

PIERRE-MALO DENIÉLOU

Address: 30 rue Carnot
26500 Bourg-lès-Valence
FRANCE
Cell-phone: + 33.6.71.73.18.17

E-mail: pierre-malo.deniélou@inria.fr
Webpage: <http://moscova.inria.fr/~denielou/>
French Citizen
26 years old

RESEARCH INTERESTS

Programming languages, semantics, type systems, concurrency theory, language-based security, mechanized proofs.

EDUCATION

2005-2009 PhD thesis supervised by Jean-Jacques Lévy and James J. Leifer in the Moscova team at INRIA Rocquencourt. Joined later the MSR-INRIA Joint Centre to also work with Ricardo Corin, Cédric Fournet and Karthik Bhargavan [1, 2, 3]. Provisional title is *Safety and security in distributed languages*. Defense expected: March 2009.

2004-2005 Passed with honors a Master of Science (called *Master Parisien de Recherche en Informatique – MPRI*) in theoretical computer science in Paris. This Master is organized by ENS, ENS Cachan, École Polytechnique and Université Paris VII. Rank : 2nd/44.

2002-2004 Computer Science studies at ENS Cachan (Prof. Hubert Comon).

1999-2002 Studied 3 years Mathematics and Physics in *Classes préparatoires aux Grandes Écoles* at *Lycée du Parc* in Lyon.

1999 Passed with honors a scientific “Baccalauréat” in mathematics, physics and biology.

INTERNSHIPS

May-June/2008 Research internship at Microsoft Research in Cambridge, UK. Work with Cédric Fournet and Karthik Bhargavan. *Cryptographic Protocol Synthesis and Verification for Multiparty Sessions* [2]

June-August/2006 Research internship at Microsoft Research in Cambridge, UK. Work with Cédric Fournet and Karthik Bhargavan. *Secure Implementations of Typed Session Abstractions* [1]

March-August/2005 Master’s internship with Jean-Jacques Lévy and James Leifer at INRIA Rocquencourt in the Moscova team. *Abstraction preservation and subtyping in distributed languages* [4].

March-August/2004 Research internship in Philadelphia, Pennsylvania. Work with Prof. Benjamin C. Pierce, University of Pennsylvania, on the *Unison* and *Harmony* projects.

June-July/2003 Research internship in the IRIT institute in Toulouse with Philippe Besnard. *Combining Logics*.

WORK EXPERIENCE

2005-2008	Teaching assistant at Université Paris VII Denis Diderot.
2002-2006	Work as a “fonctionnaire stagiaire - élève normalien” for the French State.

LANGUAGES

English	Fluent.
Programming	Ocaml, F#, Java, C, (X)HTML, Coq.

HOBBIES

Computer Science	Programming. Webmastering.
Sport	Table tennis and walks in the Alps.
Music	Clarinet.
Charity	Helping young teenagers with their studies.

RESEARCH PAPERS

[2] Cryptographic Protocol Synthesis and Verification for Multiparty Sessions

(with Karthikeyan Bhargavan, Ricardo Corin, Cédric Fournet and James J. Leifer)

Submitted for publication, October 2008. <http://www.msr-inria.inria.fr/projects/sec/sessions/>

We present a compiler for generating custom cryptographic protocols from high-level multiparty sessions. Sessions specify pre-arranged patterns of message exchanges between distributed participants and their data accesses to a shared store. We define integrity and confidentiality properties of sessions, in a setting where the network and arbitrary compromised parties may be controlled by an adversary. Our compiler enforces the security properties by guarding the sending and receiving of session messages by optimized cryptographic operations and checks. Given a session, our compiler generates ML modules and interfaces that can be linked to application code for each party. We prove that the generated code is secure by relying on a recent refinement type system for ML. The send and receive functions in the module interface have dependent types that express invariants of the session state. We type-check the program against this interface, and complete the proof by a brief, hand-crafted argument. We illustrate and evaluate our implementation on a series of typical protocols, inspired by web services. In comparison with prior work, our source language is more expressive, our implementation more efficient, and our proof technique novel. Most of the proof is performed by mechanised type checking of the generated code, and does not rely on the correctness of our compiler. We obtain the strongest session security guarantees to date in a model that accounts for the actual details of protocol code.

[1] Secure Implementations for Typed Session Abstractions

(with Karthikeyan Bhargavan, Ricardo Corin, Cédric Fournet and James J. Leifer)

In *20th IEEE Computer Security Foundations Symposium (CSF'07)*, Venice, Italy, July 2007.

Distributed applications can be structured as parties that exchange messages according to some pre-arranged communication patterns. These sessions (or contracts, or protocols) simplify distributed programming: when coding a role for a given session, each party just has to follow the intended message flow, under the assumption that the other parties are also compliant. In an adversarial setting, remote parties may not be trusted to play their role. Hence, defensive implementations also have to monitor one another, in order to detect any deviation from the assigned roles of a session. This task involves low-level coding below session abstractions, thus giving up most of their benefits. We explore language-based support for sessions. We extend the ML language with session types that express flows of messages between roles, such that well-typed programs always play their roles. We compile session type declarations to cryptographic communication protocols that can shield programs from any low-level attempt by coalitions of remote peers to deviate from their roles.

[3] A protocol compiler for secure sessions in ML

(with Ricardo Corin)

In G. Barthe and C. Fournet, editors, *Third Symposium on Trustworthy Global Computing (TGC'07)*, Sophia Antipolis, France, Lecture Notes in Computer Science. Springer Verlag, November 2007.

Distributed applications can be structured using sessions that specify flows of messages between roles. We design a small specific language to declare sessions. We then build a compiler, called `s2ml`, that transforms these declarations down to ML modules securely implementing the sessions. Every run of a well-typed program executing a session through its generated module is guaranteed to follow the session specification, despite any low-level attempt by coalitions of remote peers to deviate from their roles. We detail the inner workings of our compiler, along with our design choices, and illustrate the usage of `s2ml` with two examples: a simple remote procedure call session, and a complex session for a conference management system.

[4] Abstraction Preservation and Subtyping in Distributed Languages

(with James J. Leifer)

In *11th International Conference on Functional Programming (ICFP)*, October 2006.

In most programming languages, type abstraction is guaranteed by syntactic scoping in a single program, but is not preserved by marshalling during distributed communication. A solution is to generate hash types at compile time that consist of a fingerprint of the source code implementing the data type. These hash types can be tupled with a marshalled value and compared efficiently at unmarshall time to guarantee abstraction safety. In this paper, we extend a core calculus of ML-like modules, functions, distributed communication, and hash types, to integrate structural subtyping, user-declared subtyping between abstract types, and bounded existential types. Our semantics makes two contributions: (1) the explicit tracking of the interaction between abstraction boundaries and subtyping; (2) support for user-declared module upgrades with propagation of the resulting subhashing relation throughout the network during communication. We prove type preservation, progress, determinacy, and erasure for our system.