Practical and Scalable Security Verification of Security-Aware Hardware Architectures

Prof. Jakub Szefer, W. Xiong, and S. Deng
Yale University (**NSF Grant #1524680**)

Prof. Onur Demir and D. Gümüşoğlu
Yeditepe University, Turkey

Prof. Ruby B. Lee and T. Zhang
Princeton University (**NSF Grant #1526493**)

Towards design of **SecChisel**, a hardware security verification toolset

Due to lack of practical and scalable security verification tools and methodologies, very few of the proposed hardware security architectures have been thoroughly checked at the design time. To address the issue, this project develops a security verification methodology that is applicable to different hardware security architectures during design time.

Hardware Security Verification

- Prove that system holds desired properties with respect to not just functionality but also security
- Focus on information flow and non-interference
  - Can reason about confidentiality and integrity using these properties

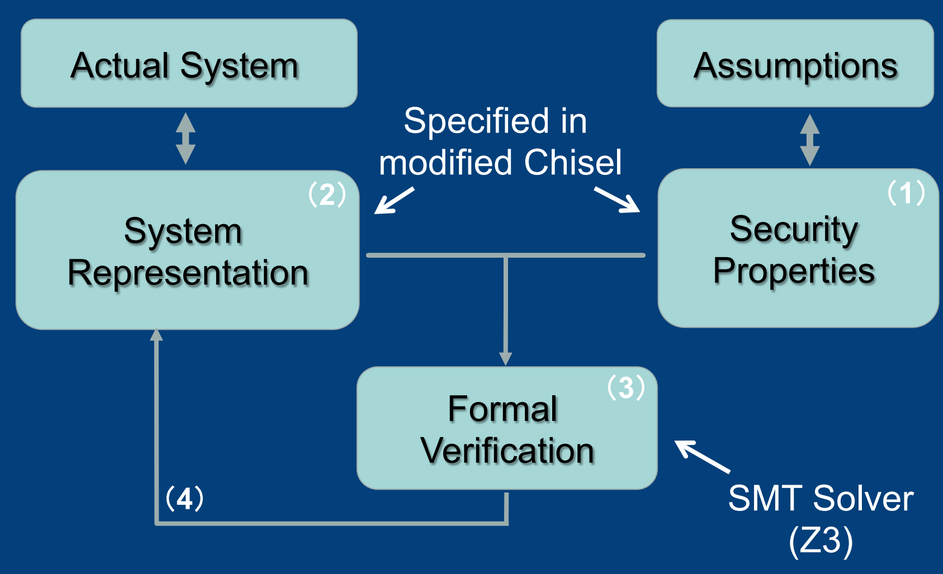
Hardware Verification at Design Time

- Hardware is almost impossible to patch once it is fabricated, has to be secure from the start

Verification Scope

- Focus on checking hardware architecture design
- Manufacturing problems are not in scope

Security Verification Methodology



Security verification process: considers security properties (1), and system representation (2), verifies the design (3), and provides feedback to fix any issues (4)

Approach for design of SecChisel

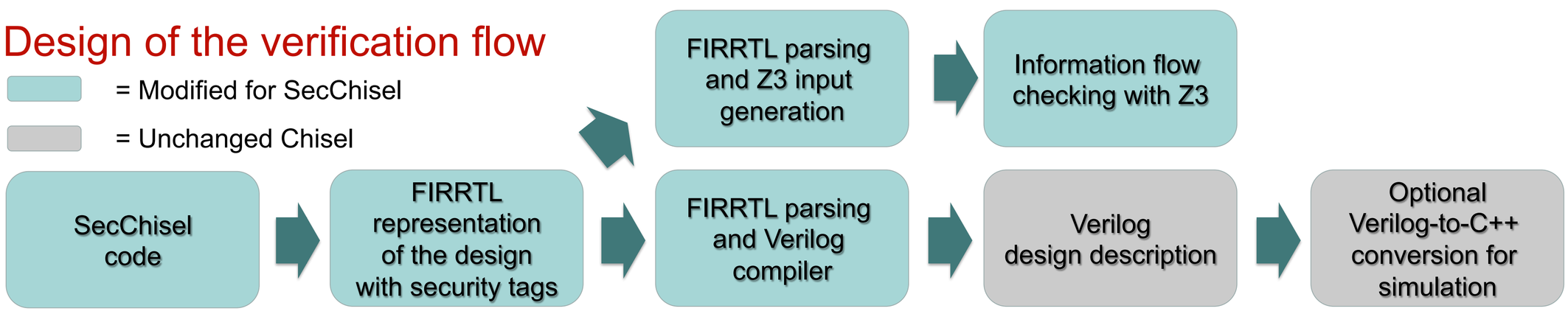
Modified Chisel language & tools

- The base for the project is the existing Chisel tool developed at Berkeley
- It is used for hardware construction
- Extend data types with security tags and use lattice model for the tags

Security checking using Z3

- Focus on using model checkers to check security properties
- Connect Chisel output (modified FIRRTL) to Z3 SMT solver
- Check for information flow based on security tags

Design of the verification flow



Progress on SecChisel

- Modified Chisel 3 to add static tags for wires, registers, etc.
- Pass tags to FIRRTL, and then generate Z3 input from the modified FIRRTL
- Information flow checks in Z3

Future work and research targets

- Static and dynamic security tag development and checking
- Verification of advanced designs: secure caches and processor pipelines
- Consider modular & scalable verification

Interested in meeting the PIs? Attach post-it note below!

National Science Foundation
WHERE DISCOVERIES BEGIN

NSF Secure and Trustworthy Cyberspace Principal Investigator Meeting
January 9-11, 2017
Arlington, Virginia