Development (1/2)

› Develop EDHOC implementation for Rust/hacspec [1][2] (Mališa Vučinić)
  - Provable correctness and memory safety; side-channel resistance
  - Verifiable code, but for microcontrollers
  - Portable to other environments

› Hackathon goal: interop testing of “minimal” implementation in Rust
  - Initiator only; Method 3 only (Static-Static); ID CRED: integer ‘kid’
  - No dependencies, execute without standard library (i.e., no_std)
  - Rely on hardware acceleration where possible

[1] Denis Merigoux, Franziskus Kiefer, Karthikeyan Bhargavan. Hacspec: succinct, executable, verifiable specification for high-assurance cryptography embedded in Rust, [https://hal.inria.fr/hal-03176482/document](https://hal.inria.fr/hal-03176482/document)
[2] [https://github.com/hacspec/hacspec](https://github.com/hacspec/hacspec)
Development (2/2)

› **Available at Hackathon opening**
  – Partial implementation of EDHOC
    › Unit functions and unit tests
    – Hacspec full crypto support
    – CC2538 hardware abstraction layer in Rust
    – nRF52840 hardware abstraction layer in Rust

› **Available at Hackathon closing [3]**
  – API
  – Protocol ”state machine”
  – Multi-target build support
  – AES and SHA-256 hardware-accelerated for CC2538

   *branch: ietf113-hackathon*
Testing

- Aligned with the Editor’s copy of EDHOC → to-be version -13
  - Mališa Vučinić (INRIA): Rust/hacspec - Initiator
  - Marco Tiloca (RISE): Java (Eclipse Californium) - Responder

- Tested configuration
  - Cipher suite 2
  - Method 3 (Static-Static)
  - Credential Type (CCS-CCS)
  - ID Credential Type (integer ‘kid’)
  - Minimum message_2 size (45 bytes)
  - Correctly completed EDHOC execution
Next steps

› More implementations are currently under update
  – Christian Amsüss: Python (*aiocoap*)
    ‣ Building on the *py-edhoc* from Timothy
  – Stefan Hristozov: C
  – Timothy Claeys: Python (*py-edhoc*) ; C

› Run more interop tests soon
  – As more implementations get updated / available
  – Consider use of message_4 and error messages

› Possible side-testing
  – (EDHOC message_3) + (first OSCORE-protected request) ➔ Single request on the wire
Thank you!
Backup
Interop test configurations

› **Tested configurations – [Cipher suite, Method, Cred Type, ID Cred Type]**
  – 1 implementation pair: [2, 3, CCS, kid (integer)] ➔ **Minimum message_2 size (45 bytes)**
  – TBD implementation pairs: (TBD, TBD, TBD)
  – TBD implementation pair:
    › (TBD, TBD, TBD) ; (TBD, TBD, TBD) ; (TBD, TBD, TBD)
    › (TBD, TBD, TBD) ; (TBD, TBD, TBD) ; (TBD, TBD, TBD)
    › (TBD, TBD, TBD) ; (TBD, TBD, TBD)
Interop test results

- 1 tested implementation pairs – 0 in both directions (*)
  - Marco with:
    - Mališa (Initiator) → Success
    - TBD
    - TBD (*)
    - TBD (*)
  - TBD with TBD (*)
  - TBD with TBD