C-FLAT: Control Flow Attestation for Embedded Systems Software

Thomas Nyman, Andrew Paverd, Jan-Erik Ekberg, N. Asokan
Tigist Abera, Lucas Davi, Ahmad-Reza Sadeghi, Gene Tsudik

How can a trusted verifier learn about run-time attacks and the dynamic behavior of an embedded device?

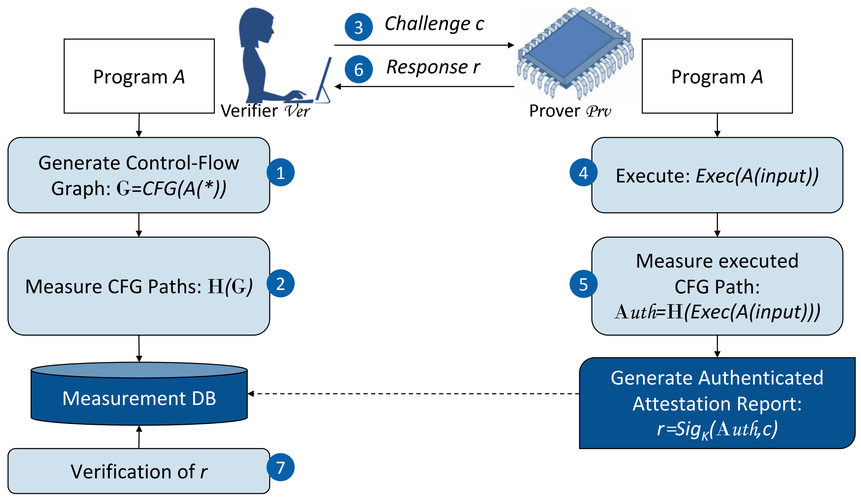
We present C-FLAT, a novel attestation scheme for run-time behavior

Why run-time attestation?

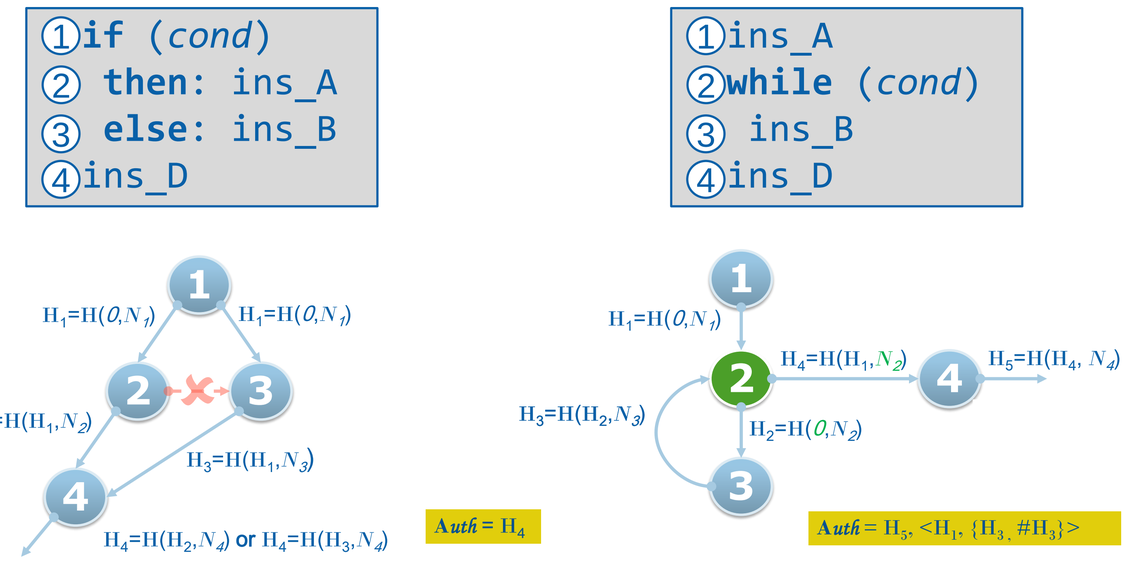
- Current remote attestation schemes measure integrity of a program binary
- However, control flow and data-oriented attacks tamper runtime state of program
- Modern countermeasures not available for resource-constrained embedded systems

High-level idea

- Trace and record control flow of prover
- Aggregate measurement in hash chain
  - Problem: efficient computation of CFG and explorations of all execution paths
  - Solution: allow verifier solve subproblems; prover treats loops as separate sub-graphs



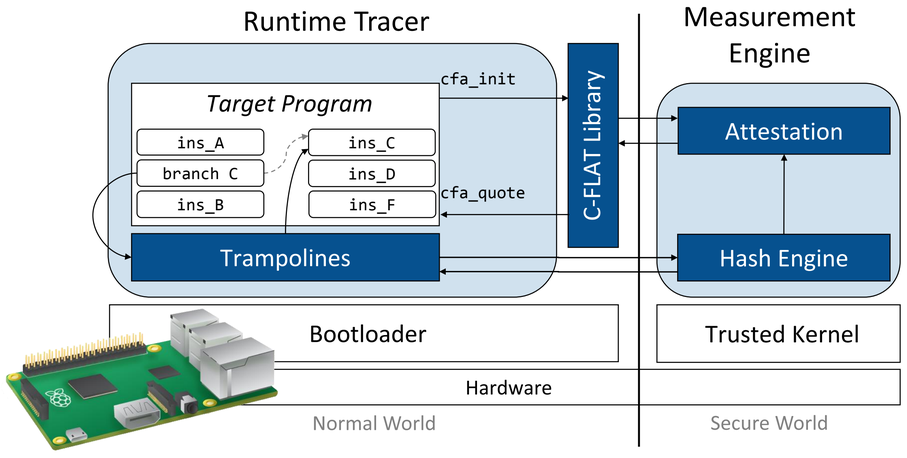
Overview of C-FLAT System Model



Sample controls flows; conditional and loop

Platform Requirements

- A small trust anchor, such as TrustZone-M or TrustLite for isolating Measurement Engine



C-FLAT Proof-of-Concept System Architecture

Proof-of-Concept on ARM TrustZone-A

- Bare-metal prototype on Raspberry Pi 2
- Single-purpose program instrumented using binary-rewriting
- Runtime Monitor written in ARM assembler
- Measurement Engine isolated in TrustZone-A Secure World