SUIT

Secure Updates for Internet of Things

IETF SUIT

- Software Updates for Internet of Things (SUIT)
- Technology agnostic proposals of architecture and information model (manifest) for IoT updates
- Underlying assumptions:
  - Constrained networks and devices
  - Class 1 devices (~10 KiB RAM, ~100 KiB flash)
  - PKI
- For this thesis, proposed network stack is CoAP over UDP with DTLS

Architecture goals

- Not to make assumptions about means of distribution
- Support different image formats without requiring alterations to them
- State-of-the-art security mechanisms such as asymmetric cryptography using relatively high-end algorithms
- Prevent rollback attacks and operate with high reliability
- Support different modes of operation (push, pull, hybrid)

Manifest elements

- Mandatory elements:
  - Version identifier
  - Monotonic sequence number
  - Payload format
  - Payload digest
  - Size
- Mandatory (conditionally)
  - XIP address
  - Dependencies
  - Precursor image digest
  - Content key distribution method

Manifest elements cont.

- Recommended:
  - Vendor ID
  - Class ID
- Optional:
  - Required image list
  - Best-before timestamp
  - Component identifier
  - URIs
  - Directives
  - Aliases
  - Processing steps

Thesis considerations

- Are all conditions to be considered for the thesis?
  - They can at least be prepared for in manifest to comply with SUIT
- Differential updates
  - Could be implemented if time allows.
- Nested formats
  - Same as above, focus should be on a simple, working implementation first
- Heterogeneous storage systems/multiple MCUs
  - Firefly does not fall under this category, skip?
- Different key distribution methods
  - Solved by EST-coaps?

Thesis considerations cont.

- Architecture mentions device management and different distribution methods of firmware/payload
  - Does not feel central to the thesis

Thoughts so far

- CoAP observe option for clients to receive updated manifests
- CoAP block option for delivery of manifest and image
- EST-coaps for signing of manifest and exchange of DTLS keys
- SHA-256 for digests as both Contiki-NG and Firefly supports it
  - Can perhaps use a larger digest size, but as many digests has to be supported in the same manifest, size is of importance

Thoughts so far cont.

- Put the absolute minimum information needed for a singular update in manifest, rest as options
- Implement options as in CoAP

Example workflow

- Create/generate manifest in JSON on server side
  - ARM mbed has a prototype, can be used for inspiration https://github.com/ARMmbed/suit-manifest-generator
- Encode in CBOR, transmit to device
- Decode and run a small parser
- If valid, fetch image
- If no directives given, apply update