Constrained BRSKI (cBRSKI)

draft-ietf-anima-constrained-voucher-21

IETF 118, Prague, November 2023

Esko Dijk – IoTconsultancy.nl
(presenting)

Co-authors:
M. Richardson
P. van der Stok
P. Kampanakis

Image: various brands of constrained brewski
Recap & Goal – Constrained BRSKI

› cBRSKI = Constrained BRSKI

› cBRSKI = CoAP BRSKI 😊
Recap & Goal – Constrained BRSKI

› **TLDR:** BRSKI onboarding, for constrained (IoT) devices & networks
Issue 1: Discovery Variations

Discovery method for Pledge → Join Proxy?
- 1. CoAP discovery (link-local multicast request)
- 2. mDNS (link-local multicast request)
- 3. GRASP (link-local advertisements)
- 4. Network-specific (e.g. Thread, 6TiSCH ... )

Discovery method for Join Proxy → Registrar?
- 1. CoRE Resource Directory (RD)
- 2. Unicast DNS-SD
- 3. GRASP
- 4. Network-specific
Issue 1: Discovery Variations

- Discovery method for Pledge → Join Proxy?
  - Method used by Pledge MUST be supported by Join Proxy, else no interoperability!

- Discovery method for Join Proxy → Registrar?
  - Method used to register the Registrar MUST match discovery method of Join Proxy, else no interoperability!

- Mixing methods in a single deployment is unwanted complexity

- Are we okay with *allowing* multiple discovery methods?
  - Implies that interoperability needs to be achieved outside IETF or by a future document.
Issue 1: Discovery Variations

Possible Proposal

- Push the issue out of scope

- Only define 1 discovery variant: CoAP Discovery using CoRE Link Format payloads

- Leave further discovery optimizations to future documents, such as draft-eckert-anima-brski-discovery

- This draft should define general mechanisms that are easily translated between the different discovery technologies
Issue 2: support 2-tier and 3-tier CAs #275

› Current draft defines a simplification ...
› ... which allows a Pledge to get only 1 domain CA certificate via the “/cacerts” request

› But: this hampers the support of 2-tier and 3-tier CA structures – which will get more & more common

› Proposed Solution: a new format that avoids PKCS#7 container
› Using CoAP multipart (RFC 8710) – a CBOR array

\[
[60, \text{<CBOR-encoded-integer-number-N>}, 287, \text{<first-X509-binary-cert> } ]
\]

where N is the total number of CA certs available. Once N is known, the client can CoAP-GET the remaining (N-1) CA certificates (GET /.well-known/est/crts/<N> )
Thank you!

Comments/questions?

https://github.com/anima-wg/constrained-voucher/
Recap & Goal – Constrained BRSKI

› cBRSKI = Constrained BRSKI = CoAP BRSKI 😊
   – draft-ietf-anima-constrained-voucher-21

› BRSKI onboarding, for constrained (IoT) devices & networks
   – Suitable for wireless 6LoWPAN mesh networks and other constrained IP networks
   – Minimize time & code overhead: round-trips, format parsing, optional functions, ...

  – CoAP + DTLS ⇒ instead of HTTP + TLS
  – COSE-signed CBOR ⇒ instead of CMS-signed JSON
  – Constrained EST-coaps ⇒ instead of ‘classic’ EST
Updates Since -18 (@IETF-115)

› **YANG extensions** to the Voucher (extending RFC 8366) all **moved** to draft-RFC8366-bis

› Old Voucher / Voucher-Request **examples updated** + more examples

› IANA section added for **GRASP discovery**

› Detailed CoAP discovery
› DTLS text pulled into one section (version, handshake fragmentation, ... )
› Major editorial text rewrites!
Open Issues (kept in Github)

› https://github.com/anima-wg/constrained-voucher/issues

› 10 issues open
  – (Issues labeled “future” or “interop” are not for the document)

› Media type application/voucher+cose – are we okay? #264
  instead of ‘application/voucher-cose+cbor’
Pending Updates (Github PR)

- [https://github.com/anima-wg/constrained-voucher/pulls](https://github.com/anima-wg/constrained-voucher/pulls)

**Scheduled for next version -22:**

- Update terminology to RFC8366-bis (#280)
- Add mDNS discovery (#279)
- Rename Voucher media type to application/voucher+cose (#277)
FYI - Implementations & Interop

› Minerva.sandelman.ca
  – Registrar – Fountain
  – MASA – Highway
  – Pledge (simulated) – Reach

› IoTconsultancy.nl OpenThread Registrar fork
  – includes Registrar, MASA, Pledge (simulated)
  – code for OpenThread embedded Pledge (not public)
  – aims for integration into an automated testing framework ~ also testing “out of spec” cases
  – using Github issue tracker

› petervanderstok BRSKI
  – and test MASA

› Siemens-BT Registrar & MASA