Workgroup: ACE Working Group
Internet-Draft:
draft-ietf-ace-revoked-token-notification-04
Published: 13 March 2023
Intended Status: Standards Track
Expires: 14 September 2023

Authors: M. Tiloca   L. Seitz   F. Palombini   S. Echeverria
         RISE AB   Combitech   Ericsson AB   CMU SEI
         G. Lewis
         CMU SEI

Notification of Revoked Access Tokens in the Authentication and Authorization for Constrained Environments (ACE) Framework

Abstract

This document specifies a method of the Authentication and Authorization for Constrained Environments (ACE) framework, which allows an Authorization Server to notify Clients and Resource Servers (i.e., registered devices) about revoked Access Tokens. The method allows Clients and Resource Servers to access a Token Revocation List on the Authorization Server, with the possible additional use of resource observation for the Constrained Application Protocol (CoAP). Resulting (unsolicited) notifications of revoked Access Tokens complement alternative approaches such as token introspection, while not requiring additional endpoints on Clients and Resource Servers.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the Authentication and Authorization for Constrained Environments Working Group mailing list (ace@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/ace/.

Source for this draft and an issue tracker can be found at https://github.com/ace-wg/ace-revoked-token-notification.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Authentication and Authorization for Constrained Environments (ACE) [RFC9200] is a framework that enforces access control on IoT devices acting as Resource Servers. In order to use ACE, both Clients and Resource Servers have to register with an Authorization Server (AS) and become a registered device. Once registered, a Client can send a request to the AS, to obtain an Access Token for a Resource Server (RS). For a Client to access the RS, the Client must present the issued Access Token at the RS, which then validates it before storing it (see Section 5.10.1.1 of [RFC9200]).

Even though Access Tokens have expiration times, there are circumstances by which an Access Token may need to be revoked before its expiration time, such as: (1) a registered device has been compromised, or is suspected of being compromised; (2) a registered device is decommissioned; (3) there has been a change in the ACE profile for a registered device; (4) there has been a change in access policies for a registered device; and (5) there has been a change in the outcome of policy evaluation for a registered device (e.g., if policy assessment depends on dynamic conditions in the execution environment, the user context, or the resource utilization).

As discussed in Section 6.1 of [RFC9200], only client-initiated revocation is currently specified [RFC7009] for OAuth 2.0 [RFC6749], based on the assumption that Access Tokens in OAuth are issued with a relatively short lifetime. However, this is not expected to be the case for constrained, intermittently connected devices, that need Access Tokens with relatively long lifetimes.

This document specifies a method for allowing registered devices to access and possibly subscribe to a Token Revocation List (TRL) resource on the AS, in order to obtain an updated list of revoked, but yet not expired, pertaining Access Tokens. In particular, registered devices can subscribe to the TRL at the AS by using resource observation [RFC7641] for the Constrained Application Protocol (CoAP) [RFC7252].

Unlike in the case of token introspection (see Section 5.9 of [RFC9200]), a registered device does not provide an owned Access Token to the AS for inquiring about its current state. Instead, registered devices simply obtain an updated list of revoked, but yet not expired, pertaining Access Tokens, as efficiently identified by corresponding hash values.

The benefits of this method are that it complements token introspection, and it does not require any additional endpoints on the registered devices. The only additional requirements for registered devices are a request/response interaction with the AS to access and possibly subscribe to the TRL (see Section 2), and the lightweight computation of hash values to use as Token identifiers (see Section 3).

1.1. 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 expected to be familiar with the terms and concepts described in the ACE framework for Authentication and Authorization [RFC9200], as well as with terms and concepts related to CBOR Web Tokens (CWTs) [RFC8392], and JSON Web Tokens (JWTs) [RFC7519].

The terminology for entities in the considered architecture is defined in OAuth 2.0 [RFC6749]. In particular, this includes Client, Resource Server (RS), and Authorization Server (AS).

Readers are also expected to be familiar with the terms and concepts related to CBOR [RFC8949], JSON [RFC8259], the CoAP protocol [RFC7252], CoAP Observe [RFC7641], and the use of hash functions to name objects as defined in [RFC6920].

Note that, unless otherwise indicated, the term "endpoint" is used here following its OAuth definition, aimed at denoting resources such as /token and /introspect at the AS, and /authz-info at the RS. This document does not use the CoAP definition of "endpoint", which is "An entity participating in the CoAP protocol."

This specification also refers to the following terminology.

*Token hash: identifier of an Access Token, in binary format encoding. The token hash has no relation to other possibly used token identifiers, such as the 'cti' (CWT ID) claim of CBOR Web Tokens (CWTs) [RFC8392].

*Token Revocation List (TRL): a collection of token hashes such that the corresponding Access Tokens have been revoked but are not expired yet.

*TRL resource: a resource on the AS, with a TRL as its representation.

*TRL endpoint: an endpoint at the AS associated with the TRL resource. The default name of the TRL endpoint in a url-path is '/revoke/trl'. Implementations are not required to use this name, and can define their own instead.

*Registered device: a device registered at the AS, i.e., as a Client, or an RS, or both. A registered device acts as a caller of the TRL endpoint.

*Administrator: entity authorized to get full access to the TRL at the AS, and acting as a caller of the TRL endpoint. An administrator is not necessarily a registered device as defined above, i.e., a Client requesting Access Tokens or an RS consuming Access Tokens. How the administrator authorization is established and verified is out of the scope of this specification.

*Pertaining Access Token:

  -With reference to an administrator, an Access Token issued by the AS.

  -With reference to a registered device, an Access Token intended to be owned by that device. An Access Token pertains to a Client if the AS has issued the Access Token for that Client following its request. An Access Token pertains to an

RS if the AS has issued the Access Token to be consumed by that RS.

Examples throughout this document are expressed in CBOR diagnostic notation without the tag and value abbreviations.

2. Protocol Overview

This protocol defines how a CoAP-based Authorization Server informs Clients and Resource Servers, i.e., registered devices, about pertaining revoked Access Tokens. How the relationship between a registered device and the AS is established is out of the scope of this specification.

At a high level, the steps of this protocol are as follows.

- Upon startup, the AS creates a single TRL resource. At any point in time, the TRL resource represents the list of all revoked Access Tokens issued by the AS that are not expired yet.

- When a device registers at the AS, it also receives the url-path to the TRL resource.

  After the registration procedure is finished, the registered device can send an Observation Request to the TRL resource as described in [RFC7641], i.e., a GET request including the CoAP Observe Option set to 0 (register). By doing so, the registered device effectively subscribes to the TRL resource, as interested to receive notifications about its update. Upon receiving the request, the AS adds the registered device to the list of observers of the TRL resource.

