

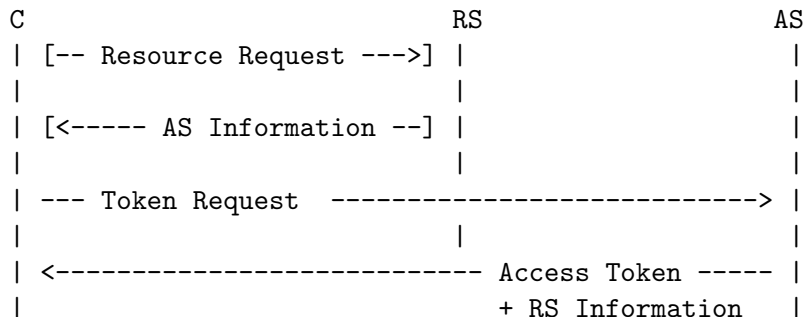
Datagram Transport Layer Security (DTLS) Profile for Authentication and Authorization for Constrained Environments (ACE)

draft-gerdes-ace-dtls-authorize

S. Gerdes, O. Bergmann, **C. Bormann**, G. Selander, L. Seitz

IETF97, 2016-11-17, Seoul

ACE Framework



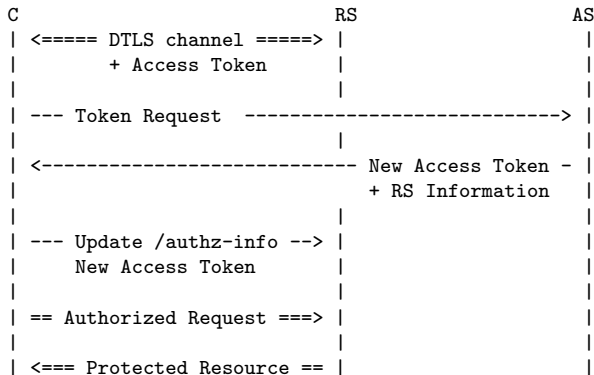
- ▶ RS has registered at AS for profile `coap_dtls`
- ▶ Optional unauthorized request (RS declines with AS info)
- ▶ C requests access token from AS for communication with RS
 - ▶ general assumption: access tokens are PoP tokens
- ▶ AS includes RS information in AS-to-Client response

Authorized Communication

C	RS	AS
[--- Access Token ----->]		
<== DTLS channel setup ==>		
== Authorized Request ==>		
<=== Protected Resource ==		

- ▶ C uploads access token to RS (/authz-info)
- ▶ C uses RS information to establish DTLS channel
 - ▶ RPK mode or PSK mode
- ▶ DTLS session identifies C
 - ▶ All access tokens for C apply

Dynamic Update of Authorization Information



- ▶ C retrieves new access token from AS and uploads to RS (/authz-info)
- ▶ C MAY re-negotiate DTLS session based on new token

RPK Mode: Client-to-AS Request

- ▶ Client-to-AS request MUST contain cnf object either with
 - ▶ C's raw public key, or
 - ▶ a known unique identifier of C's public key.

```
POST coaps://as.example.com/token
Content-Format: application/cbor
{
  grant_type:    client_credentials,
  aud:           "tempSensor4711",
  cnf: {
    COSE_Key: {
      kty: EC2,
      crv: P-256,
      x:   h'...',
      y:   h'...'
    }
  }
}
```

RPK Mode: AS-to-Client Response

2.01 Created

Location-Path: /authz-info/37

Content-Format: application/cbor

```
{
  access_token: b64'SlAV32hkKG ...
    (remainder of CWT omitted for brevity;
    CWT contains COSE_Key in the 'cnf' claim)',
  profile: coap_dtls,
  expires_in: 3600,
  cnf: {
    COSE_Key: { ... }
  }
}
```

- ▶ profile is coap_dtls
- ▶ Contains cnf object with RS's public key
- ▶ C uploads access token to RS before DTLS handshake
- ▶ C MUST use RPK denoted in Client-to-AS request in DTLS handshake

PSK Mode: Client-to-AS Request

- ▶ Client-to-AS request MAY contain `cnf` object with `kid` for existing session key generated by AS
 - simplify dynamic updates

```
POST coaps://as.example.com/token
Content-Format: application/cbor
{
  grant_type:    client_credentials,
  aud:          "tempSensor4711",
}
```

PSK Mode: AS-to-Client Response

2.01 Created

Content-Format: application/cbor

Location-Path: /token/asdjaskd

Max-Age: 86400

```
{
  access_token: b64'SlAV32hkKG ...
  token_type:   pop,
  alg:          HS256,
  expires_in:   86400,
  profile:      coap_dtls,
  cnf: {
    COSE_Key: {
      kty: symmetric,
      k: h'7365737369666e6b6579'
    }
  }
}
```

- ▶ profile is coap_dtls
- ▶ Contains cnf object with symmetric session key
- ▶ C uploads access token to RS before DTLS handshake or includes it in *psk_identity*

PSK Mode: DTLS Channel Setup

- ▶ C uses key from AS-to-Client response as shared secret
- ▶ RS extracts shared secret from access token
 - ▶ encrypted with some key known by RS and AS, or
 - ▶ derived from access token and some key known by RS and AS (HKDF SHA-256 as mandatory KDF)

- ▶ Updating authorization information
 - ▶ upload new access token, or
 - ▶ optionally re-negotiate DTLS session with access token as `psk_identity`, or
 - ▶ perform a new DTLS handshake.

Open Issues

1. Move AS discovery (*unauthorized request* + *AS Information response*) to framework document?
 - ▶ unprotected AS information in RS response
 - ▶ C should check against list of trustworthy authorization servers
 - ▶ currently documented in Security Considerations
2. Move description of error handling to framework document?
 - ▶ E.g., specification when to send 4.03 and 4.05 for CoAP requests received on a secure DTLS channel.
3. Using timestamps as nonce
 - ▶ Need to consider recent work on time synchronization between AS and RS