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

draft-ietf-ace-authcred-dtls-profile-01

Marco Tiloca, RISE
John Preuß Mattsson, Ericsson

IETF 122 Meeting – Bangkok – March 19th, 2025

Recap

- **The DTLS profile of ACE is defined in RFC 9202**
  - "RPK mode": asymmetric authentication credentials as raw public keys (RPKs), only as COSE Keys
- **Update to RFC 9202: enable the use of alternative formats for public authentication credentials**
  - Seamlessly applicable if TLS is used between Client and Resource Server (RFC 9430)
- **Update breakdown**
  - Extend the "RPK mode", to support also:
    - CWT Claims Sets (CCSs) [1] transported by value
    - COSE Keys identified by reference [2]
  - Define a new "Certificate mode", to support X.509 [3] and C509 [4] public key certificates
    - Possible to transport by value or to identify by reference

[1] https://datatracker.ietf.org/doc/rfc8392/
[2] https://datatracker.ietf.org/doc/rfc9679/
[3] https://datatracker.ietf.org/doc/rfc5280/
[4] https://datatracker.ietf.org/doc/draft-ietf-cose-cbor-encoded-cert/

Recent updates

- **The draft was adopted as a WG Document after IETF 121**

- **Editorial improvements**
  - Expanded acronyms on first use
  - Fixed examples in CBOR Diagnostic Notation

- **Placeholder CBOR abbreviations for CWT confirmation methods**
  - Values start after 5, now taken by the registered ‘ckt’

- **Updated references related to X.509 certificates**
  - Removed RFC 8398 and RFC8399, as obsoleted by RFC 9598 and RFC 9549
  - Added RFC 9549, RFC 9598, RFC 9608, and RFC 9618

Recent updates

- **New Section 2.2 “Raw Public Keys as COSE_Keys Identified by Reference”**
  - For the “RPK mode”, as an alternative to transporting COSE_Keys by value
  - Using the CWT confirmation method ‘ckt’ defined in RFC 9679
    - A “ckt” structure transports a thumbprint of a COSE Key (see Section 3 of RFC 9679)
    - SHA-256 MUST be used as hash function; it is mandatory to implement in RFC 9679

- **For the client’s authentication credential COSE_KEY_C, a “ckt” structure is included:**
  - In the “req_cnf” parameter of the C-to-AS access token request
  - In the “cnf” claim of the issued access token

- **For the resource server’s authentication credential COSE_KEY_RS, a “ckt” structure is included:**
  - In the “rs_cnf” parameter of the AS-to-C access token response

- **Both COSE_KEY_C and COSE_KEY_RS can be identified by reference, or only one instead**

Example in “RPK mode” using ‘ckt’

Client → Authorization Server

Access Token Request in “RPK mode”

```
POST coaps://as.example.com/token
Content-Format: 19 (application/ace+cbor)
Payload:
{
  / grant_type / 33 : 2 / client_credentials /,
  / audience /    5 : "tempSensor4711",
  / req_cnf /     4 : {
    / ckt / 5 : h'd3550f1b5b763ee09d058fc7aef69900
                  1279903a4a15bdc3953d32b10f7cb8b1'
  }
}
```

Client’s RPK as COSE_Key by reference

Authorization Server → Client

Access Token Response in “RPK mode”

```
2.01 Created
Content-Format: 19 (application/ace+cbor)
Max-Age: 3560
Payload:
{
  / access_token / 1 : h'd83dd083...643b',
    / (remainder of CWT omitted for brevity;
    CWT contains the client's RPK in the cnf claim) /
  / expires_in /   2 : 3600,
  / rs_cnf /      41 : {
    / ckt / 5 : h'db60f4d371fffac3e1040566154a5c36
                  1e0bf835a4ad4c58069cf6edc9ac58a3'
  }
}
```

Resource server’s RPK as COSE_Key by reference

Overview of available options

- **“RPK mode”**
  - COSE_Key transported by value (‘COSE_Key’) // already in RFC 9202
  - CCS transported by value (‘kccs’) (*)
  - COSE_Key identified by reference (‘ckt’)

- **“Certificate mode”**
  - X.509 certificate transported by value (‘x5chain’ or ‘x5bag’) (*)
  - C509 certificate transported by value (‘c5c’ or ‘c5b’) (*)
  - X.509 certificate identified by reference (‘x5t’ or ‘x5u’) (*)
  - C509 certificate identified by reference (‘c5t’ or ‘c5u’) (*)

