Robbing the Bank with a Theorem Prover
(Abstract)

Paul Youn[1], Ben Adida[2], Mike Bond[3], Jolyon Clulow[3], Jonathan Herzog[4], Amerson Lin[5], Ronald L. Rivest[5], and Ross Anderson[3]

[1] Oracle Corporation
[2] Center for Research on Computation and Society, Harvard University
[3] Computer Laboratory, University of Cambridge
[4] Naval Postgraduate School, Monterey CA
[5] Computer Science and Artificial Intelligence Laboratory,
Massachusetts Institute of Technology

In this work, we present the first automated analysis of security application programming interfaces (security APIs). In particular, we analyze the API of the IBM 4758 CCA, a hardware security module for banking networks. Adapting techniques from formal analyses of security protocols, we model the API purely according its specification and assuming ideal encryption primitives. We then use the automated theorem-prover Otter to analyze this model, combining its standard reasoning strategies with novel techniques of our own (also presented here). In this way, we derive not only all published API-level attacks against the 4758 CCA, but an extension to these attacks as well. Thus, this work represents the first step toward fully-automated, rigorous analyses of security APIs.

Our main contribution to the analysis of security APIs is thus three-fold:

1. We provide what we believe to be the first application of formal automated reasoning techniques to the problem of API security.
2. We define a modeling methodology which greatly speeds up the analysis of APIs, thus making such analysis practical.
3. We demonstrate the ability of the tool to discover complicated sequences of API calls that violate security properties.

The work presented here is described in detail in University of Cambridge Computer Laboratory Technical Report number 644, which is available on-line at: `http://www.cl.cam.ac.uk/TechReports/UCAM-CL-TR-644.pdf`.

B. Christianson et al. (Eds.): Security Protocols 2007, LNCS 5964, p. 171, 2010.