Cryptol: The Language of ~~Cryptography~~ Cryptanalysis

Sally A. Browning
sally@galois.com

Joe Hurd
joe@galois.com

Galois, Inc
www.galois.com
www.cryptol.net

Introduction

Cryptol was designed by Galois, Inc. for the NSA as a domain-specific language for specifying cryptographic algorithms, eliminating the need for separate and voluminous natural language documentation. Cryptol is tailored to the unique needs of cryptography and cryptographic implementations. It is fully executable, allowing cryptographers to experiment with their programs incrementally as their designs evolve, with the compiler checking the consistency of data types and array lengths at every stage. These same attributes make Cryptol a good language for expressing cryptanalysis algorithms, providing a platform to explore different approaches and carry out experiments at low cost.

In addition, Cryptol provides a refinement methodology to bridge the conceptual gap between specification and low-level implementation, and can generate both hardware and software implementations from high-level specifications, as well as formal models for verification. For example, Cryptol allows engineers and mathematicians to program cryptographic algorithms on FPGAs as if they were writing software, and the Cryptol verification toolset can show functional equivalence between the specification and the implementation at each stage of the tool-chain. In addition, the Cryptol verification toolset can be usefully applied to the reference specification of cryptographic algorithms. Proving desirable high-level properties of a cryptographic algorithm gives assurance of its robustness, while conversely finding counter-examples of desirable properties may inspire approaches to cryptanalysis.

Cryptol Case Studies

- *Exploring an algorithm.* The AIM crypto-engine engineers at General Dynamics C4 Systems use the Cryptol modeling language as part of the development process. Cryptol provides four basic benefits leading to the certification of crypto equipment. First, Cryptol allows the design engineer to rapidly express an algorithm in a common mathematical notation, which is fully executable on the Cryptol interpreter, providing verification that the algorithm is completely understood. Second, the Cryptol notation for the various components of the algorithm are used to annotate the AIM micro sequencer code which provides much greater readability of that extremely dense assembly language. Third, component testing of AIM code, from small snippets through major subroutines is greatly facilitated with Cryptol generated test vectors derived from end-to-end test vectors provided in algorithm source specifications. Finally, Cryptol models directly support the certification effort.

- *Produce and refine a family of designs.* A team of developers from Rockwell Collins, Inc. and Galois, Inc. has successfully designed, implemented, simulated, integrated, analyzed, and tested a complex embedded Cryptographic Equipment Application (CEA) in less than 3 months. An algorithm core generated from a Cryptol specification for AES-256 running in Electronic Codebook mode demonstrated throughput in excess of 16 Gbps. These high-speed CEA implementations comprise a mixture of software and VHDL, and target a compact new embedded platform designed by Rockwell Collins. Notably, almost no traditional low-level interface code was required in order to implement these high-performance CEAs. In addition, automated formal methods prove that algorithm implementations faithfully implement their high-level specifications.

- *Gaining confidence in an implementation*. Van der Waerden's theorem states that for any positive integers *r* and *k* there exists a positive integer *N* such that if the integers *{1, 2, ..., N}* are each colored with one of *r* different colors, then there are at least *k* integers in arithmetic progression all of the same color. For any *r* and *k*, the smallest such *N* is the van der Waerden number *W(r,k)*. Van de Waerden numbers are difficult to compute. In 2007, Dr. Michal Kouril of the University of Cincinnati established that *W(2,6)=1132* (i.e., 1132 is the smallest integer *N* such that every 2-coloring of *{1, 2, ..., N}* contains a monochromatic arithmetic progression of length 6). The most recent previous result, *W(2,5)=178*, was discovered some 30 years earlier. Kouril computed *W(2,6)* using a special SAT-solver and clever techniques to bound the search and employed FPGAs to speed up the search. In order to convince himself that the FPGA ensemble was doing what he expected, he wrote a Cryptol specification for the algorithm running in the FPGA ensemble, generated formal models for both the Cryptol specification and the VHDL implementation, and verified that the two were equivalent.

- *Gaining confidence in a third-party implementation*. Skein is a suite of cryptographic hash algorithms targeted at the NIST SHA-3 competition. At its core, Skein uses a tweakable block cipher named Threefish. The unique block iteration (UBI) chaining mode defines the mode of operation by the repeated application of the block cipher function. Galois has developed and published a Cryptol specification for Skein, and verified two independently developed VHDL implementations of Skein against our specification, finding an ambiguity bug in one of them.

- *Building a MILS FPGA*. The Xilinx Single Chip Cryptographic (SCC) technology enables Multiple Independent Levels of Security (MILS) on a single FPGA. Galois' Cryptol Workbench provides a tool flow that puts FPGA implementation into the hands of mainline developers, improving both productivity and assurance, without sacrificing performance. These two technologies fit seamlessly into a single development flow. The combined solution can address high-grade cryptographic application requirements (redundancy, performance, red/black data, and multiple levels of security on a single chip) as well as high assurance development needs (high-level designs, automatic generation of implementation from design, automatically-generated equivalence evidence), and has the potential to significantly reduce the time of costs of developing Type-1 cryptographic applications.

Try Cryptol for your Applications

The Cryptol interpreter is freely available at `www.cryptol.net`. Documentation and evaluation copies of the full Cryptol toolset are also available at that site.