

David Mazières

Associate Professor of Computer Science
(and, by courtesy, of Electrical Engineering)

Stanford University
353 Serra Mall, Room 290
Stanford, CA 94305

(650) 723-8777
<http://www.scs.stanford.edu/~dm/>

Education

Massachusetts Institute of Technology Cambridge, MA

Ph.D. in Electrical Engineering and Computer Science September, 2000

Thesis title: *Self-certifying File System*.

Advisor: Prof. M. Frans Kaashoek.

S.M. in Electrical Engineering and Computer Science September, 1997

Thesis title: *Security and Decentralized Control in the SFS Global File System*.

Harvard University Cambridge, MA

A.B. with honors in Computer Science June, 1994

Thesis title: *Abstract Execution in a Multi-Tasking Environment*.

Interests

Computer systems, especially operating systems, security, and privacy.

Research

2005–present

HiStar. With Nikolai Zeldovich, Silas Boyd-Wickizer, Eddie Kohler, and Ryan Stutsman, working on HiStar, a new operating system based on Asbestos labels. HiStar provides a Unix-like environment in an untrusted library on top of a low-level kernel that provides six object types for which all information flow is explicit. This allows strict control over which components of a system may affect which other. HiStar solves a number of challenges. In particular, it tracks information flow dynamically without allowing the tracking mechanism itself to leak information. Moreover, by decoupling resource allocation and revocation from all other forms of resource access, HiStar makes it possible to administer an operating system that has no inherent notion of superuser.

2004–present

Asbestos. With Russ Cox, Petros Efstathopoulos, Cliff Frey, Daniel Giffin, Max Krohn, Steve VanDeBogart, David Ziegler, Frans Kaashoek, Eddie Kohler, Robert Morris, and others, designing and implementing the Asbestos operating system [2]. The goal of Asbestos is to reason about the behavior and security of applications without understanding the applications themselves. Asbestos is a message-passing kernel with a new label mechanism that provides information flow control with decentralized declassification privileges.

2000–present

Untrusted file servers. With Jinyuan Li, Maxwell Krohn, and Dennis Shasha, developed the SUNDR [8] file system at NYU. SUNDR, “secure untrusted data repository,” is a network file system designed to run on untrusted servers. Novel protocols [15] allow SUNDR to make strong guarantees on the integrity and freshness of file data without requiring any on-line trusted parties. Additionally, SUNDR will recover from misbehaving servers by integrating digitally signed data from large persistent caches on untrusted clients.

