Functional Correctness: C code

Description Illustrate how SAW and Cryptol are used to verify the correctness of C functions. Functional correctness is concerned with whether the C function adheres to its functional specification. In this case the specification is written in Cryptol.

Purpose Getting familiar with how to use SAW to get Cryptol to cooperate with C code to show functional equivalence.

Audience This module is intended for:

1 The general public
2 K-12 and college classes on Cyber Defense and Math Logic
3 preparation for proficiency in the use of tools and a computing environment suitable for the study of cyber defense

Objectives After completing the module:

1 You will know how to write a setup function in SAW that verifies C code against a Cryptol specification.
2 Many C functions have pointer inputs and outputs. Cryptol does not have pointers. You will learn about SAW functions that are able to deal with this dichotomy.
3 You will learn to reuse several SAW functions to build setup files for different verification examples.

Keywords Cryptol, SAW, Yices, ABC, Z3, CVC5, Boolector, stdint.h, primitive data types, verification, functional correctness

Category cybersecurity > education

Delivery java applets and written documentation in pdf format

Team John Franco and Ethan Link

Assessment The applets provide the means for experimentation. Questions are asked in the documentation that help with the set up of experiments. The ideas that learners come up with is evidence that the module was successful.

Workflow No particular schedule was established

Environment All materials are contained in a single jar file. The jar file can be run on any computer where java version 14 or higher and some pdf reader such as acroread or evince are available. The jar file may be executed in the cyber range or learners may download the jar file (which is considered to be an executable file) and run it on their personal computers.