server. SFS supports flexible server key management, several user authentication mechanisms, revocation, and access control lists. SFS is built at user-level primarily in C++ using an asynchronous RPC library. SFS is open-source and is in daily use in research, industry, and education; it runs on several platforms, including Linux, OpenBSD, FreeBSD, and MacOS X. [3,5]

Research Assistant, January 2003–present **[Dissertation]**

Designed and built support for remote users and groups in SFS. In a global file system, people want to share their files with remote users and groups—users and groups defined in remote administrative domains—without assuming any kind of pre-existing administrative relationship. Supporting remote groups, in particular, is challenging because their definitions (membership lists) are maintained remotely and because those membership lists can themselves contain other remote groups. SFS supports remote groups by restricting what file owners can place on Access Control Lists (ACLs). SFS ACLs can contain only local users and groups. These local groups, however, can contain remote users and groups. When a user authenticates to the file system, the server need only determine the local groups to which a user belongs in order to perform access control. Because determining local group membership at authentication time would be too slow—it could involve fetching several remote group membership lists—SFS authentication servers expand all locally defined groups ahead of time, periodically in the background, and cache the results. Then, at authentication time, the server can determine group membership of a user based on its local user and group database plus this cache. [2]

Research Assistant, June 2001–January 2003, November 2003–present **[Dissertation]**

Co-designed and co-built REX, a remote login and execution tool for the SFS environment. REX has a novel architecture designed to provide extensibility, security, and transparent connections in the absence of global routing. REX offers extensibility by emulating file description passing between processes on different machines; this abstraction allows users to add new features to REX without changing the protocol. The exploitable code (facing the network) and the trusted code (running with enhanced privileges) in REX is small, making it easy to audit. REX also supports session caching, session resumption under changing IP addresses, trust policies for logging into less-trusted machines and it works across NAT. [1]

Research Assistant, August 1998–June 2000 **[Master's Thesis]**

Helped design and implement the SFS agent, a user-controlled program that interacts with other parts of the SFS system on behalf of the user. The SFS agent handles user authentication (signing authentication requests with the user's private key), server key management, and revocation. My Master's research dealt with the problem of how to achieve flexible server key management in secure systems. The SFS agent offers a toolbox approach to key management; users register *certification programs* with their agents to customize the authentication process. Certification programs can provide SFS users with a variety of key management techniques such as SSH's `.known_hosts` file, SSL, and Kerberos.

Mazu Networks, Cambridge, MA

Consultant, 2002–present

Developed software related to network traffic profiling. Analyzed performance properties of a hash table implementation; designed and developed a Java-based worm visualization applet; added SQL infrastructure to a Postgres database to allow external reporting through Crystal Reports software.

Xerox Palo Alto Research Center (PARC)

Intern, Summer 1998

Advisers: Paul Dourish, Keith Edwards, Anthony LaMarca, Michael Salisbury

Designed and built a set of software tools for Programmable Embodied Agents—portable, wireless, interactive devices that take the form of identifiable characters. The primary focus of the research was to create a protocol stack to communicate with the ActiMates Barney doll. This protocol stack was used to build applications that allowed users to interact with their environments through Barney. [4]

Intern, Summer 1997

Adviser: David Goldberg

Explored text entry on pen-based user interfaces by extending the Palm Pilot's operating system to record characters entered using the PalmOS's Graffiti writing system. Users could then upload those strokes onto a desktop computer for analysis.

Intern, Summer 1996

Adviser: Bernardo Huberman

Designed and built a system for collaborative filtering and sharing of information called Beehive. Users of the system join dynamic "communities of practice" which have as members other users with similar interests. Through a simple GUI, a user can distribute a particularly interesting piece of information to the other members of the community. [6]

UC Berkeley, Department of Electrical Engineering and Computer Science

Undergraduate Research, Spring 1997

Advisers: Eric Brewer, David Wagner, Ian Goldberg

Extended two in-progress projects: Janus and IPSE. Janus executes programs in a so-called "sand box," denying all external actions (system calls) except those explicitly allow by the user; helped design a Janus configuration to run the *sendmail* program safely. IPSE, the Internet Protocol Scanning Engine, is a module-driven packet-sniffer; extended IPSE to work cross-platform and with different types of network interfaces; added ability to write packets in response to certain traffic (e.g., a network attacks).