- 2002–present **Peer-to-peer systems.** With Petar Maymounkov, worked on Kademlia [30], a scalable distributed hash table boasting several desirable properties not previously achievable simultaneously. Notably, Kademlia’s symmetric and unidirectional distance metric avoids the need for either special-purpose route maintenance messages or a separate routing mechanism for neighboring nodes.
- With Michael Freedman, working on Coral [11], a peer-to-peer content distribution system. Coral indexes local data, such as the files in a node’s web cache, so as to ensure that content is replicated exactly in proportion to its use. Coral uses a new abstraction, called a *distributed sloppy hash table* (DSHT), that lets nodes locate files, regardless of their popularity, without causing hot spots in the indexing infrastructure. Based on the DSHT interface, Coral implements a decentralized clustering algorithm that lets nodes find nearby data without needing to query more distant machines.
- 2004–present **Mail Avenger.** Mail Avenger [34] puts users in control of how servers behave when receiving mail. Several research projects have picked up Mail Avenger because it collects interesting information and allows easy prototyping of new spam solutions.
- 1996–present **SFS.** Conceived and led the SFS project. Designed and built SFS [20], the “self-certifying file system,” with contributions from Kaminsky, Kaashoek, Krohn, and others. SFS was motivated by the lack of any practical, secure software system that could scale to the size of the Internet. Most secure systems either hinder their own deployment with cumbersome key certification procedures, fail to work well across administrative realms, or both. SFS solves these problems through *self-certifying pathnames*—file names that explicitly specify a server’s public key. Self-certifying pathnames let multiple key management schemes coexist, so that users can choose the one most convenient for any given server. SFS is available from <http://www.fs.net/> and in the package collections of several operating systems including Debian Linux and FreeBSD.
- 2000–2003 **Censorship-resistant publishing.** With Marc Waldman, developed Tangler [17], a censorship-resistant publishing system. Tangler breaks documents into blocks that must be combined with previously published blocks to recover contents. This “entanglement” dissociates responsibility for documents from any particular server and makes replicating others’ documents an inherent part of publishing. Tangler supports secure updates and circular dependencies, letting documents link to each other securely. Its design uses storage certificates and automatic self-policing to resist flooding and other denial-of-service attacks to which anonymous publishing makes other systems vulnerable.
- 1996–2002 **Robust anonymous systems.** Conceived, designed, built, and operated the `nym.alias.net` pseudonym server [21]. `nym.alias.net` lets people create email aliases anonymously. The system is designed to prevent even its own administrators from learning users’ real identities. Anonymous services draw attacks; their defense is complicated by attackers exploiting anonymity as well as the fact that many traditional defense techniques can hurt the privacy of legitimate users. I developed numerous means of protecting anonymous servers and field-tested them against real attacks on a system in day-to-day use by thousands of people.
- 1995–1997 **Exokernel.** Designed and built *xok* [22], an exokernel for the x86 architecture, based on the philosophy of the first generation MIPS exokernel. Novel contributions included *xok*’s protection architecture [33] and its approach to global resource management. *Xok* formed the basis of Exotec corporation’s software product.
- 1994–1995 **Performance Analysis.** *alpha architecture:* Wrote a call graph profiler for the Digital Unix kernel. Developed tools to manipulate low-level PALcode that runs underneath the operating system. Combined PALcode with the ATOM instrumentation tool to analyse cache performance. *x86 architecture:* Analyzed and compared the performance of operating systems using the Pentium hardware event counters [23]. *sparc architecture:* Extended Larus’s *qpt* tool and modified the BSD 4.4 kernel to gather fully interleaved multitasking address traces with low overhead [37].

Teaching

Operating Systems (CS140). An introduction to operating systems, with rigorous programming assignments.

Computer and Network Security (CS155). Covers principles of computer systems security, including attack techniques and how to defend against them, with programming assignments.

Distributed Systems (CS244B). Covers the principles of distributed systems, including replication, consistency, Byzantine fault tolerance, scalability, naming, and many examples of systems addressing these challenges. Includes two short programming assignments followed by a small research project done in teams.

Advanced Topics in Operating Systems (CS 240). Survey of classic and new papers on operating systems topics, covering virtual memory, synchronization, virtual machines, file systems, scalability, kernel architectures, reliability, and security.

Distributed Storage Systems (CS240D). A detailed examination of a number of topics including RPC programming, disk layout, transactions, and reconciliation in weakly consistent systems. The class includes lab assignments and a project in which students implement a fully-functional cryptographic file system.

Advanced Operating Systems Implementation (CS240C). Covers the design and implementation of operating systems. Successive laboratory assignments have students build a real, working, operating system for the x86 personal computer, using a design similar to the Exokernel [22]. Lectures cover the technical details of hardware necessary to build an operating system, but concentrate mostly on a detailed survey of operating system research, discussing papers about experimental operating systems from MULTICS all the way to recent publications.

NYU Classes

Computer Networks (undergraduate course). An introduction to networking from the physical layer to communications protocols to the design of distributed applications. Includes programming assignments in C to make the details more concrete.

Honors Operating Systems (graduate course). A survey of recent operating systems research combined with lab assignments and a final project. Students' final projects have included a Unix file system, a gnutella-like peer-to-peer file sharing system, a distributed buffer cache for web server farms, and a privacy-preserving peer-to-peer plagiarism detection system. As with much contemporary research, the class focuses on distributed systems and thus does not overlap with the undergraduate Advanced Operating Systems class listed above.

Computer System Security (graduate course). A survey of numerous aspects of computer system security, including authentication protocols, key management, electronic privacy, automated program checking, and mandatory access control systems.

Positions

9/07–present

Associate Professor. Stanford University, Stanford, CA.

