ARM Verification (Abstract)

Joe Hurd, Anthony Fox, Mike Gordon and Konrad Slind
joe.hurd@comlab.ox.ac.uk, anthony.fox@cl.cam.ac.uk,
mjcg@cl.cam.ac.uk and slind@cs.utah.edu

The cryptographic modules of a system are usually critical to maintaining its security policy, and it is thus desirable to have a guarantee that the hardware and software implementations of these modules are correct before the system is fielded. This is the motivation driving the project *Verified Implementation of Cryptographic Algorithms*, a joint effort by the University of Cambridge, University of Oxford and University of Utah.

The project plan is to start from a mathematical specification of elliptic curve cryptography, and deploy a verifying compiler to produce an ARM machine code implementation together with a proof that its behaviour satisfies the mathematical specification. Here is the verification flow in more detail, annotated with the names and institutions of the researchers mainly responsible for each step:

- A formal specification of elliptic curve operations derived from mathematics (Hurd, Oxford).
- A verifying compiler from higher order logic functions to a low level assembly language (Slind & Li, Utah).
- A verifying back-end targeting ARM assembly programs (Tuerk, Cambridge).
- An assertion language for ARM assembly programs (Myreen, Cambridge).
- A very high fidelity model of the ARM instruction set derived from a processor model (Fox, Cambridge).

The whole verification takes place in the HOL4 theorem prover [3], a verification environment emphasizing soundness used by many formal methods researchers over the past twenty years. Thus the main assumptions in the final correctness result are:

1. **Specification:** The formalized theory of elliptic curve cryptography is faithful to standard mathematics.
2. **Model:** The formalized ARM machine code is faithful to the real world execution environment.

The formalization of elliptic curves in higher order logic follows the source textbook *Elliptic Curves in Cryptography* [1]. To justify its use as a specification in this project, three methods are used to gather evidence of its correctness: comparing the formalized version to a standard mathematics textbook; deducing properties known to be true of elliptic curves; and deriving checkable calculations for example curves. It is being actively developed, and there is a report available describing its design [4].

The ARM model is a formalization of the Instruction Set Architecture (ISA) of the processor in higher order logic. It derives from a previous project to formally verify the correspondance between the ARM6 microarchitecture and ISA in higher order logic using the HOL4 theorem prover [2]. The successful completion of this verification is the main evidence that the formalized ARM model is faithful to the real world execution environment. To reason about a wider class of ARM codes, the formalized ISA model has recently been extended from ARMv3 to ARMv4 (with the intention to move to ARMv5). The ARM model can be used for testing, by extracting a processor simulator in the ML programming language. The compiled simulator executes at the rate of 10,000 instructions per second.

The verifying compiler takes as input a program implemented in the functional programming language of higher order logic, and outputs ARM machine code. In addition, it generates a HOL4 theorem that the ARM machine code has the same behaviour as the input program. At the present state of the project results are available compiling test implementations of finite field operations to ARM machine code. The compositional format (due to Myreen) of the verifying compiler theorems allows external code to be verified separately and linked to the compiled code as a verified module.

References

[1] Ian Blake, Gadiel Seroussi, and Nigel Smart. *Elliptic Curves in Cryptography*, volume 265 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 1999.

[2] Anthony Fox. Formal specification and verification of ARM6. In David Basin and Burkhart Wolff, editors, *16th International Conference on Theorem Proving in Higher Order Logics: TPHOLs 2003*, volume 2758 of *Lecture Notes in Computer Science*, pages 25–40, Rome, Italy, September 2003. Springer.

[3] M. J. C. Gordon and T. F. Melham, editors. *Introduction to HOL (A theorem-proving environment for higher order logic)*. Cambridge University Press, 1993.

[4] Joe Hurd. Formalizing elliptic curve cryptography in higher order logic. Available from the author's web site, October 2005.