  At any time, the registered device can send a GET request to the TRL endpoint. When doing so, it can request for: the current list of pertaining revoked Access Tokens (see Section 6); or the most recent TRL updates occurred over the list of pertaining revoked Access Tokens (see Section 7). In either case, the registered device may also rely on an Observation Request for subscribing to the TRL resource as discussed above.

- When an Access Token is revoked, the AS adds the corresponding token hash to the TRL. Also, when a revoked Access Token eventually expires, the AS removes the corresponding token hash from the TRL.

  In either case, after updating the TRL, the AS sends Observe notifications as per [RFC7641]. That is, an Observe notification is sent to each registered device subscribed to the TRL resource and to which the Access Token pertains.

Depending on the specific subscription established through the observation request, the notification provides the current updated list of revoked Access Tokens in the portion of the TRL pertaining to that device (see Section 6), or rather the most recent TRL updates occurred over that list of pertaining revoked Access Tokens (see Section 7).

Further Observe notifications may be sent, consistently with ongoing additional observations of the TRL resource.

*An administrator can access and subscribe to the TRL like a registered device, while getting the full updated representation of the TRL.

Figure 1 shows a high-level overview of the service provided by this protocol. In particular, it shows the Observe notifications sent by the AS to one administrator and four registered devices, upon revocation of the issued Access Tokens t1, t2 and t3, with token hash th1, th2 and th3, respectively. Each dotted line associated with a pair of registered devices indicates the Access Token that they both own.

```
                    +----------------------+
                    | Authorization Server |
                    +-----------o----------+
                    revoke/trl  |  TRL: {th1,th2,th3}
                                |
 +-----------------+------------+------------+------------+
 |                 |            |            |            |
 | th1,th2,th3     | th1,th2    | th1        | th3        | th2,th3
 v                 v            v            v            v
+---------------+ +----------+ +----------+ +----------+ +----------+
| Administrator | | Client 1 | | Resource | | Client 2 | | Resource |
|               | |          | | Server 1 | |          | | Server 2 |
+---------------+ +----------+ +----------+ +----------+ +----------+
                     :    :        :           :            :    :
                     :    :   t1   :           :     t3     :    :
                     :    :........:           :............:    :
                     :                   t2                      :
                     :...........................................:
```

Figure 1: Protocol Overview

Appendix C provides examples of the protocol flow and message exchange between the AS and a registered device.

3. Token Hash

The token hash of an Access Token is computed as follows.

1. The AS defines ENCODED_TOKEN, as the content of the 'access_token' parameter in the AS-to-Client response (see Section 5.8.2 of [RFC9200]), where the Access Token was included and provided to the requesting Client.

   Note that the content of the 'access_token' parameter is either:

   - A CBOR byte string, if the Access Token was transported using CBOR. With reference to the example in Figure 2, and assuming the string's length in bytes to be 119 (i.e., 0x77 in hexadecimal), then ENCODED_TOKEN takes the bytes {0x58 0x77 0xd0 0x83 0x44 0xa1 ...}, i.e., the raw content of the 'access_token' parameter.

   - A text string, if the Access Token was transported using JSON. With reference to the example in Figure 3, ENCODED_TOKEN takes "2YotnFZFEjr1zCsicMWpAA", i.e., the raw content of the 'access_token' parameter.

2. The AS defines HASH_INPUT as follows.

   - If CBOR was used to transport the Access Token (as a CWT or JWT), HASH_INPUT takes the same value of ENCODED_TOKEN.

   - If JSON was used to transport the Access Token (as a CWT or JWT), HASH_INPUT takes the serialization of ENCODED_TOKEN.

     In either case, HASH_INPUT results in the binary representation of the content of the 'access_token' parameter from the AS-to-Client response.

3. The AS generates a hash value of HASH_INPUT as per Section 6 of [RFC6920]. The resulting output in binary format is used as the token hash. Note that the used binary format embeds the identifier of the used hash function, in the first byte of the computed token hash.

   The specifically used hash function MUST be collision-resistant on byte-strings, and MUST be selected from the "Named Information Hash Algorithm" Registry [Named.Information.Hash.Algorithm].

   The AS specifies the used hash function to registered devices during their registration procedure (see Section 9).

```
2.01 Created
Content-Format: application/ace+cbor
Max-Age: 85800
Payload:
{
   "access_token" : h'd08344a1 ...
    (remainder of the Access Token omitted for brevity) ...',
   "token_type" : pop,
   "expires_in" : 86400,
   "profile" : coap_dtls,
   (remainder of the response omitted for brevity)
}
```

