Status update: EDHOC-C formal verification

Timothy Claeys - Inria

EDHOC-C

- C implementation of EDHOC targeting microcontrollers and constrained systems.
- Goals:
  - All authentication methods
  - At least cipher suite 0, 1, 2 and, 3
  - Standalone application (https://github.com/openwsn-berkeley/EDHOC-C)
  - Integration in RIOT-OS and OpenWSN
- Status:
  - Tested signature-based authentication method
  - cipher suite 0
  - Integration in RIOT (WIP) with François-Xavier Molina
  - Integration in OpenWSN (WIP)

Formally verifying the implementation

RIOT-fp project:

- Joint work between Inria-EVA (Mališa Vučinić, Timothy Claeys) and Inria-Prosecco (Karthikeyan Bhargavan)

Goals:

- Memory safety
- Mitigations against timing side-channels
- Functional correctness w.r.t. a high-level specification (EDHOC spec)

Formally verifying the implementation

Status (since first meeting with Karthik january 28th):

- Refactored code in APIs that facilitate formal verification
  - Message processing API, Message formatting API, crypto API, credential API
  - Each API can be translated to Low* independently and tested with a unit test suite.

Roadmap:

- Two implementations:
  - a high-level “obviously” correct version, written in hacspec (subset of Rust)
  - a low-level implementation (Low*) translation of EDHOC-C code
- Compile hacspec implementation to F*
- Combine F* and Low* to generate “correct” C code