Update of access rights

https://mailarchive.ietf.org/arch/msg/ace/dLkW-MYHXfZqmtY7AP7ZBDJpxOw/

Francesca Palombini, Ericsson

Status of OSCORE Profile of ACE

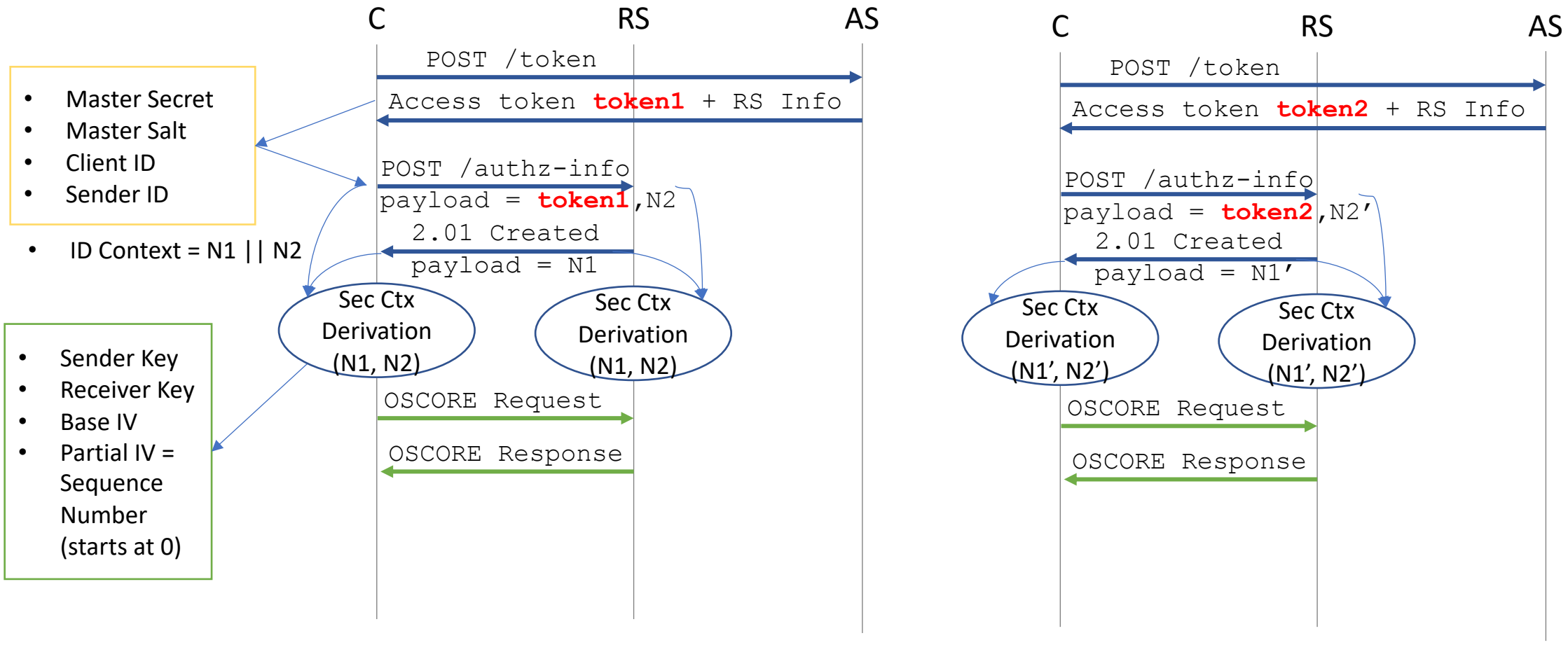
https://tools.ietf.org/html/draft-ietf-ace-oscore-profile-10

- Answered Ben's review in v-10

- V-11 in PR:
  - Answers OCF comments
  - Adresses Ben re-review
  - Attempts to address 2 leftover github issues from Jim

- Still missing:
  - Text explaining why we recommend 64 bits nonces
  - Update of access rights

Update of access rights - now



Update of access rights - now

- 1. Client retrieves access token T1 from AS
- 2. Client posts T1 to RS, together with nonce N1
- 3. RS replies with 2.01 and nonce N2
- 4. Client and RS derive OSCORE Sec Ctx "Sec1" from T1 ("osc" object), N1, N2
- 5. Client uses Sec1 to protect its request to RS
- 6. RS uses Sec1 to verify request. Verification success => Sec1 is validated and associated with T1 (at the RS)

- 7. Client wants to update its access rights: retrieves T2 from AS. Note that this T2 has different authorization info, but does not contain input keying material ("osc"), only a reference to identify Sec1 ("kid" in "cnf")
- 8. Client posts T2 to RS, together with nonce N1’
- 9. RS replies with 2.01 and nonce N2’
- 10. Client and RS derive OSCORE Sec Ctx "Sec2" from T1 keying input material ("osc" object), N1', N2’
- 11. Client uses Sec2 to protect its request to RS 12. RS uses Sec2 to verify request. Verification success => Sec2 is validated and associated with T2 (at the RS) ; T1 is removed ; Sec1 is removed