Figure 2: Example of AS-to-Client response using CBOR

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache
Payload:
{
   "access_token" : "2YotnFZFEjr1zCsicMWpAA ...
    (remainder of the Access Token omitted for brevity) ...",
   "token_type" : "pop",
   "expires_in" : 86400,
   "profile" : "coap_dtls",
   (remainder of the response omitted for brevity)
}
```

Figure 3: Example of AS-to-Client response using JSON

4. The TRL Resource

Upon startup, the AS creates a single TRL resource, encoded as a CBOR array.

Each element of the array is a CBOR byte string, with value the token hash of an Access Token. The order of the token hashes in the CBOR array is irrelevant, and the CBOR array MUST be treated as a set in which the order of elements has no significant meaning.

The TRL is initialized as empty, i.e., the initial content of the TRL resource representation MUST be an empty CBOR array.

4.1. Update of the TRL Resource

The AS updates the TRL in the following two cases.

*When a non-expired Access Token is revoked, the token hash of the Access Token is added to the TRL resource representation. That is, a CBOR byte string with the token hash as its value is added to the CBOR array used as TRL resource representation.

- When a revoked Access Token expires, the token hash of the Access Token is removed from the TRL resource representation. That is, the CBOR byte string with the token hash as its value is removed from the CBOR array used as TRL resource representation.

5. The TRL Endpoint

Consistent with Section 6.5 of [RFC9200], all communications between a caller of the TRL endpoint and the AS MUST be encrypted, as well as integrity and replay protected. Furthermore, responses from the AS to the caller MUST be bound to the caller's request.

Following a request to the TRL endpoint, the messages defined in this document that the AS sends as response use Content-Format "application/ace-trl+cbor". Their payload is formatted as a CBOR map, and the CBOR values for the parameters included therein are defined in Section 11.

The AS MUST implement measures to prevent access to the TRL endpoint by entities other than registered devices and authorized administrators.

The TRL endpoint supports only the GET method, and allows two types of query of the TRL.

- Full query: the AS returns the token hashes of the revoked Access Tokens currently in the TRL and pertaining to the requester.

  The AS MUST support this type of query. The processing of a full query and the related response format are defined in Section 6.

- Diff query: the AS returns a list of diff entries. Each diff entry is related to one of the most recent updates, in the portion of the TRL pertaining to the requester.

  The entry associated with one of such updates contains a list of token hashes, such that: i) the corresponding revoked Access Tokens pertain to the requester; and ii) they were added to or removed from the TRL at that update.

  The AS MAY support this type of query. In such a case, the AS maintains the history of updates to the TRL resource as defined in Section 5.1. The processing of a diff query and the related response format are defined in Section 7.

If it supports diff queries, the AS MAY additionally support its "Cursor" extension, which has two benefits. First, the AS can avoid excessively big latencies when several diff entries have to be transferred, by delivering one adjacent subset at the time, in different diff query responses. Second, a requester can retrieve diff entries associated with TRL updates that, even if not the most recent ones, occurred after a TRL update indicated as reference point.

If it supports the "Cursor" extension, the AS stores additional information when maintaining the history of updates to the TRL resource, as defined in Section 5.1.1. Also, the processing of full query requests and diff query requests, as well as the related response format, are further extended as defined in Section 8.

Appendix B provides an aggregated overview of the parameters used by the TRL endpoint, when the AS supports diff queries and the "Cursor" extension.

5.1. Supporting Diff Queries

If the AS supports diff queries, it is able to transfer a list of diff entries, as a series of TRL updates. That is, when replying to a diff query performed by a requester, the AS specifies the most recent updates to the portion of the TRL pertaining to that requester.

The following defines how the AS builds and maintains consistent histories of TRL updates for each registered device and administrator, hereafter referred to as requesters.

For each requester, the AS maintains an update collection of maximum MAX_N series items, where MAX_N is a pre-defined, constant positive integer. The AS MUST keep track of the MAX_N most recent updates to the portion of the TRL that pertains to each requester. The AS SHOULD provide requesters with the value of MAX_N, upon their registration (see Section 9).

The series items in the update collection MUST be strictly ordered in a chronological fashion. That is, at any point in time, the current first series item is the one least recently added to the update collection and still retained by the AS, while the current last series item is the one most recently added to the update collection. The particular method used to achieve this is implementation-specific.

Each time the TRL changes, the AS performs the following operations for each requester.

1. The AS considers the portion of the TRL pertaining to that requester. If the TRL portion is not affected by this TRL

update, the AS stops the processing for that requester. Otherwise, the AS moves to step 2.

2. The AS creates two sets "trl_patch" of token hashes, i.e., one "removed" set and one "added" set, as related to this TRL update.

3. The AS fills the two sets with the token hashes of the removed and added Access Tokens, respectively, from/to the TRL portion considered at step 1.

4. The AS creates a new series item, which includes the two sets from step 3.

5. If the update collection associated with the requester currently includes MAX_N series items, the AS MUST delete the oldest series item in the update collection.

   This occurs when the number of TRL updates pertaining to the requester and currently stored at the AS is equal to MAX_N.

6. The AS adds the series item to the update collection associated with the requester, as the most recent one.

5.1.1. Supporting the "Cursor" Extension

If it supports the "Cursor" extension for diff queries, the AS performs also the following actions.

The AS defines the constant, unsigned integer MAX_INDEX <= ((2 ** 64) - 1), where "**" is the exponentiation operator. In particular, the value of MAX_INDEX is REQUIRED to be at least (MAX_N - 1), and is RECOMMENDED to be at least ((2 ** 32) - 1). Note that MAX_INDEX is practically expected to be order of magnitudes greater than MAX_N.

When maintaining the history of updates to the TRL resource, the following applies separately for each update collection.

- Each series item X in the update collection is also associated with an unsigned integer 'index', whose minimum value is 0 and whose maximum value is MAX_INDEX. The first series item ever added to the update collection MUST have 'index' with value 0.

  If i_X is the value of 'index' associated with a series item X, then the following series item Y will take 'index' with value i_Y = (i_X + 1) % (MAX_INDEX + 1). That is, after having added a series item whose associated 'index' has value MAX_INDEX, the next added series item will result in a wrap-around of the 'index' value, and will thus take 'index' with value 0.