- **The authentication credentials of C and RS can independently be in either of the 7 formats above**

(*) CWT confirmation method defined in *draft-ietf-ace-edhoc-oscore-profile*

Next steps

- **Placeholder CBOR abbreviations for CWT confirmation methods**
  - Align with values in other documents, especially in [5]
- **Informative reference to [6]**
  - The new formats can also be used in the “rs_cnf2” parameter of the access token response
  - What is defined in [6] can assist when the AS (C) has forgotten the credential of C (RS)
- **Security considerations on validating CCS**
  - Starting point: related security considerations in Section 9.8 of RFC 9528 (EDHOC)
- **More examples with hybrid settings in Appendix A**
- **Comments are welcome!**

[5] https://datatracker.ietf.org/doc/draft-ietf-ace-edhoc-oscore-profile/

[6] https://datatracker.ietf.org/doc/draft-ietf-ace-workflow-and-params/

Thank you!

Comments/questions?

https://github.com/ace-wg/ace-authcred-dtls-profile

Backup

Example in “RPK mode”

Client → Authorization Server

Access Token Request in “RPK mode”

```
POST coaps://as.example.com/token
Content-Format: 19 (application/ace+cbor)
Payload:
{
  / grant_type / 33 : 2 / client_credentials /,
  / audience /    5 : "tempSensor4711",
  / req_cnf /     4 : {
    e'kccs' : {
      / sub / 2 : "42-50-31-FF-EF-37-32-39",
      / cnf / 8 : {
        / COSE_Key / 1 : {
          / kty /    1 : 2 / EC2 /,
          / crv /   -1 : 1 / P-256 /,
          / x /     -2 : h'd7cc072de2205bdc1537a543d53c60a6
                            acb62eccd890c7fa27c9e354089bbe13',
          / y /     -3 : h'f95e1d4b851a2cc80fff87d8e23f22af
                            b725d535e515d020731e79a3b4e47120'
        }
      }
    }
  }
}
```

Client’s RPK as CCS

Authorization Server → Client

Access Token Response in “RPK mode”

```
2.01 Created
Content-Format: 19 (application/ace+cbor)
Max-Age: 3560
Payload:
{
  / access_token / 1 : h'd83dd083...643b',
    / (remainder of CWT omitted for brevity;
    CWT contains the client's RPK in the cnf claim) /
  / expires_in /   2 : 3600,
  / rs_cnf /      41 : {
    e'kccs' : {
      / sub / 2 : "AA-BB-CC-00-01-02-03-04",
      / cnf / 8 : {
        / COSE_Key / 1 : {
          / kty /  1 : 2 / EC2 /,
          / crv / -1 : 1 / P-256 /,
          / x /   -2 : h'bbc34960526ea4d32e940cad2a234148
                          ddc21791a12afbcbac93622046dd44f0',
          / y /   -3 : 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: 19 (application/ace+cbor)
Payload:
{
  / grant_type / 33 : 2 / client_credentials /,
  / audience /    5 : "tempSensor4711",
  / req_cnf /     4 : {
    e'x5chain' : h'3081ee3081a1a00302010202046231 9ec430
                   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: 19 (application/ace+cbor)
Max-Age: 3560
Payload:
{
  / access_token / 1 : h'd83dd083...2fa6',
    / (remainder of CWT omitted for brevity;
    CWT contains the client's X.509 certificate in the cnf claim) /
  / expires_in /   2 : 3600,
  / rs_cnf /      41 : {
    e'x5chain' : h'3081ee3081a1a00302010202046231 9ea030
                   0506032b6570301d311b301906035504030c
                   124544484f4320526f6f7420456432353531
                   39301e170d3232303331363038323430305a
                   170d3239313233313233303030305a302231
                   20301e06035504030c174544484f4320496e
                   69746961746f722045643235353139302a30
                   0506032b6570032100ed06a8ae61a829ba5f
                   a54525c9d07f48dd44a302f43e0f23d8cc20
                   b73085141e300506032b6570034100521241
                   d8b3a770996bcfc9b9ead4e7e0a1c0db353a
                   3bdf2910b39275ae48b756015981850d27db
                   6734e37f67212267dd05eeff27b9e7a813fa
                   574b72a00b430b'
  }
}
```

Resource server’s X.509 certificate