Verifying Electronic Commerce Protocols

Lawrence C. Paulson

Security protocols prevent network communications from being compromised by adversaries. Those designed for electronic commerce have many differences from traditional security protocols. They must address the issue of whether other parties to the transaction can be trusted.

SET is an extremely complicated protocol suite designed for on-line shopping. It addresses the issue of trust through its *registration* process, a hurdle that gives successful candidates credentials serving as evidence that they are trustworthy.

This project is a continuation of EPSRC project GR/K77051, which introduced the inductive method for analyzing security protocols [7]. This method works by specifying the behaviour (operational semantics) of the system, including the adversary. Properties are proved using Isabelle [6], an interactive proof assistant.

We (Bella, Massacci, Paulson) have achieved our goals: we have formally analysed the main part of the SET protocols as well as two other electronic commerce protocols. We were able to verify the main claims about each protocol, and we identified a number of minor weaknesses.

The most obvious outcome of this project is that even the most complicated protocols are subject to formal anslysis. In the mid-1990s, the methods for verifying security protocols were promising but immature. Since that time, methods based on operational semantics have achieved great success. Both model-checking, pioneered by Lowe [5], and theorem proving are valuable. Among the protocols analysed during the previous Cambridge project was TLS, a descendant of SSL [8]. With the analysis of SET, all questions about "scaling up" can be swept aside.

We analysed two additional protocols (non-repudiation [4] and certified e-mail [1]), both designed for an environment where the two main participants do not trust one another. Both protocols are designed to be *fair*: either both parties receive certain information, or neither does. We found that our existing Isabelle protocol theories could model these protocols with minimal changes, and we were able to prove a collection of guarantees that a party could rely upon whether or not the opposite party was honest.

The project has yielded many publications [1, 2, 3, 4]. Tnese are available from the Web page .[1] The full proof scripts will be included in the next release of Isabelle.

[1] http://www.cl.cam.ac.uk/users/lcp/papers/protocols.html

References

[1] Giampaolo Bella, Cristiano Longo, and Lawrence C. Paulson. Verifying second-level security protocols. In David Basin and Burkhart Wolff, editors, *Theorem Proving in Higher Order Logics: TPHOLs 2003*, LNCS 2758, pages 352–366. Springer, 2003. Online at `http://link.springer.de/link/service/series/0558/tocs/t2758.htm`.

[2] Giampaolo Bella, Fabio Massacci, and Lawrence C. Paulson. The verification of an industrial payment protocol: The SET purchase phase. In Vijay Atluri, editor, *9th ACM Conference on Computer and Communications Security*, pages 12–20. ACM Press, 2002.

[3] Giampaolo Bella, Fabio Massacci, and Lawrence C. Paulson. Verifying the SET registration protocols. *IEEE Journal on Selected Areas in Communications*, 21(1):77–87, 2003.

[4] Giampaolo Bella and Lawrence C. Paulson. Mechanical proofs about a non-repudiation protocol. In Richard J. Boulton and Paul B. Jackson, editors, *Theorem Proving in Higher Order Logics: TPHOLs 2001*, LNCS 2152, pages 91–104. Springer, 2001. Online at `http://link.springer.de/link/service/series/0558/tocs/t2152.htm`.

[5] Gavin Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using CSP and FDR. In T. Margaria and B. Steffen, editors, *Tools and Algorithms for the Construction and Analysis of Systems: second international workshop, TACAS '96*, LNCS 1055, pages 147–166. Springer, 1996.

[6] Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*. Springer, 2002. LNCS Tutorial 2283.

[7] Lawrence C. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 6:85–128, 1998.

[8] Lawrence C. Paulson. Inductive analysis of the Internet protocol TLS. *ACM Transactions on Information and System Security*, 2(3):332–351, August 1999.