

## Additional OAuth Parameters for Authentication and Authorization for Constrained Environments (ACE)

### Abstract

This specification defines new parameters and encodings for the OAuth 2.0 token and introspection endpoints when used with the framework for Authentication and Authorization for Constrained Environments (ACE). These are used to express the proof-of-possession (PoP) key the client wishes to use, the PoP key that the authorization server has selected, and the PoP key the resource server uses to authenticate to the client.

### Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc9201>.

### Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

### Table of Contents

1. Introduction
2. Terminology
3. Parameters for the Token Endpoint
  - 3.1. Client-to-AS Request
  - 3.2. AS-to-Client Response

5.	Confirmation Method Parameters
6.	CBOR Mappings
7.	Requirements When Using Asymmetric Keys
8.	Security Considerations
9.	Privacy Considerations
10.	IANA Considerations
10.1.	OAuth Parameter Registration
10.2.	OAuth Parameters CBOR Mappings Registration
10.3.	OAuth Token Introspection Response CBOR Mappings Registration
11.	References
11.1.	Normative References
11.2.	Informative References
	Acknowledgments
	Author's Address

## 1. Introduction

The Authentication and Authorization for Constrained Environments (ACE) specification [RFC9200] requires some new parameters for interactions with the OAuth 2.0 [RFC6749] token and introspection endpoints, as well as some new claims to be used in access tokens. These parameters and claims can also be used in other contexts and have therefore been put into a dedicated document to facilitate their use in a manner independent of [RFC9200].

Note that although all examples are shown in Concise Binary Object Representation (CBOR) [RFC8949], JSON [RFC8259] MAY be used as an alternative for HTTP-based communications, as specified in [RFC9200].

## 2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Readers are assumed to be familiar with the terminology from [RFC9200], especially the terminology for entities in the architecture such as client (C), resource server (RS), and authorization server (AS).

Terminology from [RFC8152] is used in the examples, especially COSE\_Key, which is defined in Section 7 of [RFC8152].

Note that the term "endpoint" is used here following its OAuth 2.0 [RFC6749] definition, which is to denote resources such as token and introspection at the AS and authz-info at the RS. The Constrained Application Protocol (CoAP) [RFC7252] definition, which is "[a]n entity participating in the CoAP protocol", is not used in this specification.

## 3. Parameters for the Token Endpoint

This section defines additional parameters for the interactions with

the token endpoint in the ACE framework [RFC9200].

### 3.1. Client-to-AS Request

This section defines the `req_cnf` parameter allowing clients to request a specific PoP key in an access token from a token endpoint in the ACE framework [RFC9200]:

`req_cnf`

OPTIONAL. This field contains information about the key the client would like to bind to the access token for proof of possession. It is RECOMMENDED that an AS rejects a request containing a symmetric key value in the `req_cnf` field (`kty=Symmetric`), since the AS is expected to be able to generate better symmetric keys than a constrained client. (Note: this does not apply to key identifiers referencing a symmetric key.) The AS MUST verify that the client really is in possession of the corresponding key. Profiles of [RFC9200] using this specification MUST define the PoP method used by the AS if they allow clients to use this request parameter. Values of this parameter follow the syntax and semantics of the `cnf` claim either from Section 3.1 of [RFC8747] for CBOR-based interactions or from Section 3.1 of [RFC7800] for JSON-based interactions.

Figure 1 shows a request for an access token using the `req_cnf` parameter to request a specific public key as a PoP key. The content is displayed in CBOR diagnostic notation with line breaks for better readability.

Header: POST (Code=0.02)  
Uri-Host: "as.example.com"  
Uri-Path: "token"  
Content-Format: application/ace+cbor  
Payload:

```
{
  / req_cnf / 4 : {
    / COSE_Key / 1 : {
      / kty / 1 : 2 /EC2/,
      / kid / 2 : h'11',
      / crv / -1 : 1 /P-256/,
      / x / -2 : h'BAC5B11CAD8F99F9C72B05CF4B9E26D24
                    4DC189F745228255A219A86D6A09EFF',
      / y / -3 : h'20138BF82DC1B6D562BE0FA54AB7804A3
                    A64B6D72CCFED6B6FB6ED28BBFC117E'
    }
  }
}
```

Figure 1: Example Request for an Access Token Bound to an Asymmetric Key

### 3.2. AS-to-Client Response

This section defines the following additional parameters for an AS response to a request to the token endpoint:

cnf

REQUIRED if the token type is "pop" and a symmetric key is used. MAY be present for asymmetric PoP keys. This field contains the PoP key that the AS selected for the token. Values of this parameter follow the syntax and semantics of the cnf claim either from Section 3.1 of [RFC8747] for CBOR-based interactions or from Section 3.1 of [RFC7800] for JSON-based interactions. See Section 5 for additional discussion of the usage of this parameter.

rs\_cnf

OPTIONAL if the token type is "pop" and asymmetric keys are used. MUST NOT be present otherwise. This field contains information about the public key used by the RS to authenticate. If this parameter is absent, either the RS does not use a public key or the AS knows that the RS can authenticate itself to the client without additional information. Values of this parameter follow the syntax and semantics of the cnf claim either from Section 3.1 of [RFC8747] for CBOR-based interactions or from Section 3.1 of [RFC7800] for JSON-based interactions. See Section 5 for additional discussion of the usage of this parameter.