For example, assuming MAX_N = 3, the values of 'index' in the update collection chronologically evolve as follows, as new series items are added and old series items are deleted.

- ...

- (i_A = MAX_INDEX - 2, i_B = MAX_INDEX - 1, i_C = MAX_INDEX)

- (i_B = MAX_INDEX - 1, i_C = MAX_INDEX, i_D = 0)

- (i_C = MAX_INDEX, i_D = 0, i_E = 1)

- (i_D = 0, i_E = 1, i_F = 2)

- ...

*The unsigned integer 'last_index' is also defined, with minimum value 0 and maximum value MAX_INDEX.

If the update collection is empty (i.e., no series items have been added yet), the value of 'last_index' is not defined. If the update collection is not empty, 'last_index' has the value of 'index' currently associated with the latest added series item in the update collection.

That is, after having added V series items to the update collection, the last and most recently added series item has 'index' with value 'last_index' = (V - 1) % (MAX_INDEX + 1).

As long as a wrap-around of the 'index' value has not occurred, the value of 'last_index' is the absolute counter of series items added to that update collection until and including V, minus 1.

When processing a diff query using the "Cursor" extension, the values of 'index' are used as cursor information, as defined in Section 8.2.

For each update collection, the AS also defines a constant, positive integer MAX_DIFF_BATCH <= MAX_N, whose value specifies the maximum number of diff entries to be included in a single diff query response. The specific value depends on the specific registered device or administrator associated with the update collection in question. If supporting the "Cursor" extension, the AS SHOULD provide registered devices and administrators with the value of MAX_DIFF_BATCH, upon their registration (see Section 9).

5.2. Query Parameters

A GET request to the TRL endpoint can include the following query parameters. The AS MUST silently ignore unknown query parameters.

- 'diff': if included, it indicates to perform a diff query of the TRL (see Section 7). Its value MUST be either:

  - the integer 0, indicating that a (notification) response should include as many diff entries as the AS can provide in the response; or

  - a positive integer strictly greater than 0, indicating the maximum number of diff entries that a (notification) response should include.

  If the AS does not support diff queries, it ignores the 'diff' query parameter when present in the GET request, and proceeds like when processing a full query of the TRL (see Section 6).

  Otherwise, the AS MUST return a 4.00 (Bad Request) response in case the 'diff' query parameter of the GET request specifies a value other than 0 or than a positive integer. The response MUST have Content-Format "application/ace-trl+cbor". The payload of the response is a CBOR map, which MUST include the 'error' parameter with value 0 ("Invalid parameter value") and MAY include the 'error_description' parameter to provide additional context.

- 'cursor': if included, it indicates to perform a diff query of the TRL together with the "Cursor" extension, as defined in Section 8.2. Its value MUST be either 0 or a positive integer.

  If included, the 'cursor' query parameter specifies an unsigned integer value that was provided by the AS in a previous response from the TRL endpoint (see Section 8.1, Section 8.2.2 and Section 8.2.3).

  If the AS does not support the "Cursor" extension, it ignores the 'cursor' query parameter when present in the GET request. In such a case, the AS proceeds: i) like when processing a diff query of the TRL (see Section 7), if it supports diff queries and the 'diff' query parameter is present in the GET request; or ii) like when processing a full query of the TRL (see Section 6) otherwise.

  If the AS supports both diff queries and the "Cursor" extension, and the GET request specifies the 'cursor' query parameter, then the AS MUST return a 4.00 (Bad Request) response in case any of the following conditions holds.

- The GET request does not specify the 'diff' query parameter.

  The 'error' parameter within the CBOR map carried in the response payload MUST have value 1 ("Invalid set of parameters").

- The 'cursor' query parameter has a value other than 0 or than a positive integer.

  The 'error' parameter within the CBOR map carried in the response payload MUST have value 0 ("Invalid parameter value").

- The 'cursor' query parameter has a value strictly greater than MAX_INDEX (see Section 5.1.1).

  The 'error' parameter within the CBOR map carried in the response payload MUST have value 0 ("Invalid parameter value"). The CBOR map MUST also include the 'cursor' parameter, which MUST specify either: the CBOR simple value "null" (0xf6), if the update collection associated with the requester is empty; or the corresponding current value of 'last_index' otherwise.

- All of the following hold: the update collection associated with the requester is not empty; no wrap-around of its 'index' value has occurred; and the 'cursor' query parameter has a value strictly greater than the current 'last_index' on the update collection (see Section 5.1.1).

  The 'error' parameter within the CBOR map carried in the response payload MUST have value 2 ("Out of bound cursor value"). The CBOR map MUST also include the 'cursor' parameter, which MUST specify the current value of 'last_index' for the update collection associated with the requester.

The 4.00 (Bad Request) response MUST have Content-Format "application/ace-trl+cbor". The payload of the response MUST be a CBOR map, which MUST include the 'error' parameter and MAY include the 'error_description' parameter to provide additional context.

6. Full Query of the TRL

In order to produce a (notification) response to a GET request asking for a full query of the TRL, the AS performs the following actions.

1. From the current TRL resource representation, the AS builds a set HASHES, such that:

   - If the requester is a registered device, HASHES specifies the token hashes of the Access Tokens pertaining to that registered device. The AS can use the authenticated identity of the registered device to perform the necessary filtering on the TRL resource representation.

   - If the requester is an administrator, HASHES specifies all the token hashes in the current TRL resource representation.

2. The AS sends a 2.05 (Content) response to the requester. The response MUST have Content-Format "application/ace-trl+cbor". The payload of the response is a CBOR map, which MUST be formatted as follows.

   - The 'full_set' parameter MUST be included and specifies a CBOR array 'full_set_value'. Each element of 'full_set_value' specifies one of the token hashes from the set HASHES, encoded as a CBOR byte string. If the set HASHES is empty, the 'full_set' parameter specifies the empty CBOR array.

     The order of the token hashes in the CBOR array is irrelevant, i.e., the CBOR array MUST be treated as a set in which the order of elements has no significant meaning.

   - The 'cursor' parameter MUST be included if the AS supports both diff queries and the related "Cursor" extension (see Section 5.1 and Section 5.1.1). Its value is specified according to what is defined in Section 8.1, and provides the requester with information for performing a follow-up diff query using the "Cursor" extension (see Section 8.2).

     If the AS does not support both diff queries and the "Cursor" extension, this parameter MUST NOT be included. In case the requester does not support both diff queries and the "Cursor" extension, it MUST silently ignore the 'cursor' parameter if present.

Figure 4 provides the CDDL definition [RFC8610] of the CBOR array 'full_set_value' specified in the response from the AS, as value of the 'full_set' parameter.

