

Additional Authentication Credentials for the Datagram Transport Layer Security (DTLS) Profile for Authentication and Authorization for Constrained Environments (ACE)

draft-tiloca-ace-authcred-dtls-profile-00

Marco Tiloca, RISE
John P. Mattsson, Ericsson

IETF 117 Meeting – San Francisco – July 24th, 2023

Motivation

- › **The DTLS profile of ACE is defined in RFC 9202**
 - It has an “RPK mode” based on asymmetric authentication credentials
 - Authentication credentials are raw public keys (RPKs), only as COSE Keys

- › **Other types of asymmetric authentication credentials exist**
 - Other representations of RPKs → DTLS handshake: just like in the main case above
 - Public key certificates → DTLS handshake: ready to use those

- › **Good to support other credential formats for the Client (C) and Resource Server (RS)**

- › **Early idea shared during the ACE session at IETF 116**

Contribution

› Proposed update to RFC 9202

- Enable the use of alternative formats for public authentication credentials

› Update breakdown

- Extend the “RPK mode”, to support also CWT Claims Sets (CCSs) [1]
- Define a new “Certificate mode”, to support also X.509 [2] and C509 [3] public key certificates

› Seamlessly applicable if TLS is used between C and RS, as defined in RFC 9430

[1] <https://datatracker.ietf.org/doc/rfc8392/>

[2] <https://datatracker.ietf.org/doc/rfc5280/>

[3] <https://datatracker.ietf.org/doc/draft-ietf-cose-cbor-encoded-cert/>

Mechanics

› Use of new CWT Confirmation Methods

- "kccs", "x5bag", "x5chain", "c5b", and "c5c" – Defined in *draft-ietf-ace-edhoc-oscore-profile*

› As usual, specify authentication credentials in

- "req_cnf" parameter of C-to-AS token request (C's authentication credential)
- "cnf" claim of the access token (C's authentication credential)
- "rs_cnf" parameter of AS-to-C token response (RS' authentication credential)

› Possible to combine different credential formats

- Public keys in the "RPK mode": both as CCS / both as COSE key / one as CCS and one as COSE Key
- Certificates in the "Certificate mode": both X.509 / both C509 / one X.509 and one C509
- One public key as RPK, the other one in a certificate – Ok for (D)TLS, see Section 5.3 of RFC 7250

Example in “RPK mode”

Client → Authorization Server

Access Token Request in “RPK mode”

```
POST coaps://as.example.com/token
Content-Format: application/ace+cbor
Payload:
{
  "grant_type" : 2,
  "audience" : "tempSensor4711",
  "req_cnf" : {
    "kccs" : {
      "sub" : "42-50-31-FF-EF-37-32-39",
      "cnf" : {
        "COSE_Key" : {
          "kty" : 2,
          "crv" : 1,
          "x" : h'd7cc072de2205bdc1537a543d53c60a6
            acb62eccd890c7fa27c9e354089bbe13',
          "y" : h'f95e1d4b851a2cc80fff87d8e23f22af
            b725d535e515d020731e79a3b4e47120'
        }
      }
    }
  }
}
```

Client's RPK as CCS

Authorization Server → Client

Access Token Response in “RPK mode”

```
2.01 Created
Content-Format: application/ace+cbor
Max-Age: 3560
Payload:
{
  "access_token" : b64'S1AV32hk'/. ...
  (remainder of CWT omitted for brevity;
  CWT contains the client's RPK in the cnf claim),
  "expires_in" : 3600,
  "rs_cnf" : {
    "kccs" : {
      "sub" : "AA-BB-CC-00-01-02-03-04",
      "cnf" : {
        "COSE_Key" : {
          "kty" : 2,
          "crv" : 1,
          "x" : h'bbc34960526ea4d32e940cad2a234148
            ddc21791a12afbcbac93622046dd44f0',
          "y" : h'4519e257236b2a0ce2023f0931f1f386
            ca7afda64fcde0108c224c51eabf6072'
        }
      }
    }
  }
}
```

Resource Server's RPK as CCS

Example in “Certificate mode”

Client → Authorization Server

Access Token Request in “Certificate Mode”

```
POST coaps://as.example.com/token
Content-Format: application/ace+cbor
Payload:
```

```
{
  "grant_type" : 2,
  "audience" : "tempSensor4711",
  "req_cnf" : {
    "x5chain" : h'3081ee3081a1a003020102020462319ec430
      0506032b6570301d311b301906035504030c
      124544484f4320526f6f7420456432353531
      39301e170d3232303331363038323433365a
      170d3239313233313233303030305a302231
      20301e06035504030c174544484f43205265
      73706f6e6465722045643235353139302a30
      0506032b6570032100a1db47b95184854ad1
      2a0c1a354e418aace33aa0f2c662c00b3ac5
      5de92f9359300506032b6570034100b723bc
      01eab0928e8b2b6c98de19cc3823d46e7d69
      87b032478fecfaf14537a1af14cc8be829c6
      b73044101837eb4abc949565d86dce51cfae
      52ab82c152cb02'
  }
}
```

Client's X.509 Certificate

Authorization Server → Client

Access Token Response in “Certificate Mode”

```
2.01 Created
Content-Format: application/ace+cbor
Max-Age: 3560
Payload:
```

```
{
  "access_token" : b64'SlAV32hk'/...
  (remainder of CWT omitted for brevity;
  CWT contains the client's X.509 certificate in the cnf claim)/,
  "expires_in" : 3600,
  "rs_cnf" : {
    "x5chain" : h'3081ee3081a1a003020102020462319ea030
      0506032b6570301d311b301906035504030c
      124544484f4320526f6f7420456432353531
      39301e170d3232303331363038323430305a
      170d3239313233313233303030305a302231
      20301e06035504030c174544484f4320496e
      69746961746f722045643235353139302a30
      0506032b6570032100ed06a8ae61a829ba5f
      a54525c9d07f48dd44a302f43e0f23d8cc20
      b73085141e300506032b6570034100521241
      d8b3a770996bcfc9b9ead4e7e0a1c0db353a
      3bdf2910b39275ae48b756015981850d27db
      6734e37f67212267dd05eef27b9e7a813fa
      574b72a00b430b'
  }
}
```

Resource Server's X.509 Certificate

Next steps

- › **Consider the transfer of certificates by reference**
 - Using CWT Confirmation Methods “x5t” and “c5t”
 - Already defined in *draft-ietf-ace-edhoc-oscore-profile*

- › **More security considerations**
 - E.g., on validating a CCS, see also Section 9.8 of *draft-ietf-lake-edhoc-20*

- › **Comments and feedback are welcome!**

Thank you!

Comments/questions?

<https://gitlab.com/crimson84/draft-tiloca-ace-authcred-dtls-profile>