Figure 2 shows an AS response containing a token and a cnf parameter with a symmetric PoP key.

Header: Created (Code=2.01)

Content-Format: application/ace+cbor

Payload:

```
{
  / access_token / 1 : h'4A5015DF686428/...
  (remainder of CWT omitted for brevity;
  CWT contains COSE_Key in the "cnf" claim) / ',
  / cnf / 8 : {
    / COSE_Key / 1 : {
      / kty / 1 : 4 / Symmetric /,
      / kid / 2 : h'DFD1AA97',
      / k / -1 : h'849B5786457C1491BE3A76DCEA6C427108'
    }
  }
}
```

Figure 2: Example AS Response with an Access Token Bound to a Symmetric Key

Figure 3 shows an AS response containing a token bound to a previously requested asymmetric PoP key (not shown) and an rs\_cnf parameter containing the public key of the RS.

Header: Created (Code=2.01)

Content-Format: application/ace+cbor

Payload:

```
{
  / access_token / 1 : h'D08343A1010AA1054D2A45DF6FBC5A5A/...
  (remainder of CWT omitted for brevity) / ',
  / rs_cnf / 41 : {
    / COSE_Key / 1 : {
```

```

    / kty / 1 : 2 /EC2/,
    / kid / 2 : h'12',
    / crv / -1 : 1 /P-256/,
    / x / -2 : h'BCEE7EAAC162F91E6F330F5771211E220
              B8B546C96589B0AC4AD0FD24C77E1F1',
    / y / -3 : h'C647B38C55EFBBC4E62E651720F002D5D
              75B2E0C02CD1326E662BCA222B90416'
  }
}
}

```

Figure 3: Example AS Response Including the RS's Public Key

#### 4. Parameters for the Introspection Endpoint

This section defines the use of CBOR instead of JSON for the `cnf` introspection response parameter specified in Section 9.4 of [RFC8705].

If CBOR is used instead of JSON in an interaction with the introspection endpoint, the AS MUST use the parameter mapping specified in Table 1 and the value must follow the syntax of `cnf` claim values from Section 3.1 of [RFC8747].

Figure 4 shows an AS response to an introspection request including the `cnf` parameter to indicate the PoP key bound to the token.

Header: Created (Code=2.01)  
 Content-Format: application/ace+cbor  
 Payload:

```

{
  / active / 10 : true,
  / scope / 9 : "read",
  / aud / 3 : "tempSensor4711",
  / cnf / 8 : {
    / COSE_Key / 1 : {
      / kty / 1 : 2 /EC2/,
      / kid / 2 : h'11',
      / crv / -1 : 1 /P-256/,
      / x / -2 : h'BAC5B11CAD8F99F9C72B05CF4B9E26D24
                4DC189F745228255A219A86D6A09EFF',
      / y / -3 : h'20138BF82DC1B6D562BE0FA54AB7804A3
                A64B6D72CCFED6B6FB6ED28BBFC117E'
    }
  }
}

```

Figure 4: Example Introspection Response

#### 5. Confirmation Method Parameters

The confirmation method parameters are used in [RFC9200] as follows:

- \* `req_cnf` in the access token request C -> AS, OPTIONAL to indicate the client's raw public key or the key identifier of a previously established key between the C and RS that the client wishes to use

for proof of possession of the access token.

- \* `cnf` in the token response AS -> C, OPTIONAL if using an asymmetric key or a key that the client requested via a key identifier in the request. REQUIRED if the client didn't specify a `req_cnf` and symmetric keys are used. Used to indicate the symmetric key generated by the AS for proof of possession of the access token.
- \* `cnf` in the introspection response AS -> RS, REQUIRED if the access token that was subject to introspection is a PoP token, absent otherwise. Indicates the PoP key bound to the access token.
- \* `rs_cnf` in the token response AS -> C, OPTIONAL to indicate the public key of the RS if it uses one to authenticate itself to the client and the binding between the key and RS identity is not established through other means.

Note that the `COSE_Key` structure in a confirmation claim or parameter may contain an `alg` or `key_ops` parameter. If such parameters are present, a client MUST NOT use a key that is incompatible with the profile or PoP algorithm according to those parameters. An RS MUST reject a proof of possession using such a key with a response code equivalent to the CoAP code 4.00 (Bad Request).

If an access token is issued for an audience that includes several RSs, the `rs_cnf` parameter MUST NOT be used, since the client cannot determine for which RS the key applies. This document recommends to specify a different endpoint that the client can use to acquire RS authentication keys in such cases. The specification of such an endpoint is out of scope for this document.

## 6. CBOR Mappings

If CBOR is used, the new parameters and claims defined in this document MUST be mapped to CBOR types, as specified in Table 1, using the given integer abbreviation for the map key.