9/05–8/07

Assistant Professor. Stanford University, Stanford, CA.

9/05–8/07

Associate Professor (on leave). New York University, New York, NY.

9/00–8/05

Assistant Professor. New York University, New York, NY.

6/99–8/99

Summer Intern. Bell Labs, Murray Hill, NJ.

6/95–8/95

Summer Intern. DEC Western Research Laboratory, Palo Alto, CA.

6/94–5/95

Operating Systems Programmer. Harvard University, Cambridge, MA.

6/93–8/93

Operating Systems Summer Intern. Kendall Square Research, Waltham, MA.

2/91–5/95

Late-night Rock DJ. WHRB-FM, Cambridge, MA.

Students

PhD

Current: Ryan Stutsman.

Graduated: Marc Waldman, Michael Kaminsky, Jinyuan Li, Antonio Nicolosi, Michael Freedman, Nickolai Zeldovich, Siddhartha Annapureddy.

Masters

Graduated: Silas Boyd-Wickizer, David Euresti, Michael Kaminsky, Petar Maymounkov, Eric Peterson, Kyle Rose, George Savvides.

Awards

Alfred P. Sloan Research Fellow, 2002 (see support).

Best paper award, USENIX 2001 [19].

NSF Career award, 2001 (see support).

MIT George R. Sprows award for best thesis in computer science, 2000.

Refereed conference and journal publications

- [1] Nickolai Zeldovich, Silas Boyd-Wickizer, and David Mazières. Securing distributed systems with information flow control. In *Proceedings of the 6th Symposium on Networked Systems Design and Implementation*, pages 293–308, San Francisco, CA, April 2008.
- [2] Steve VanDeBogart, Petros Efstathopoulos, Eddie Kohler, Maxwell Krohn, Cliff Frey, David Ziegler, Frans Kaashoek, Robert Morris, and David Mazières. Labels and event processes in the Asbestos operating system. *ACM Transactions on Computer Systems*, 25(4):11:1–43, December 2007. A version appeared in *Proceedings of the 20th ACM Symposium on Operating System Principles*, 2005.
- [3] Jinyuan Li and David Mazières. Beyond one-third faulty replicas in Byzantine fault tolerant systems. In *Proceedings of the 4th Symposium on Networked Systems Design and Implementation*, pages 131–144, Cambridge, MA, April 2007.
- [4] Nickolai Zeldovich, Silas Boyd-Wickizer, Eddie Kohler, and David Mazières. Making information flow explicit in HiStar. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*, pages 263–278, Seattle, WA, November 2006.
- [5] Michael J. Freedman, Karthik Lashkminarayanan, and David Mazières. OASIS: Anycast for any service. In *Proceedings of the 3rd Symposium on Networked Systems Design and Implementation*, pages 129–142, San Jose, CA, May 2006.
- [6] Scott Garriss, Michael Kaminsky, Michael J. Freedman, Brad Karp, David Mazières, and Haifeng Yu. RE: Reliable email. In *Proceedings of the 3rd Symposium on Networked Systems Design and Implementation*, pages 297–310, San Jose, CA, May 2006.
- [7] Siddhartha Annapureddy, Michael J. Freedman, and David Mazières. Shark: Scaling file servers via cooperative caching. In *Proceedings of the 2nd Symposium on Networked Systems Design and Implementation*, pages 129–142, Boston, MA, May 2005.
- [8] Jinyuan Li, Maxwell Krohn, David Mazières, and Dennis Shasha. Secure untrusted data repository (SUNDR). In *Proceedings of the 6th Symposium on Operating Systems Design and Implementation*, pages 91–106, San Francisco, CA, December 2004.
- [9] 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*, pages 199–212, Boston, MA, June–July 2004. USENIX.

