

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-02

Marco Tiloca, RISE
John Preuß Mattsson, Ericsson

IETF 123 Meeting – Madrid – July 24th, 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

› Editorial improvements, including:

- The CoAP definition of “endpoint” from RFC 7252, as not used here
- Clearer overview of extensions to the RPK mode in Section 1
- Homogenized way to refer to “public keys” and “authentication credentials”

› Minor fixes in the examples


- Mostly to disambiguate values of audiences for different Resource Servers

› Updated references

- Added entries used in the security considerations
- More recent reference to SHA-256, i.e., FIPS PUB 180-4 (August 2015)

› Updated CBOR abbreviations of CWT Confirmation Methods

- More efficient use of codepoints
- Aligned with the abbreviations used in [5]



```
; CWT Confirmation Methods
x5t = 6
c5t = 8
kccs = 11
x5chain = 24
c5c = 26
```

Recent updates

› Considerations on providing credentials by value or by reference

- Providing a credential by reference assumes that:
 - › The consumer is already storing the identified credential; or
 - › The consumer can retrieve the identified credential from a trusted source
- Provide by value or by reference? It depends on context and application policies
- C and AS might explicitly coordinate, e.g., using the method in [6]
 - › That helps when the AS (C) has forgotten the credential of C (RS)

› Security considerations on validating CCSs and certificates

- Largely based on related security considerations in Section 9.8 of RFC 9528 (EDHOC)
- C and the RS are still responsible for verifying integrity and validity
- CAs and trust anchors have to be chosen carefully; revocation should be supported

Recent updates

› Appendix A: added two more examples with hybrid settings

- Certificate Mode (Certificates of Different Formats)
 - › C's credential : X.509 certificate by reference { 'x5t' }
 - › RS' credential : C509 certificate by reference { 'c5t' }
- Combination of RPK Mode and Certificate Mode
 - › C's credential : COSE_Key by reference { 'ckt' }
 - › RS' credential : X.509 certificate by reference { 'x5t' }

{ ... } CWT confirmation
method used

› Full set of available examples:

- C: { 'kccs' } and RS: { 'kccs' } ||| C: { 'ckt' } and RS: { 'ckt' }
- C: { 'x5chain' } and RS: { 'x5chain' } ||| C: { 'x5t' } and RS: { 'x5t' }

- C: { 'COSE_Key' } and RS: { 'kccs' }
- C: { 'x5chain' } and RS: { 'c5c' } ||| C: { 'x5t' } and RS: { 'c5t' }
- C: { 'kccs' } and RS: { 'x5chain' } ||| C: { 'ckt' } and RS: { 'x5t' }

Document body

Appendix A

Overview of available options

› “RPK mode”

- COSE_Key transported by value { ‘COSE_Key’ } // already possible in RFC 9202
- CCS transported by value { ‘kccs’ } (*)
- COSE_Key identified by reference { ‘ckt’ }

{ ... } CWT confirmation
method used

› “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*

(**) C509 certificates defined in *draft-ietf-cose-cbor-encoded-cert*

Next steps

- › **All the intended content has been included; no open points or issues**
- › **Ready for WG Last Call, but two normative references are Internet Drafts**
 - (A) <https://datatracker.ietf.org/doc/draft-ietf-ace-edhoc-oscore-profile/>
 - (B) <https://datatracker.ietf.org/doc/draft-ietf-cose-cbor-encoded-cert/>
- › **Until both A and B are sent to the IESG ...**
 - Hold on any further procedural steps forward for this document?
 - Hold on the publication request for this document?
- › **Directions from WG Chairs and AD?**

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 : "tempSensor1",
  / 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'f95e1d4b851a2cc80ffff87d8e23f22af
                    b725d535e515d020731e79a3b4e47120'
        }
      }
    }
  }
}
```

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 : "tempSensor3",
  / req_cnf / 4 : {
```

```
    e'x5chain' : h'3081ee3081a1a003020102020462319ec430
    0506032b6570301d311b301906035504030c
    124544484f4320526f6f7420456432353531
    39301e170d3232303331363038323433365a
    170d3239313233313233303030305a302231
    20301e06035504030c174544484f43205265
    73706f6e6465722045643235353139302a30
    0506032b6570032100a1db47b95184854ad1
    2a0c1a354e418aace33aa0f2c662c00b3ac5
    5de92f9359300506032b6570034100b723bc
    01eab0928e8b2b6c98de19cc3823d46e7d69
    87b032478fecfaf14537a1af14cc8be829c6
    b73044101837eb4abc949565d86dce51cfae
    52ab82c152cb02'
```

```
  }
```

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'3081ee3081a1a003020102020462319ea030
    0506032b6570301d311b301906035504030c
    124544484f4320526f6f7420456432353531
    39301e170d3232303331363038323430305a
    170d3239313233313233303030305a302231
    20301e06035504030c174544484f4320496e
    69746961746f722045643235353139302a30
    0506032b6570032100ed06a8ae61a829ba5f
    a54525c9d07f48dd44a302f43e0f23d8cc20
    b73085141e300506032b6570034100521241
    d8b3a770996bcfc9b9ead4e7e0a1c0db353a
    3bdf2910b39275ae48b756015981850d27db
    6734e37f67212267dd05eeff27b9e7a813fa
    574b72a00b430b'
```

```
  }
```

Resource server's X.509 certificate