CAPSL: Common Authentication Protocol Specification Language

Jonathan K. Millen
The MITRE Corporation
Bedford, MA 01730
jkm@mitre.org

CAPSL is a formal language for expressing authentication and key-exchange protocols. It is intended to capture enough of the abstract features of these protocols to perform an analysis for protocol failures. The impetus for such a language grew out of project work in protocol analysis.

A common protocol specification language seems necessary to bridge the gap between the typical informal presentations of protocols given in papers and the precise characterizations required to conduct formal analysis. It is hoped that proponents of different analysis techniques will offer algorithms for compiling this language into whatever form they require. Doing so will go a long way toward ensuring that the assumptions made by different techniques, as well as the analysis results, are comparable.

Since Denning and Sacco published a replay attack on the Needham-Schroeder protocol in 1981, it has been well known that protocols for exchanging cryptographic keys over data networks can be vulnerable to message modification attacks. The abundance of flaws in published protocols led to the development of formal techniques for their security analysis. The proposed techniques, as represented by some of the earlier papers on the subject, include the use of goal-directed state search tools implemented in Prolog, the application of general purpose specification and verification tools, a specially-designed logic of belief, and the application of a model-checking tool for CSP specifications.

It has become evident that it was difficult for analysts other than the developers of the various techniques to apply them. One reason for this difficulty is the fact that the protocols had to be re-specified for each technique, and it was not easy to transform the published description of the protocol into the required formal system. Some tool developers began work on translators or compilers that would perform the transformation automatically. The input to any such translator still requires a formally-defined language, but it can be made similar to the message-oriented protocol descriptions that are typically published. Besides our initial work on CAPSL for the Interrogator at MITRE, there were independent efforts by Steve Brackin, with a language ISL, from which CAPSL borrowed much of its style, supporting an application of HOL to an extension of the GNY logic; and Gavin Lowe, with a similar language, CASPER, for the application of FDR using a CSP model-checking approach. The idea of having a single common protocol specification language that could be used as the input format for any formal analysis technique was first presented at the 1996 Isaac Newton Institute Programme on Computer Security, Cryptology, and Coding Theory.

The design of CAPSL is still in progress. Current documentation for the language, and discussions on design alternatives and extensions, may be found at the CAPSL home page on the World-Wide Web, at the URL http://www.mitre.org/research/capsl.