Proposal

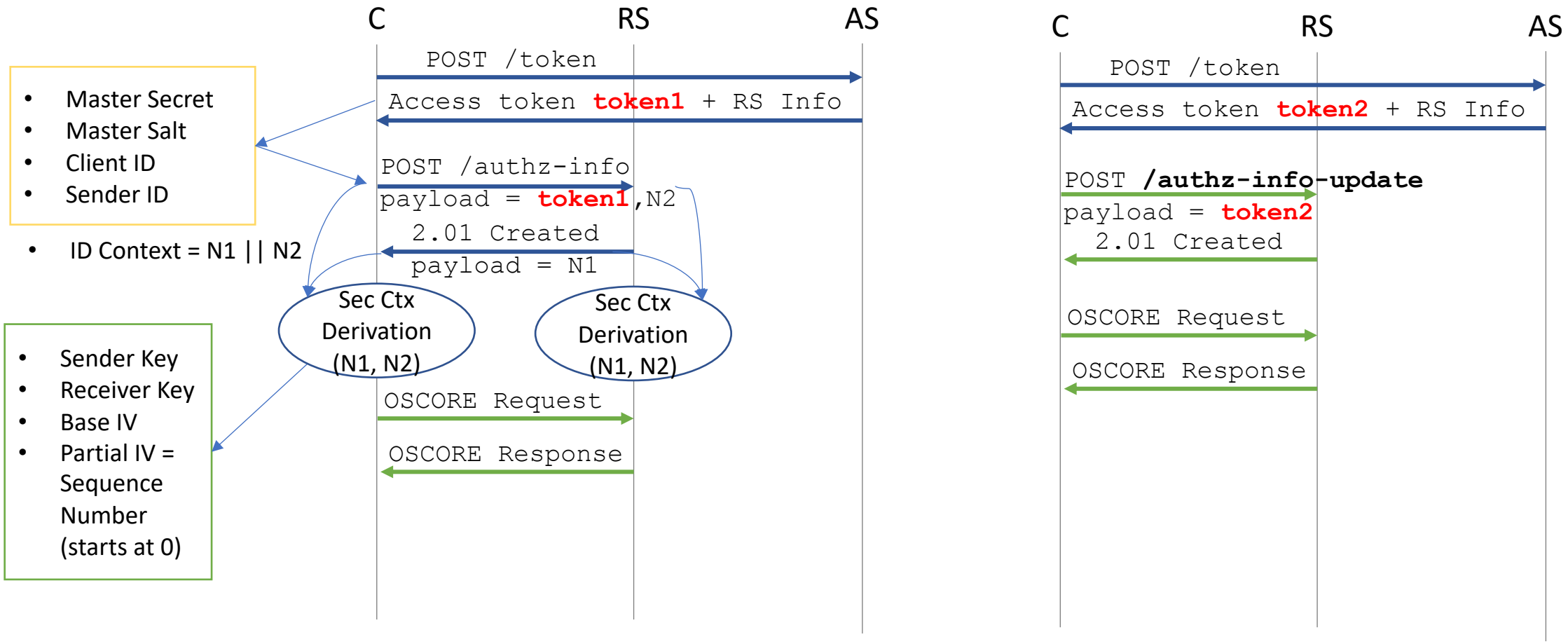
1: mandate that the access token to update the access rights MUST be sent over the secure channel.

- in OSCORE and DTLS profiles
- in framework too?

1.b: separate /authz-info and /authz-info-update endpoints at the RS

- simplifies processing and implementations
- /authz-info-update unprotected messages are rejected
- messages to /authz-info → always new security association C-RS

Update of access rights - proposal



Update of access rights - proposal

- 1. Client retrieves access token T1 from AS
- 2. Client posts T1 to RS, together with nonce N1
- 3. RS replies with 2.01 and nonce N2
- 4. Client and RS derive OSCORE Sec Ctx "Sec1" from T1 ("osc" object), N1, N2
- 5. Client uses Sec1 to protect its request to RS
- 6. RS uses Sec1 to verify request. Verification success => Sec1 is validated and associated with T1 (at the RS)

- 7. Client wants to update its access rights: retrieves T2 from AS. Note that this T2 has different authorization info, but does not contain input keying material, only a reference to identify Sec1
- 8. Client posts T2 to RS, **without nonce protected with Sec1**
- 9. RS **verifies that this is an update of access right, replacing T1 (associated with Sec1) ; Sec1 is associated with T2; T1 is removed**; RS replies with 2.01 **without nonce protected with Sec1**
- 10. Client uses **Sec1** to protect its request to RS

Feedback

- Ludwig → 1 yes. 1.b not necessary
- Rikard, Marco (ace OSCORE implementation) → 1 is doable even without 1.b for their implementation
- Michael R. → considerations on access rights(T1) and access rights(T2) (superset, subset, disjoint, subset + something else)
- Ben → possibility of collisions of kid (talk about key)