CFI CaRE: Hardware-supported Call and Return Enforcement for Commercial Microcontrollers

How can Control-Flow Integrity be realized on low-end IoT devices?

Goals and contributions
- First interrupt-aware CFI scheme for low-end (ARM) microcontrollers
- Hardware-based shadow stack protection using ARM TrustZone-M security architecture
- Memory layout-preserving binary instrumentation that can be realized on-device

Interrupt-awareness
- Existing CFI schemes designed for userspace code, and lack support for hardware interrupts
- CaRE validates returns from interrupt handlers to the point where external interrupt occurred

Layout-preserving instrumentation
- Prior binary instrumentation approaches for embedded ARM code are complex:
  - Binary rewriting introduces new instructions into existing binary and requires rewriting of instructions dependent on their address
  - Binary patching makes space for new instructions by moving instructions to trampolines, duplicating CFI code
- CaRE preserves memory layout, and places CFI logic in reusable Branch Monitor

Shadow stack protection
- A shadow stack contains control-flow information used to validate return events
- A strong adversary can both read and modify arbitrary program memory, incl. shadow stack
- Prior software-based shadow stack protection incurs orders of magnitude higher overhead compared to cost of instrumentation
- CaRE efficiently isolates shadow stack in protected memory, only accessible in TZ-M secure state

Proof-of-concept implementation
- PoC implementation on ARM Versatile Express Cortex-M Prototyping System+ synthesized as 25MHz Cortex-M23 processor with TZ-M

CaRE architecture overview