Name	CBOR Key	Value Type	Usage
<code>req_cnf</code>	4	map	token request
<code>cnf</code>	8	map	token response
<code>cnf</code>	8	map	introspection response
<code>rs_cnf</code>	41	map	token response

Table 1: CBOR Mappings for New Parameters and Claims

## 7. Requirements When Using Asymmetric Keys

An RS using asymmetric keys to authenticate to the client MUST NOT hold several different asymmetric key pairs applicable to the same authentication algorithm. For example, when using DTLS, the RS MUST

NOT hold several asymmetric key pairs applicable to the same cipher suite. The reason for this restriction is that the RS has no way of determining which key to use before the client's identity is established. Therefore, authentication attempts by the RS could randomly fail based on which key the RS selects, unless the algorithm negotiation produces a unique choice of key pair for the RS.

## 8. Security Considerations

This document is an extension to [RFC9200]. All security considerations from that document apply here as well.

## 9. Privacy Considerations

This document is an extension to [RFC9200]. All privacy considerations from that document apply here as well.

## 10. IANA Considerations

### 10.1. OAuth Parameter Registration

This section registers the following parameters in the "OAuth Parameters" registry [IANA.OAuthParameters]:

Name: req\_cnf  
Parameter Usage Location: token request  
Change Controller: IETF  
Reference: Section 5 of RFC 9201

Name: rs\_cnf  
Parameter Usage Location: token response  
Change Controller: IETF  
Reference: Section 5 of RFC 9201

Name: cnf  
Parameter Usage Location: token response  
Change Controller: IETF  
Reference: Section 5 of RFC 9201

### 10.2. OAuth Parameters CBOR Mappings Registration

This section registers the following parameter mappings in the "OAuth Parameters CBOR Mappings" registry established in Section 8.10 of [RFC9200].

Name: req\_cnf  
CBOR Key: 4  
Value Type: map  
Reference: Section 3.1 of RFC 9201  
Original Specification: RFC 9201

Name: cnf  
CBOR Key: 8  
Value Type: map  
Reference: Section 3.2 of RFC 9201  
Original Specification: RFC 9201

Name: rs\_cnf  
CBOR Key: 41  
Value Type: map  
Reference: Section 3.2 of RFC 9201  
Original Specification: RFC 9201

### 10.3. OAuth Token Introspection Response CBOR Mappings Registration

This section registers the following parameter mapping in the "OAuth Token Introspection Response CBOR Mappings" registry established in Section 8.12 of [RFC9200].

Name: cnf  
CBOR Key: 8  
Value Type: map  
Reference: Section 4 of RFC 9201  
Original Specification: [RFC8705]

## 11. References

### 11.1. Normative References

- [IANA.OAuthParameters] IANA, "OAuth Parameters", <<https://www.iana.org/assignments/oauth-parameters>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC6749] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework", RFC 6749, DOI 10.17487/RFC6749, October 2012, <<https://www.rfc-editor.org/info/rfc6749>>.
- [RFC7800] Jones, M., Bradley, J., and H. Tschofenig, "Proof-of-Possession Key Semantics for JSON Web Tokens (JWTs)", RFC 7800, DOI 10.17487/RFC7800, April 2016, <<https://www.rfc-editor.org/info/rfc7800>>.
- [RFC8152] Schaad, J., "CBOR Object Signing and Encryption (COSE)", RFC 8152, DOI 10.17487/RFC8152, July 2017, <<https://www.rfc-editor.org/info/rfc8152>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <<https://www.rfc-editor.org/info/rfc8259>>.
- [RFC8705] Campbell, B., Bradley, J., Sakimura, N., and T. Lodderstedt, "OAuth 2.0 Mutual-TLS Client Authentication



and Certificate-Bound Access Tokens", RFC 8705,  
DOI 10.17487/RFC8705, February 2020,  
<<https://www.rfc-editor.org/info/rfc8705>>.

- [RFC8747] Jones, M., Seitz, L., Selander, G., Erdtman, S., and H. Tschofenig, "Proof-of-Possession Key Semantics for CBOR Web Tokens (CWTs)", RFC 8747, DOI 10.17487/RFC8747, March 2020, <<https://www.rfc-editor.org/info/rfc8747>>.
- [RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <<https://www.rfc-editor.org/info/rfc8949>>.
- [RFC9200] Seitz, L., Selander, G., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "Authentication and Authorization for Constrained Environments (ACE) Using the OAuth 2.0 Framework (ACE-OAuth)", RFC 9200, DOI 10.17487/RFC9200, August 2022, <<https://www.rfc-editor.org/info/rfc9200>>.

## 11.2. Informative References

- [RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <<https://www.rfc-editor.org/info/rfc7252>>.

## Acknowledgments

This document is a product of the ACE Working Group of the IETF. Special thanks to Brian Campbell for his thorough review of this document.

Ludwig Seitz worked on this document as part of the CelticNext projects CyberWI and CRITISEC with funding from Vinnova.

## Author's Address

Ludwig Seitz  
Combitech  
Djäknegatan 31  
SE-211 35 Malmö  
Sweden  
Email: [ludwig.seitz@combitech.com](mailto:ludwig.seitz@combitech.com)