```
token_hash = bytes
full_set_value = [* token_hash]
```

Figure 4: CDDL definition of 'full_set_value'

Figure 5 shows an example of response from the AS, following a full query request to the TRL endpoint. In this example, the AS does not support the "Cursor" extension (if it supports diff queries at all), hence the 'cursor' parameter is not included in the payload of the response. Also, full token hashes are omitted for brevity.

```
2.05 Content
Content-Format: application/ace-trl+cbor
Payload:
{
   "full_set" : [
     h'01fa51cc ... ', h'01748190 ... '
   ]
}
```

Figure 5: Example of response following a Full Query request to the TRL endpoint

7. Diff Query of the TRL

In order to produce a (notification) response to a GET request asking for a diff query of the TRL, the AS performs the following actions.

Note that, if the AS supports both diff queries and the related "Cursor" extension, the steps 3 and 4 defined below are extended as defined in Section 8.2.

1. The AS defines the positive integer NUM as follows. If the value N specified in the 'diff' query parameter in the GET request is equal to 0 or greater than the pre-defined positive integer MAX_N (see Section 5.1), then NUM takes the value of MAX_N. Otherwise, NUM takes N.

2. The AS determines U = min(NUM, SIZE), where SIZE <= MAX_N is the number of TRL updates pertaining to the requester and currently stored at the AS.

3. The AS prepares U diff entries. If U is equal to 0 (e.g., because SIZE is equal to 0 at step 2), then no diff entries are prepared.

The prepared diff entries are related to the U most recent TRL updates pertaining to the requester, as maintained in the update collection for that requester (see Section 5.1). In particular, the first diff entry refers to the most recent of such updates, the second diff entry refers to the second from last of such updates, and so on.

Each diff entry is a CBOR array 'diff_entry', which includes the following two elements.

- The first element is a CBOR array 'removed'. Each element of the array is a CBOR byte string, with value the token hash of an Access Token such that: it pertained to the requester; and it was removed from the TRL during the update associated with the diff entry.

- The second element is a CBOR array 'added'. Each element of the array is a CBOR byte string, with value the token hash of an Access Token such that: it pertains to the requester; and it was added to the TRL during the update associated with the diff entry.

The order of the token hashes in the CBOR arrays 'removed' and 'added' is irrelevant. That is, the CBOR arrays 'removed' and 'added' MUST be treated as a set in which the order of elements has no significant meaning.

4. The AS prepares a 2.05 (Content) response for the requester. The response MUST have Content-Format "application/ace-trl+cbor". The payload of the response is a CBOR map, which MUST be formatted as follows.

   - The 'diff_set' parameter MUST be present and specifies a CBOR array 'diff_set_value' of U elements. Each element of 'diff_set_value' specifies one of the CBOR arrays 'diff_entry' prepared above as diff entry. Note that U might have value 0, in which case 'diff_set_value' is the empty CBOR array.

     Within 'diff_set_value', the CBOR arrays 'diff_entry' MUST be sorted to reflect the corresponding updates to the TRL in reverse chronological order. That is, the first 'diff_entry' element of 'diff_set_value' relates to the most recent update to the portion of the TRL pertaining to the requester. The second 'diff_entry' element relates to the second from last most recent update to that portion, and so on.

   - The 'cursor' parameter and the 'more' parameter MUST be included if the AS supports both diff queries and the related "Cursor" extension (see Section 5.1.1). Their values are specified according to what is defined in Section 8.2, and provide the requester with information for performing a follow-up query to the TRL endpoint (see Section 8.2).

If the AS does not support both diff queries and the "Cursor" extension, these parameters MUST NOT be included. In case the requester does not support both diff queries and the "Cursor" extension, it MUST silently ignore the 'cursor' parameter and the 'more' parameter if present.

Figure 6 provides the CDDL definition [RFC8610] of the CBOR array 'diff_set_value' specified in the response from the AS, as value of the 'diff_set' parameter.

