TEARSHEET

| galois |

Cryptol Workbench

High assurance, high performance, high level design

Empower the experts

A *domain-specific language* (DSL) is a programming language targeted at producing solutions in a given problem domain by empowering subject-matter experts to design solutions in terms they are familiar with and at a level of abstraction that makes most sense to them. In addition, a good DSL opens the way for powerful tool support: simulations for design exploration, automatic testing and automatic generation of test harnesses, generation of highly specialized code for multiple targets, and generation of formal evidence for correctness, safety and security properties.

Galois has designed successful DSLs for a variety of domains, including cryptography, operating system security policy, communication routing policy, and more.

Cryptol: the language of cryptography

Cryptol is a domain specific language designed by Galois for the National Security Agency as a standard for specifying cryptographic algorithms. The Cryptol workbench facilitates the deployment of cryptographic modules across the entire software process, from specification and implementation to verification and certification. Using the Cryptol Workbench can significantly reduce the overall lifecycle costs by addressing the key cost drivers in the deployment of crypto:

Rapid design cycle

Cryptol specifications are fully executable, allowing designers to experiment with their programs incrementally as their designs evolve. The Cryptol tools support a refinement methodology that bridges the conceptual gap between specification and low-level implementation, thereby reducing time-to-market. For example, Cryptol allows engineers and mathematicians to program cryptographic algorithms on FPGAs *as if they were writing software.*

Reuseable Specification

Cryptol Studio provides a platform-neutral specification language that generates or guides implementations on multiple platforms. The Cryptol tools can generate C, C++, and Haskell software implementations, VHDL hardware implementations, and formal models for verification from a single specification.

Accelerated certification

A Cryptol reference specification becomes the formal documentation for the cryptographic module, eliminating the need for separate and voluminous English descriptions. In addition, Cryptol verification tools show functional equivalence between the specification and the implementation at each stage of the toolchain.

The Cryptol Workbench

Available for Windows, Mac OS X, and Linux systems.

Component	Description
Cryptol interpreter	Executes Cryptol programs interactively
QuickCheck	Generates random test vectors
FPGA generator	Translates Cryptol to VHDL
SBV generator	Inlined C, C++, Haskell, and Isabelle code Formal models for SMT solvers
C generator	Compiles Cryptol to modular C code
Isabelle generator	Generates formal specifications for the Isabelle theorem prover
Symbolic evaluatior	Produces a formal model from a Cryptol specification
Symbolic simulator	Produces formal model from VHDL via FPGA vendor tools
Jaig	Statically compares two formal models for functional equivalene

Galois, Inc.
www.galois.com
421 SW 6th Ave., Suite 300
Portland, Oregon 97204
503/626-6616

Cryptol Workbench in Action

Cryptol Workbench Use Case

A team of developers from Rockwell Collins, Inc. and Galois, Inc. has successfully produced high-speed embedded Cryptographic Equipment Applications (CEAs), automatically generated from high-level specifications. An algorithm core generated from a Cryptol specification for AES-256 and running in Electronic Codebook mode demonstrated throughput *in excess of 16 Gbps*.

The "crypto waveform" logic uses the Model-Based Development language Simulink. 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* to implement these high-performance CEAs.

Automated formal methods based tools provided by Cryptol proved that algorithm implementations faithfully implement their high-level specifications.

When feedback from the output stage to the input was introduced, thereby defeating the advantage gained by "unrolling" AES rounds, encryption performance for AES-256 still *exceeded 1 Gbps while consuming less than 2% of the available programmable logic* for the algorithm core.

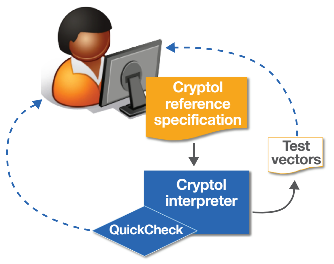
Most importantly, the Rockwell Collins/Galois team was able to design, implement, simulate, integrate, analyze, and test a complex CEA on the new hardware, including AES-256 and Galois Counter Mode (GCM), in *less than 3 months*.

.

Get in on the action!

At www.cryptol.net:

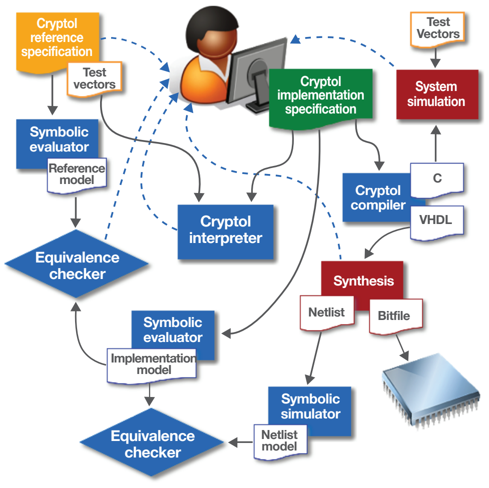
- Read a whitepaper describing the Rockwell Collins project.
- Download the freely-available Cryptol interpreter.
- Register to evaluate the Cryptol Workbench.



Cryptographer,

creating a new cryptographic algorithm.

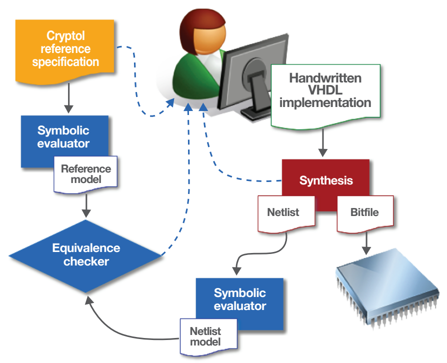
- Create a Cryptol reference specification.
- Execute the specification, including assertion checks.
- Generate test vectors with Quickcheck to bundle with the reference specification.



Crypto vendor,

designing a new crypto product.

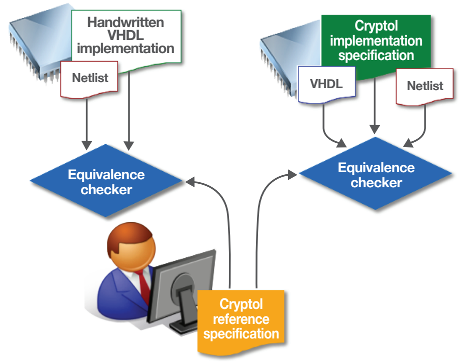
- Quickly and easily refine the customer-supplied Cryptol reference specification to a family of implementations, trading off space, time, and other performance metrics.
- Compile the implementation for multiple targets: C/C++, Haskell, and VHDL are currently supported.
- Equivalence check an implementation against the reference specification. VHDL implementations are converted into bitfiles for FPGAs using third-party FPGA vendor tools; verification tools can be utilized at several points in that vendor tool-chain.



Hardware designer,

creating high-assurance crypto hardware.

- Run the interpreter to generate test vectors from the reference specification for testing your VHDL in a simulator.
- Equivalence check the implementation models against the reference model at several points in the synthesis tool chain.


Handwritten VHDL implementation
Netlist
Cryptol implementation specification
VHDL
Netlist
Equivalence checker
Equivalence checker
Cryptol reference specification

Crypto device evaluator,

verifying and certifying a crypto device.

- Create a reference specification and associated formal model.
- Equivalence check your reference spec against the vendor-written one, if it is different.
- Equivalence check the implementation models, at several points in the tool chain, against the reference specification.

🟦 Galois tools
🟥 Third party tools
→ Input to tool
- - → Input / Feedback to designer

galois
421 SW 6th Ave., Suite 300
Portland, Oregon 97204
503/626-6616
www.galois.com