Christopher T Lesniewski-Laas

Updated March 2010

Contact Information

MIT Computer Science and Artificial Intelligence Laboratory
32 Vassar Street 32-G996
Cambridge, MA 02139

+1 617 253 0004
ctl@mit.edu
lesniewski.org

Research Interests

Distributed and decentralized computer systems, communication networks, social networks, scaling, and security.

Education

Massachusetts Institute of Technology

Ph.D. candidate, Computer Science — 2003–2010 (expected)
Dissertation: *A Secure and Decentralized Distributed Hash Table*
Advisor: Professor M. Frans Kaashoek

M.Eng., Electrical Engineering and Computer Science — 2001–2003
Thesis: *SSL Splitting and Barnraising: Cooperative Caching with Authenticity Guarantees*

S.B., Electrical Engineering and Computer Science — 1997–2003
S.B., Mathematics — 1997–2001
Minor: Physics

Employment

MIT CSAIL, Cambridge, MA — 2001–Present
Research Assistant, Parallel and Distributed Operating Systems Group.

University of Cambridge Computer Lab, Cambridge, UK — Summer 2004
Visiting scholar, Cambridge-MIT Institute, *"Next generation peer-to-peer networks"* project.

Permabit, Cambridge, MA — Summer–Fall 2001
Designed and developed robust and scalable data storage system.

Microsoft Research, Redmond, WA — Summer 2000
Designed cryptographic protocols for smart card based access control.

SensAble Technologies, Cambridge, MA — Summer 1999
Developed hardware and software for a robotic haptic interface.

MIT AI Lab, Mathematics and Computation, Cambridge, MA — Summer 1998
Developed software to simulate amorphous computers.

Teaching

Recitation Instructor (position usually filled by faculty) — Spring 2005
Head Teaching Assistant — Spring 2004
Teaching Assistant — Spring 2003
MIT 6.033: *Computer Systems Engineering*

Lab Assistant — Fall 2000
MIT 6.001: *Structure and Interpretation of Computer Programs*

Refereed Conference, Workshop, and Journal Publications

[1] **Whānau: A Sybil-proof Distributed Hash Table.**
Chris Lesniewski-Laas and M. Frans Kaashoek.
In *Proceedings of the Symposium on Networked System Design and Implementation,* San Jose, California, April 2010.
Abstract PDF PS.

[2] **Device Transparency: a new model for mobile storage.**
Jacob Strauss, Chris Lesniewski-Laas, Justin Mazzola Paluska, Bryan Ford, Robert Morris, and M. Frans Kaashoek.
In *Proceedings of the SOSP Workshop on Hot Topics in Storage and File Systems (HotStorage),* Big Sky, Montana, October 2009.
Abstract PDF PS.

[3] **A Sybil-proof one-hop DHT.**
Chris Lesniewski-Laas.
In *Proceedings of the Workshop on Social Network Systems,* Glasgow, Scotland, April 2008.
Abstract PDF PS.

[4] **Alpaca: extensible authorization for distributed services.**
Chris Lesniewski-Laas, Bryan Ford, Jacob Strauss, Robert Morris, and M. Frans Kaashoek.
In *Proceedings of the ACM Conference on Computer and Communications Security,* ACM, Alexandria, Virginia, October 2007.
Abstract PDF PS.

[5] **Persistent personal names for globally connected mobile devices.**
Bryan Ford, Jacob Strauss, Chris Lesniewski-Laas, Sean Rhea, M. Frans Kaashoek, and Robert Morris.
In *Proceedings of the Symposium on Operating System Design and Implementation,* Seattle, Washington, November 2006.
Abstract HTML PDF PS.

[6] **User-relative names for globally connected personal devices.**
Bryan Ford, Jacob Strauss, Chris Lesniewski-Laas, Sean Rhea, M. Frans Kaashoek, and Robert Morris.
In *Proceedings of the International Workshop on Peer-to-Peer Systems,* Santa Barbara, California, February 2006.
Abstract PDF PS.

[7] **Sybil-resistant DHT routing.**
George Danezis, Chris Lesniewski-Laas, M. Frans Kaashoek, and Ross Anderson.
In *Proceedings of the European Symposium On Research In Computer Security,* Milan, Italy, September 2005.
Abstract PDF PS.

[8] **SSL splitting: securely serving data from untrusted caches.**
Chris Lesniewski-Laas and M. Frans Kaashoek.
In *Computer Networks,* 48(5):763–779, Elsevier, August 2005.
Abstract HTML PDF PS.

[9] **SSL splitting: securely serving data from untrusted caches.**
Chris Lesniewski-Laas and M. Frans Kaashoek.
In *Proceedings of the USENIX Security Symposium,* Washington, D.C. August 2003.
Abstract HTML PDF PS.

Other Publications

[10] **Whānaungatanga: Sybil-proof routing with social networks.**
Chris Lesniewski-Laas and M. Frans Kaashoek.
MIT, Technical Report MIT-CSAIL-TR-2009-045, September 2009.
Abstract PDF PS.

[11] **SSL splitting and Barnraising: cooperative caching with authenticity guarantees.**
Chris Lesniewski-Laas.
Master's Thesis, Massachusetts Institute of Technology, February 2003.
Abstract PDF PS.

External Talks

Yale University, New Haven, CT — February 2010
A Sybil-proof Distributed Hash Table.

Microsoft Research, Redmond, WA — December 2008
Defending against Sybils using the social network.

Nokia, Oulu, Finland — June 2008
Compact Internet routing.

EuroSys Workshop on Social Network Systems, Glasgow, UK — April 2008
A Sybil-proof DHT using a social network.

University of Cambridge Computer Laboratory, Cambridge, UK — March 2008
A Sybil-proof DHT using a social network.

