Logical Foundations of Computer Security

Bob Constable
Cornell University, Ithaca, USA

The first of my four lectures on the logical foundations of computer security will establish the mathematical setting in which we will work, namely computational type theory. In this setting we can pursue algorithmic ideas from their natural expression in computer science articles to their codification in executable systems and finally to their incorporation into formal arguments that a system achieves some stated purpose. For this lecture we draw on material produced in previous summer school lectures, in particular on the booklet *Naive Computational Type Theory* from Marktoberdorf 2001. Another modern source of this material is the recent monograph by M. Sorensen and P. Urzyczyn, see [7]. The Nuprl version of computational type theory is presented in the article by Allen, Bickford et. al, see [2].

The second lecture will present an abstract theory of events which my colleague Mark Bickford and I created in 2003 for specifying distributed computing problems. This theory has been implemented in the Nuprl version of Computational Type Theory (CTT) by Mark. The theory is based on the causal order relation used by Leslie Lamport in this early work on the foundations of distributed computing. One good reference for this material are the MOD05 lectures notes by Mark Bickford and me, see [4]. The material was also presented in the introduction to the paper [5].

The third lecture introduces the formalism of message automata. This is a programming language for distributed processes running on a network. It is the formulation of the standard model of asynchronous message passing computation used in textbooks such as [3]. The lecture includes examples of simple distributed systems.

The fourth lecture considers the problem of building a secure distributed secret service and offers a mechanism from type theory that makes this formally tractable. The mechanism from type theory is the implementation of atomic tokens and the rules for them that preclude assembling these tokens from smaller elements or hiding them in the computable terms of type theory. These tokens cannot be manufactured except by primitive mechanisms of an implementation of type theory. So they cannot be created by computation. This makes it possible to use these tokens to encode secrets and trace their flow through a computation. This is a new approach to secret sharing which has a very simple logical foundation. The basic ideas and a simple application will be covered in this lecture. The foundational facts are given in the article by Stuart Allen, see [1].

References

1. Stuart F. Allen. *An Abstract Semantics for Atoms in Nuprl*, Cornell Tech. Report TR2006-2032.
2. S.F. Allen, M. Bickford, R.L. Constable, R. Eaton, C. Kreitz, L. Lorigo, E. Moran. *Innovations in computational type theory using Nuprl*, Journal of Applied Logic, 4, pp. 428–469, 2006.
3. H. Attiya and J. Welsch. *Distributed Computing*, Addison Wesley, 2005.
4. Mark Bickford and Robert Constable. *A Causal Logic of Events in Formalized Computational Type Theory*, Proc. of the Marktoberdorf NATO Summer School, 2005.
5. Mark Bickford, Robert Constable, Joseph Halpern, Sabina Petride. *Knowledge-Based Synthesis of Distributed Systems Using Event Structures in Proceeding of Logic for Programming*, Artificial Intelligence, and Reasoning, LNCS 3452, Springer, NY, pp. 449–465, 2005.
6. Robert Constable. *Naive Computational Type Theory, Proof and System Reliability*, H. Schwichtenberg and R. Steinbrüggen (eds.), pp. 213-259, Springer, 2002.
7. M. Sorensen and P. Urzyczyn. *Lectures on the Curry-Howard Isomorphism*, Elsevier, 2006.