UC Berkeley, Residential Computing

Special Projects Coordinator, 1997–1998

Handled multi-platform system administration, including WWW development and various programming projects in C, Perl and SQL to support four to five on-campus residence hall computer labs running Windows, MacOS and Linux.

Technical Coordinator, 1996–1997

Oversaw the operation of over 20 Windows, Macintosh, and Linux computers connected to a Netware server for printer and file sharing at a residence hall computer lab; helped run staff meetings of 10 Computing Assistants.

Teaching Experience

MIT, Department of Electrical Engineering and Computer Science

Head Teaching Assistant, Computation Structures (6.004), Fall 2002

Handled various administrative duties related to class organization; held office and lab hours to assist students with course material and projects; reviewed and debugged quizzes with instructors (five total).

UC Berkeley, Department of Electrical Engineering and Computer Science

Teaching Assistant, Machine Structures (CS 61C), Spring 1998

Taught lab and discussion sections with approximately 30 students; graded exams; held office hours to assist students with course material.

Refereed Publications

1. **Michael Kaminsky**, Eric Peterson, Daniel B. Giffin, Kevin Fu, David Mazières, and M. Frans Kaashoek. REX: Secure, Extensible Remote Execution. In *Proceedings of the 2004 USENIX Annual Technical Conference (USENIX'04)*, Boston, Massachusetts, June 2004. To appear.
2. **Michael Kaminsky**, George Savvides, David Mazières, and M. Frans Kaashoek. Decentralized User Authentication in a Global File System. In *Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP'03)*, pages 60-73, Bolton Landing, NY, October 2003.
3. David Mazières, **Michael Kaminsky**, M. Frans Kaashoek, and Emmett Witchel. Separating key management from file system security. In *Proceedings of the 17th ACM Symposium on Operating Systems Principles (SOSP'99)*, pages 124-139, Kiawah Island, SC, December 1999.
4. **Michael Kaminsky**, Paul Dourish, Keith Edwards, Anthony LaMarca, Michael Salisbury and Ian Smith. SWEETPEA: Software Tools for Programmable Embodied Agents. In *Proceedings of the ACM Conference Human Factors in Computing Systems (CHI'99)*, pages 144-151, Pittsburg, PA, May 1999.

Other Publications

5. Kevin Fu, **Michael Kaminsky**, and David Mazières. Using SFS for a Secure Network File System. In *login: The Magazine of USENIX and Sage*, December 2002.
6. Bernardo A. Huberman and **Michael Kaminsky**. Beehive: A System for Cooperative Filtering and Sharing of Information. Technical Report, Dynamics of Computation Group, Xerox Palo Alto Research Center, 1996.

Skills

C/C++, Linux/Unix, Python, Perl, Bourne/C/Z Shell

Awards and Honors

National Science Foundation Graduate Fellowship (1998)
UC Berkeley EECS Honors Degree Program (Spring 1996–May 1998)
Eta Kappa Nu, EECS National Honors Society (member)
Golden Key National Honors Society (member)

References

Prof. M. Frans Kaashoek
MIT Computer Science and Artificial Intelligence Laboratory
200 Technology Square, Room 522
Cambridge, MA 02139, USA
617-253-7149
kaashoek@csail.mit.edu

Prof. David Mazières
NYU Department of Computer Science
715 Broadway, Room 708
New York, NY 10003, USA
212-998-3492
dm@cs.nyu.edu

Prof. Robert Morris
MIT CS and AI Laboratory
200 Technology Square, Room 509
Cambridge, MA 02139, USA
617-253-5983
rtm@csail.mit.edu

Prof. Daniel Jackson
MIT CS and AI Laboratory
200 Technology Square, Room 401
Cambridge, MA 02139, USA
617-258-8471
dnj@csail.mit.edu

Last updated: 22 March 2004