- [10] Maxwell N. Krohn, Michael J. Freedman, and David Mazières. On-the-fly verification of rateless erasure codes for efficient content distribution. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 226–240, Oakland, CA, May 2004.
- [11] Michael J. Freedman, Eric Freudenthal, and David Mazières. Democratizing content publication with Coral. In *Proceedings of the 1st Symposium on Networked Systems Design and Implementation*, pages 239–252, San Francisco, CA, March 2004.
- [12] 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*, pages 60–73, Bolton Landing, NY, October 2003. ACM.
- [13] Nickolai Zeldovich, Alexander Yip, Frank Dabek, Robert T. Morris, David Mazières, and M. Frans Kaashoek. Multiprocessor support for event-driven programs. In *Proceedings of the 2003 USENIX*, pages 239–252, San Antonio, TX, June 2003. USENIX.
- [14] Antonio Nicolosi, Maxwell Krohn, Yevgeniy Dodis, and David Mazières. Proactive two-party signatures for user authentication. In *Proceedings of the 10th Annual Network and Distributed System Security Symposium*, pages 233–248, February 2003.
- [15] David Mazières and Dennis Shasha. Building secure file systems out of Byzantine storage. In *Proceedings of the 21st Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, pages 108–117, July 2002. The full version is available as NYU computer science department technical report TR2002-826, May 2002.
- [16] Kevin Fu, M. Frans Kaashoek, and David Mazières. Fast and secure distributed read-only file system. *ACM Transactions on Computer Systems*, 20(1):1–24, February 2002. A version appeared in *Proceedings of the 4th Symposium on Operating Systems Design and Implementation*, 2000.
- [17] Marc Waldman and David Mazières. Tangler – a censorship-resistant publishing system based on document entanglements. In *Proceedings of the 8th ACM Conference on Computer and Communications Security*, pages 126–135, November 2001.
- [18] Athicha Muthitacharoen, Benjie Chen, and David Mazières. A low-bandwidth network file system. In *Proceedings of the 18th ACM Symposium on Operating Systems Principles*, pages 174–187, Chateau Lake Louise, Banff, Canada, October 2001. ACM.
- [19] David Mazières. A toolkit for user-level file systems. In *Proceedings of the 2001 USENIX*, pages 261–274. USENIX, June 2001.
- [20] 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*, pages 124–139, Kiawa Island, SC, December 1999. ACM.
- [21] David Mazières and M. Frans Kaashoek. The design, implementation and operation of an email pseudonym server. In *Proceedings of the 5th ACM Conference on Computer and Communications Security*, pages 27–36, 1998.
- [22] M. Frans Kaashoek, Dawson R. Engler, Gregory R. Ganger, Héctor M. Briceño, Russell Hunt, David Mazières, Thomas Pinckney, Robert Grimm, John Jannotti, and Kenneth Mackenzie. Application performance and flexibility on exokernel systems. In *Proceedings of the 16th ACM Symposium on Operating Systems Principles*, pages 52–65, Saint-Malo, France, October 1997. ACM.
- [23] J. Bradley Chen, Yasuhiro Endo, Kee Chan, David Mazières, Antonio Dias, Margo Seltzer, and Mike Smith. The measured performance of personal computer operating systems. *ACM Transactions on Computer Systems*, 14(1):3–40, February 1996. A version appeared in *Proceedings of the 15th ACM Symposium on Operating System Principles*, 1995.