```
token_hash = bytes
trl_patch = [* token_hash]
diff_entry = [removed: trl_patch, added: trl_patch]
diff_set_value = [* diff_entry]
```

Figure 6: CDDL definition of 'diff_set_value'

Figure 7 shows an example of response from the AS, following a Diff Query request to the TRL endpoint, where U = 3 diff entries are specified. In this example, the AS does not support the "Cursor" extension, hence the 'cursor' parameter and the 'more' parameter are not included in the payload of the response. Also, full token hashes are omitted for brevity.

```
2.05 Content
Content-Format: application/ace-trl+cbor
Payload:
{
   "diff_set" : [
     [
       [ h'01fa51cc ... ', h'01748190 ... '],
       [ h'01cdf1ca ... ', h'01be41a6 ... ']
     ],
     [
       [ h'0144dd12 ... ', h'01231fff ... '],
       []
     ],
     [
       [],
       [ h'01ca986f ... ', h'01fe1a2b ... ']
     ]
   ]
}
```

Figure 7: Example of response following a Diff Query request to the TRL endpoint

Appendix A discusses how performing a diff query of the TRL is in fact a usage example of the Series Transfer Pattern defined in [I-D.bormann-t2trg-stp].

8. Response Messages when Using the "Cursor" Extension

If it supports both diff queries and the "Cursor" extension, the AS composes a response to a full query request or diff query request as defined in Section 8.1 and Section 8.2, respectively.

The exact format of the response depends on the request being a full query or diff query request, on the presence of the 'cursor' query parameter in the diff query request, and on the current status of the update collection associated with the requester.

Error handling and the possible resulting error responses are as defined in Section 5.2.

8.1. Response to Full Query

When processing a full query request to the TRL endpoint, the AS composes a response as defined in Section 6.

In particular, the 'cursor' parameter included in the CBOR map carried in the response payload specifies either the CBOR simple value "null" (0xf6) or a CBOR unsigned integer.

The 'cursor' parameter MUST specify the CBOR simple value "null" in case there are currently no TRL updates pertinent to the requester, i.e., the update collection for that requester is empty. This is the case from when the requester registers at the AS until a first update pertaining to that requester occurs to the TRL.

Otherwise, the 'cursor' parameter MUST specify a CBOR unsigned integer. This MUST take the 'index' value of the last series item in the update collection associated with the requester (see Section 5.1.1), as corresponding to the most recent update pertaining to the requester occurred to the TRL.

8.2. Response to Diff Query

When processing a diff query request to the TRL endpoint, the AS composes a response as defined in the following.

8.2.1. Empty Collection

If the update collection associated with the requester has no elements, the AS returns a 2.05 (Content) response. The response MUST have Content-Format "application/ace-trl+cbor" and its payload MUST be a CBOR map formatted as follows.

- The 'diff_set' parameter MUST be included and specifies the empty CBOR array.

- The 'cursor' parameter MUST be included and specifies the CBOR simple value "null" (0xf6).

- The 'more' parameter MUST be included and specifies the CBOR simple value "false" (0xf4).

Note that the above applies when the update collection associated with the requester has no elements, regardless whether the 'cursor' query parameter is included or not in the diff query request, and irrespective of the specified unsigned integer value if present.

8.2.2. Cursor Not Specified in the Diff Query Request

If the update collection associated with the requester is not empty and the diff query request does not include the 'cursor' query parameter, the AS performs the same actions defined in Section 7, with the following differences.

- At step 3, the AS considers the value MAX_DIFF_BATCH (see Section 5.1.1), and prepares L = min(U, MAX_DIFF_BATCH) diff entries.

  If U <= MAX_DIFF_BATCH, the prepared diff entries are the last series items in the update collection associated with the requester, corresponding to the L most recent TRL updates pertaining to the requester.

  If U > MAX_DIFF_BATCH, the prepared diff entries are the eldest of the last U series items in the update collection associated with the requester, as corresponding to the first L of the U most recent TRL updates pertaining to the requester.

- At step 4, the CBOR map to carry in the payload of the 2.05 (Content) response MUST be formatted as follows.

  - The 'diff_set' parameter MUST be present and specifies a CBOR array 'diff_set_value' of L elements. Each element of 'diff_set_value' specifies one of the CBOR arrays 'diff_entry' prepared as diff entry.

- The 'cursor' parameter MUST be present and specifies a CBOR unsigned integer. This MUST take the 'index' value of the series item of the update collection included as first diff entry in the 'diff_set_value' CBOR array, which is specified by the 'diff_set' parameter. That is, the 'cursor' parameter takes the 'index' value of the series item in the update collection corresponding to the most recent update pertaining to the requester and returned in this diff query response.

  Note that the 'cursor' parameter takes the same 'index' value of the last series item in the update collection when U <= MAX_DIFF_BATCH.

- The 'more' parameter MUST be present and MUST specify the CBOR simple value "false" (0xf4) if U <= MAX_DIFF_BATCH, or the CBOR simple value "true" (0xf5) otherwise.

  If the 'more' parameter has value "true", the requester can send a follow-up diff query request including the 'cursor' query parameter, with the same value of the 'cursor' parameter specified in this diff query response. As defined in Section 8.2.3, this would result in the AS transferring the following subset of series items as diff entries, thus resuming from where interrupted in the previous transfer.

8.2.3. Cursor Specified in the Diff Query Request

If the update collection associated with the requester is not empty and the diff query request includes the 'cursor' query parameter with value P, the AS proceeds as follows, depending on which of the following two cases hold.

* Case A - The series item X with 'index' having value P and the series item Y with 'index' having value (P + 1) % (MAX_INDEX + 1) are both not found in the update collection associated with the requester. This occurs when the item Y (and possibly further ones after it) has been previously removed from the history of updates for that requester (see step 5 at Section 5.1).

  In this case, the AS returns a 2.05 (Content) response. The response MUST have Content-Format "application/ace-trl+cbor" and its payload MUST be a CBOR map formatted as follows.

  - The 'diff_set' parameter MUST be included and specifies the empty CBOR array.

  - The 'cursor' parameter MUST be included and specifies the CBOR simple value "null" (0xf6).

- The 'more' parameter MUST be included and specifies the CBOR simple value "true" (0xf5).

With the combination ('cursor', 'more') = ("null", "true"), the AS is signaling that the update collection is in fact not empty, but that one or more series items have been lost due to their removal. These include the item with 'index' value (P + 1) % (MAX_INDEX + 1), that the requester wished to obtain as the first one following the specified reference point with 'index' value P.

When receiving this diff query response, the requester should send a new full query request to the AS. A successful response provides the requester with the full, current pertaining portion of the TRL, as well as with a valid value of the 'cursor' parameter (see Section 8.1) to be possibly used as query parameter in a following diff query request.

*Case B - The series item X with 'index' having value P is found in the update collection associated with the requester; or the series item X is not found and the series item Y with 'index' having value (P + 1) % (MAX_INDEX + 1) is found in the update collection associated with the requester.

In this case, the AS performs the same actions defined in Section 7, with the following differences.

- At step 3, the AS considers the value MAX_DIFF_BATCH (see Section 5.1.1), and prepares L = min(SUB_U, MAX_DIFF_BATCH) diff entries, where SUB_U = min(NUM, SUB_SIZE), and SUB_SIZE is the number of series items in the update collection starting from and including the series item added immediately after X. If L is equal to 0 (e.g., because SUB_U is equal to 0), then no diff entries are prepared.

  If SUB_U <= MAX_DIFF_BATCH, the prepared diff entries are the last series items in the update collection associated with the requester, corresponding to the L most recent TRL updates pertaining to the requester.

  If SUB_U > MAX_DIFF_BATCH, the prepared diff entries are the eldest of the last SUB_U series items in the update collection associated with the requester, corresponding to the first L of the SUB_U most recent TRL updates pertaining to the requester.

- At step 4, the CBOR map to carry in the payload of the 2.05 (Content) response MUST be formatted as follows.

  oThe 'diff_set' parameter MUST be present and specifies a CBOR array 'diff_set_value' of L elements. Each element of 'diff_set_value' specifies one of the CBOR arrays 'diff_entry' prepared as diff entry. Note that L might have value 0, in which case 'diff_set_value' is the empty CBOR array.

- The 'cursor' parameter MUST be present and MUST specify a CBOR unsigned integer. In particular:

  - If L is equal to 0, i.e., the series item X is the last one in the update collection, then the 'cursor' parameter MUST take the same 'index' value of the last series item in the update collection.

  - If L is different than 0, then the 'cursor' parameter MUST take the 'index' value of the series element of the update collection included as first diff entry in the 'diff_set' CBOR array. That is, the 'cursor' parameter takes the 'index' value of the series item in the update collection corresponding to the most recent update pertaining to the requester and returned in this diff query response.

  Note that the 'cursor' parameter takes the same 'index' value of the last series item in the update collection when SUB_U <= MAX_DIFF_BATCH.

- The 'more' parameter MUST be present and MUST specify the CBOR simple value "false" (0xf4) if SUB_U <= MAX_DIFF_BATCH, or the CBOR simple value "true" (0xf5) otherwise.