Nokia Research, Cambridge, MA — January 2008
Alpaca, a really flexible authentication framework.

ACM Conference on Computer and Communications Security (CCS) — October 2007
Extensible proof-carrying authorization in Alpaca.

IRIS Student Workshop — November 2004
Does overlay routing security require admission control?

Johns Hopkins University, Baltimore, MD — August 2004
SSL Splitting.

USENIX Security Symposium — August 2003
SSL Splitting.

Software Artifacts

Whānau : secure and scalable distributed hash table. 2010
Eyo: device-transparent personal storage system. 2009
Alpaca: extensible proof-carrying-authorization framework library. 2007
UIA: naming and routing protocol suite for personal mobile devices. 2006
Barnraising: distributed caching Web proxy using SSL Splitting. 2003
SSL Splitting: drop-in replacement for OpenSSL library enabling untrusted caches. 2003

Professional Activities

Program Committee, ACM Symposium on Applied Computing (Security Track), 2007–2010.

External reviews include: SOSP 2003,2005,2007, SIGCOMM 2003, IPTPS 2003, NDSS 2004, J. Computer Networks (2004), FAST 2005, CCS 2006, ISIT 2009, SNS 2009, Trans. Vehicular Tech (2009), TISSEC (2010).

Affiliations and Honors

ACM, USENIX, SIPB, Phi Beta Kappa, Kosciuszko Foundation Fellowship.

Research Projects

Whānau — Sybil-proof Secure Distributed Hash Table 2008–Present

The topic of my dissertation, Whānau is a secure Distributed Hash Table (DHT): a structured overlay which can quickly look up the node responsible for a given key. DHTs have many applications, including key-value databases, filesystems, caching, rendezvous services, and multicast trees. Whānau uses the high connectivity of natural social networks to bootstrap a highly robust overlay network. Any attacker must infiltrate a large fraction of the social network in order to cause any damage to the system's availability. In previous DHTs, an attacker can cause a massive Denial of Service (DoS) simply by creating a large number of pseudonyms. Previous defenses against this "Sybil attack", a problem identified in 2001, required a centralized gatekeeper which is somehow able to distinguish the good identities from evil pseudonyms. For example, Amazon's Dynamo DHT operates only within Amazon's data centers, and CoralCDN's DHT contains only PlanetLab servers. Whānau eliminates this admission control function, enabling truly decentralized and cooperative P2P DHT infrastructure to be built.

This work appeared at SocialNets 2008 [3] and will appear at NSDI 2010 [1]. An earlier paper in ESORICS 2005 [7] introduced the social network model later used by Whānau.

UIA — User Information Architecture 2004–2008

Eyo — Device Transparent Storage 2008–Present

UIA is a routing and naming layer designed to organize users' many personal devices, such as laptops, phones, cameras, and media players, into a coherent cluster. Users introduce their devices to each other using a secure physical rendezvous in which the devices exchange cryptographic keys; thereafter, UIA's routing layer ensures that the devices can find and contact each other whenever they are connected to the same network. The user assigns personal names to each device and UIA propagates records appropriately to ensure that the namespace is consistent across all devices. In addition, users can assign names to other users, and can apply these names recursively to navigate the social network. For example, the name `phone.dad.bob` would refer to *Bob's father's telephone*. Since no device or server is designated as the "master" of a user's cluster, UIA's main challenge is securely handling updates to the cluster's membership, including cases in which some devices may be offline or acting maliciously.

Eyo, a continuation of the UIA project, tackles the problem of providing a consistent view of a user's data objects (such as photos, music, and email) across all of her devices. We call this property *device transparency*. As with UIA, the challenge is to provide a consistent view despite varying device capabilities and network connectivity, and without relying on a central master server. A device transparent storage system must track object updates, forward changes to running applications, handle concurrent updates, and proactively partition and replicate data across heterogeneous devices. Eyo addresses these requirements by separating objects' metadata from their content, and distributing all metadata to all devices.

UIA appeared at IPTPS 2006 [6] and OSDI 2006 [5]. Eyo appeared at HotStorage 2009 [2].

Alpaca — Extensible Proof-Carrying Authorization 2005–2008

Alpaca is a logic-based Proof-Carrying Authorization framework. It provides an API enabling network applications to state and prove logical assertions such as "the principal Alice says to delete the file X" using cryptographic operations specified in the accompanying proof. Since verifiers don't care how the proof is structured, as long as it is valid, this permits provers to use different cryptographic techniques (such as new hash functions or data transport mechanisms) without breaking compatibility with existing verifiers. Alpaca's flexibility stands in contrast to cryptographic protocols such as Kerberos and TLS, which can only be updated by installing new software. Crucially, Alpaca extensions do not need to be approved by any central authority: any user can unilaterally deploy any extension as long as it produces the correct type of proofs. Extensions preserve security because they do not expand users' privileges, they simply enable users to apply their existing privileges in novel ways.

This work appeared at CCS 2007 [4].

Barnraising and SSL Splitting — Untrusted CDN 2002–2003

Barnraising is a P2P content distribution network (CDN) which, like the later system CoralCDN, enables Web sites to delegate some of their load to a distributed network of cooperating cache hosts. Unlike CoralCDN, Barnraising uses a novel technique called *SSL Splitting* to securely serve data using untrusted caches. Because a malicious cache cannot send clients bogus data, Barnraising can safely permit any Internet host to contribute cache space; on the other hand, CoralCDN is limited to the resources available from the centrally-controlled (and under-provisioned) PlanetLab.

The SSL Splitting library is installed on a Web server as a drop-in replacement for the popular OpenSSL library, enabling the server to communicate with the untrusted Barnraising cache nodes. SSL Splitting does not require any changes to Web clients.

This work appeared at USENIX Security 2003 [9] and in *Computer Networks,* August 2005 [8].