Refereed workshop publications

- [24] Michael J. Freedman, Ion Stoica, David Mazières, and Scott Shenker. Group therapy for systems: Using link attestations to manage failures. In *Proceedings of the 5th International Workshop on Peer-to-Peer Systems*, Santa Barbara, CA, February 2006.
- [25] Maxwell Krohn, Petros Efstathopoulos, Cliff Frey, Frans Kaashoek, Eddie Kohler, David Mazières, Robert Morris, Michelle Osborne, Steve VanDeBogart, and David Ziegler. Make least privilege a right (not a privilege). In *Proceedings of the 10th Workshop on Hot Topics in Operating Systems*, Santa Fe, NM, June 2005.
- [26] Antonio Nicolosi and David Mazières. Secure acknowledgment of multicast messages in open peer-to-peer networks. In *Proceedings of the 3rd International Workshop on Peer-to-Peer Systems (IPTPS '04)*, pages 259–268, San Diego, CA, February 2004.
- [27] Petar Maymounkov and David Mazières. Rateless codes and big downloads. In *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03)*, pages 247–255, Berkeley, CA, February 2003.
- [28] Michael Freedman and David Mazières. Sloppy hashing and self-organizing clusters. In *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03)*, pages 45–55, Berkeley, CA, February 2003.
- [29] Frank Dabek, Nickolai Zeldovich, M. Frans Kaashoek, David Mazières, and Robert Morris. Event-driven programming for robust software. In *Proceedings of the 10th ACM SIGOPS European Workshop*, pages 186–189, September 2002.
- [30] Petar Maymounkov and David Mazières. Kademia: A peer-to-peer information system based on the XOR metric. In *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02)*, pages 53–65, Cambridge, MA, March 2002.
- [31] David Mazières and Dennis Shasha. Don't trust your file server. In *Proceedings of the 8th Workshop on Hot Topics in Operating Systems*, pages 113–118, May 2001.
- [32] David Mazières and M. Frans Kaashoek. Escaping the evils of centralized control with self-certifying pathnames. In *Proceedings of the 8th ACM SIGOPS European Workshop*, pages 118–125, 1998.
- [33] David Mazières and M. Frans Kaashoek. Secure applications need flexible operating systems. In *Proceedings of the 6th Workshop on Hot Topics in Operating Systems*, pages 56–61, May 1997.

Unrefereed publications and technical reports

- [34] David Mazières. Blocking unwanted mail with mail avenger. *Virus Bulletin*, pages S2–S4, July 2005.
- [35] Kevin Fu, Michael Kaminsky, and David Mazières. Using SFS for a secure network file system. *login: The Magazine of Usenix & Sage*, 27(6):6–16, December 2002.
- [36] Niels Provos and David Mazières. A future-adaptable password scheme. In *Proceedings of the 1999 USENIX, Freenix track (the on-line version)*, Monterey, CA, June 1999. USENIX. from <http://www.usenix.org/events/usenix99/provos.html>.
- [37] David Mazières and Michael D. Smith. Abstract execution in a multi-tasking environment. Technical Report TR-31-94, Harvard University, November 1994.

Support

David Mazières, Dawson Engler, Christos Kozyrakis, and Nick McKeown. A Clean-Slate Infrastructure for Information Flow Control. NSF Cybertrust. \$1M. 2007–2010.

Nick McKeown, Dan Boneh, David Mazières, and Mendel Rosenblum, linked with Aditya Akella (U. of Wisconsin). Designing Secure Networks from the Ground up. NSF NeTS-FIND award, CNS-0627112. \$759K. 2006–2009.

David Mazières, M. Frans Kaashoek, Robert Morris, and Eddie Kohler. Securing untrusted software with interposition. Joint NSF Cybertrust and DARPA grant, CNS-0430425. \$1.3M, including \$539K subaward to MIT and \$312K subaward to UCLA. 2004–2006.

M. Frans Kaashoek *et al.* Robust Large-Scale Distributed Systems. NSF Large ITR, ANI-0225660. \$12M (MIT), including \$800K subaward to NYU, transferred to Stanford 9/2005, plus extra \$233K for Stanford. 2002–2007.

Past support

David Mazières. Alfred P. Sloan fellowship, BR 4169. \$40K. 2002–2006.

David Mazières, M. Frans Kaashoek, Robert Morris, and Eddie Kohler. Asbestos: Securing untrusted software with interposition. DARPA seedling, FA8750-04-1-0090. \$150K, including \$60K subaward to MIT and \$30K subaward to UCLA. 2004.

David Mazières, M. Frans Kaashoek, Robert Morris, and Eddie Kohler. Interposition-friendly operating systems. DARPA seedling, MDA972-03-P-0015. \$43K, including \$10K subaward to MIT and \$10K subaward to ICIR. 2003.

David Mazières and M. Frans Kaashoek. Secure File Sharing Over the Internet Using SFS. DARPA Composable High-Assurance Trusted Systems program, N66001-01-1-8927. \$841K (NYU), including \$250K subaward to MIT. 2001–2003.

David Mazières. SUNDR: The Secure Untrusted Data Repository. NSF Career award, CCR-0093361. \$250K. 2000–2006.

Stanford, CA, June 27, 2008