  If 'more' has value "true", the requester can send a follow-up diff query request including the 'cursor' query parameter, with the same value of the 'cursor' parameter specified in this diff query response. This would result in the AS transferring the following subset of series items as diff entries, thus resuming from where interrupted in the previous transfer.

9. Registration at the Authorization Server

During the registration process at the AS, an administrator or a registered device receives the following information as part of the registration response.

*The url-path to the TRL endpoint at the AS.

*The hash function used to compute token hashes. This is specified as an integer or a text string, taking value from the "ID" or "Hash Name String" column of the "Named Information Hash

Algorithm" Registry [Named.Information.Hash.Algorithm], respectively.

- Optionally, a positive integer MAX_N, if the AS supports diff queries of the TRL resource (see Section 5.1 and Section 7).

- Optionally, a positive integer MAX_DIFF_BATCH, if the AS supports diff queries of the TRL resource as well as the related "Cursor" extension (see Section 5.1.1 and Section 8).

Further details about the registration process at the AS are out of scope for this specification. Note that the registration process is also out of the scope of the ACE framework for Authentication and Authorization (see Section 5.5 of [RFC9200]).

10. Notification of Revoked Access Tokens

Once completed the registration procedure at the AS, the administrator or registered device can send a GET request to the TRL resource at the AS. The request can express the wish for a full query (see Section 6) or a diff query (see Section 7) of the TRL. Also, the request can include the CoAP Observe Option set to 0 (register), in order to start an observation of the TRL resource as per Section 3.1 of [RFC7641].

In case the request is successfully processed, the AS replies with a response specifying the CoAP response code 2.05 (Content). In particular, if the AS does not support both diff queries and the related "Cursor" extension (see Section 5.1 and Section 5.1.1), then the payload of the response is formatted as defined in Section 6 or in Section 7, in case the GET request has yielded the execution of a full query or of a diff query of the TRL, respectively. Instead, if the AS supports both diff queries and the related "Cursor" extension, then the payload of the response is formatted as defined in Section 8.

When the TRL is updated (see Section 4.1), the AS sends Observe notifications to the observers whose pertaining portion of the TRL has changed. Observe notifications are sent as per Section 4.2 of [RFC7641]. If supported by the AS, an observer may configure the behavior according to which the AS sends those Observe notifications. To this end, a possible way relies on the conditional control attribute "c.pmax" defined in [I-D.ietf-core-conditional-attributes], which can be included as a "name=value" query parameter in an Observation Request. This ensures that no more than c.pmax seconds elapse between two consecutive notifications sent to that observer, regardless whether the TRL resource has changed or not.

Following a first exchange with the AS, an administrator or a registered device can send additional GET (Observation) requests to the TRL endpoint at any time, analogously to what is defined above. When doing so, the caller of the TRL endpoint can perform a full query (see Section 6) or a diff query (see Section 7) of the TRL.

10.1. Handling of Access Tokens and Token Hashes

When receiving a response from the TRL endpoint, a registered device MUST expunge every stored Access Token associated with a token hash specified in the response. In case the registered device is an RS, it MUST store the token hash.

An RS MUST NOT accept and store an Access Token, if the corresponding token hash is among the currently stored ones.

An RS stores a token hash th1 corresponding to an Access Token t1 until both the following conditions hold.

*The RS has received and seen t1, irrespective of having accepted and stored it.

*The RS has gained knowledge that t1 has expired. This can be achieved, e.g., through the following means.

  -A response from the TRL endpoint indicating that t1 has expired.

  -The value of the 'exp' claim specified in t1 indicates that t1 has expired.

  -The locally determined expiration time for t1 has passed, based on the time at the RS when t1 was first accepted and on the value of its 'exi' claim.

  -The result of token introspection performed on t1 (see Section 5.9 of [RFC9200]), if supported by both the RS and the AS.

The RS MUST NOT delete the stored token hashes whose corresponding Access Tokens do not fulfill the two conditions above, unless it becomes necessary due to memory limitations. In such a case, the RS MUST delete the earliest stored token hashes first.

Retaining the stored token hashes as specified above limits the impact from a (dishonest) Client whose pertaining Access Token: i) specifies the 'exi' claim; ii) is uploaded at the RS for the first time after it has been revoked and later expired; and iii) has the sequence number encoded in the 'cti' claim not greater than the highest sequence number among the expired Access Tokens specifying the 'exi' claim for the RS (see Section 5.10.3 of [RFC9200]). That is, the RS would not accept such a revoked and expired Access Token as long as it stores the corresponding token hash.

In order to further limit such a risk, when receiving an Access Token that specifies the 'exi' claim and for which a corresponding token hash is not stored, the RS can introspect the Access Token (see Section 5.9 of [RFC9200]), if token introspection is implemented by both the RS and the AS.

When, due to the stored and corresponding token hash th2, an Access Token t2 that includes the 'exi' claim is expunged or is not accepted upon its upload, the RS retrieves the sequence number sn2 encoded in the 'cti' claim (see Section 5.10.3 of [RFC9200]). Then, the RS stores sn2 as associated with th2. If expunging or not accepting t2 yields the deletion of th2 as per the two conditions specified above, then the RS MUST associate sn2 with th2 before continuing with the deletion of th2.

When deleting any token hash, the RS checks whether the token hash is associated with a sequence number sn_th. In such a case, the RS checks whether sn_th is greater than the highest sequence number sn* among the expired Access Tokens specifying the 'exi' claim for the RS. If that is the case, sn* MUST take the value of sn_th.

By virtue of what is defined in Section 5.10.3 of [RFC9200], this ensures that, following the deletion of the token hash associated with an Access Token specifying the 'exi' claim and uploaded for the first time after it has been revoked and later expired, the RS will not accept the Access Token at that point in time or in the future.

11. ACE Token Revocation List Parameters

This specification defines a number of parameters that can be transported in the response from the TRL endpoint, when the response payload is a CBOR map. Note that such a response MUST use the Content-Format "application/ace-trl+cbor" defined in Section 14.2 of this specification.

The table below summarizes them, and specifies the CBOR value to use as abbreviation instead of the full descriptive name.

Name	CBOR Value	CBOR Type
full_set	0	array
diff_set	1	array
cursor	2	unsigned integer / simple value "null"
more	3	simple value "false" / simple value "true"
error	4	integer
error_description	5	text string

Figure 8: CBOR abbreviations for the ACE Token Revocation List parameters

12. ACE Token Revocation List Error Identifiers

This specification defines a number of values that the AS can include as error identifiers, in the 'error' parameter of an error response from the TRL endpoint. This applies to error responses whose payload is a CBOR map and whose Content-Format is "application/ace-trl+cbor".

Value	Description
0	Invalid parameter value
1	Invalid set of parameters
2	Out of bound cursor value

Figure 9: ACE Token Revocation List Error Identifiers

13. Security Considerations

Security considerations are inherited from the ACE framework for Authentication and Authorization [RFC9200], from [RFC8392] as to the usage of CWTs, from [RFC7519] as to the usage of JWTs, from [RFC7641] as to the usage of CoAP Observe, and from [RFC6920] with regard to computing the token hashes. The following considerations also apply.

13.1. Content Retrieval from the TRL Resource

The AS MUST ensure that each registered device can access and retrieve only its pertaining portion of the TRL. To this end, the AS can perform the required filtering based on the authenticated identity of the registered device, i.e., a (non-public) identifier that the AS can securely relate to the registered device and the secure association that they use to communicate.

Disclosing any information about revoked Access Tokens to entities other than the intended registered devices may result in privacy concerns. Therefore, the AS MUST ensure that, other than registered devices accessing their own pertaining portion of the TRL, only authorized and authenticated administrators can retrieve the full TRL. To this end, the AS may rely on an access control list or similar.

13.2. Size of the TRL Resource

If many non-expired Access Tokens associated with a registered device are revoked, the pertaining portion of the TRL could grow to a size bigger than what the registered device is prepared to handle upon reception, especially if relying on a full query of the TRL resource (see Section 6).

This could be exploited by attackers to negatively affect the behavior of a registered device. Issuing Access Tokens with not too long expiration time could help reduce the size of a TRL, but an AS SHOULD take measures to limit this size.

13.3. Communication Patterns

The communication about revoked Access Tokens presented in this specification is expected to especially rely on CoAP Observe notifications sent from the AS to a registered device. The suppression of those notifications by an external attacker that has access to the network would prevent registered devices from ever knowing that their pertaining Access Tokens have been revoked.

In order to avoid this, a registered device SHOULD NOT rely solely on the CoAP Observe notifications. In particular, a registered device SHOULD also regularly poll the AS for the most current information about revoked Access Tokens, by sending GET requests to the TRL endpoint according to a related application policy.

13.4. Request of New Access Tokens

If a Client stores an Access Token that it still believes to be valid, and it accordingly attempts to access a protected resource at the RS, the Client migth still receive an unprotected 4.01 (Unauthorized) response from the RS.

This can be due to different reasons. For example, the Access Token has actually been revoked and the Client is not aware about that yet, while the RS has gained knowledge about that and has expunged the Access Token. Also, an on-path, active adversary might have injected a forged 4.01 (Unauthorized) response.

In either case, if the Client believes that the Access Token is still valid, it SHOULD NOT immediately ask for a new Access Token to the Autherization Server upon receiving a 4.01 (Unauthorized) response from the RS. Instead, the Client SHOULD send a request to the TRL resource at the AS, in order to assert whether the Access Token is still valid. If this is the case, the Client SHOULD NOT ask for a new Access Token.

13.5. Dishonest Clients

A dishonest Client may attempt to exploit its early knowledge about a revoked Access Token, in order to illegitimately continue accessing a protected resource at the RS beyond the Access Token revocation.

That is, the Client might gain knowledge about the revocation of an Access Token considerably earlier than the RS, e.g., if the Client relies on CoAP Observe to access the TRL resource at the AS, while the RS relies only on polling through individual requests.

This makes the RS vulnerable during a time interval that starts when the Client gains knowledge of the revoked Access Token and ends when the RS expunges the Access Token, e.g., after having gained knowledge of its revocation. During such a time interval, the Client would be able to illegitimately access protected resources at the RS, if this still retains the Access Token without knowing about its revocation yet.

In order to mitigate the risk of such an abuse, if an RS relies solely on polling through individual requests to the TRL resource, the RS SHOULD enforce an adequate trade-off between the polling frequency and the maximum length of the vulnerable time window.

14. IANA Considerations

This document has the following actions for IANA.

Note to RFC Editor: Please replace all occurrences of "[RFC-XXXX]" with the RFC number of this specification and delete this paragraph.

14.1. Media Type Registrations

IANA is asked to register the media type "application/ace-trl+cbor" for messages of the protocol defined in this document encoded in CBOR. This registration follows the procedures specified in [RFC6838].

Type name: application

Subtype name: ace-trl+cbor

Required parameters: N/A

Optional parameters: N/A

Encoding considerations: Must be encoded as a CBOR map containing the protocol parameters defined in [RFC-XXXX].

Security considerations: See Section 13 of this document.

Interoperability considerations: N/A

Published specification: [RFC-XXXX]

Applications that use this media type: The type is used by Authorization Servers, Clients and Resource Servers that support the notification of revoked Access Tokens, according to a Token Revocation List maintained by the Authorization Server as specified in [RFC-XXXX].

Fragment identifier considerations: N/A

Additional information: N/A

Person & email address to contact for further information: <iesg@ietf.org>

Intended usage: COMMON

Restrictions on usage: None

Author: Marco Tiloca <marco.tiloca@ri.se>

Change controller: IESG

14.2. CoAP Content-Formats Registry

IANA is asked to add the following entry to the "CoAP Content-Formats" registry within the "CoRE Parameters" registry group.

Media Type: application/ace-trl+cbor

Encoding: -

ID: TBD

Reference: [RFC-XXXX]

14.3. ACE Token Revocation List Parameters Registry

IANA is asked to establish the "ACE Token Revocation List Parameters" IANA registry within the "Authentication and Authorization for Constrained Environments (ACE)" registry group.

The registry uses the "Expert Review" registration procedure [RFC8126]. Expert Review guidelines are provided in Section 14.5. It should be noted that, in addition to the Expert Review, some portions of the registry require a specification, potentially a Standards Track RFC, to be supplied as well.

The columns of this registry are:

- Name: This field contains a descriptive name that enables easier reference to the item. The name MUST be unique. It is not used in the encoding.

- CBOR Value: This field contains the value used as CBOR abbreviation of the item. These values MUST be unique. The value can be an unsigned integer or a negative integer. Different ranges of values use different registration policies [RFC8126]. Integer values from -256 to 255 are designated as "Standards Action With Expert Review". Integer values from -65536 to -257 and from 256 to 65535 are designated as "Specification Required". Integer values greater than 65535 are designated as "Expert Review". Integer values less than -65536 are marked as "Private Use".

- CBOR Type: This field contains the CBOR type of the item, or a pointer to the registry that defines its type, when that depends on another item.

- Reference: This field contains a pointer to the public specification for the item.

This registry has been initially populated by the values in Section 11. The "Reference" column for all of these entries refers to this document.

14.4. ACE Token Revocation List Errors

IANA is asked to establish the "ACE Token Revocation List Errors" IANA registry within the "Authentication and Authorization for Constrained Environments (ACE)" registry group.

The registry uses the "Expert Review" registration procedure [RFC8126]. Expert Review guidelines are provided in Section 14.5. It should be noted that, in addition to the Expert Review, some portions of the registry require a specification, potentially a Standards Track RFC, to be supplied as well.

The columns of this registry are:

*Value: The field contains the value to be used to identify the error. These values MUST be unique. The value can be an unsigned integer or a negative integer. Different ranges of values use different registration policies [RFC8126]. Integer values from -256 to 255 are designated as "Standards Action With Expert Review". Integer values from -65536 to -257 and from 256 to 65535 are designated as "Specification Required". Integer values greater than 65535 are designated as "Expert Review". Integer values less than -65536 are marked as "Private Use".

*Description: This field contains a brief description of the error.

*Reference: This field contains a pointer to the public specification defining the error, if one exists.

This registry has been initially populated by the values in Section 12. The "Reference" column for all of these entries refers to this document.

14.5. Expert Review Instructions

The IANA registries established in this document are defined as "Expert Review". This section gives some general guidelines for what the experts should be looking for, but they are being designated as experts for a reason so they should be given substantial latitude.

Expert reviewers should take into consideration the following points:

*Point squatting should be discouraged. Reviewers are encouraged to get sufficient information for registration requests to ensure that the usage is not going to duplicate one that is already registered and that the point is likely to be used in deployments. The zones tagged as private use are intended for testing purposes and closed environments. Code points in other ranges should not be assigned for testing.

- *Specifications are required for the "Standards Action With Expert Review" range of point assignment. Specifications should exist for "Specification Required" ranges, but early assignment before a specification is available is considered to be permissible. Specifications are needed for the "Expert Review" range if they are expected to be used outside of closed environments in an interoperable way. When specifications are not provided, the description provided needs to have sufficient information to identify what the point is being used for.

- *Experts should take into account the expected usage of fields when approving point assignment. The fact that there is a range for Standards Track documents does not mean that a Standards Track document cannot have points assigned outside of that range. The length of the encoded value should be weighed against how many code points of that length are left, the size of device it will be used on, and the number of code points left that encode to that size.

- *Even for "Expert Review", specifications are recommended. When specifications are not provided for a request where "Expert Review" is the assignment policy, the description provided needs to have sufficient information to verify the code points above.

15. References

15.1. Normative References

[Named.Information.Hash.Algorithm] IANA, "Named Information Hash Algorithm", <https://www.iana.org/assignments/named-information/named-information.xhtml>.

[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>.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC

6838, DOI 10.17487/RFC6838, January 2013, <https://www.rfc-editor.org/info/rfc6838>.

[RFC6920] Farrell, S., Kutscher, D., Dannewitz, C., Ohlman, B., Keranen, A., and P. Hallam-Baker, "Naming Things with Hashes", RFC 6920, DOI 10.17487/RFC6920, April 2013, <https://www.rfc-editor.org/info/rfc6920>.

[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>.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <https://www.rfc-editor.org/info/rfc7519>.

[RFC7641] Hartke, K., "Observing Resources in the Constrained Application Protocol (CoAP)", RFC 7641, DOI 10.17487/RFC7641, September 2015, <https://www.rfc-editor.org/info/rfc7641>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[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>.

[RFC8392] Jones, M., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "CBOR Web Token (CWT)", RFC 8392, DOI 10.17487/RFC8392, May 2018, <https://www.rfc-editor.org/info/rfc8392>.

[RFC8610] Birkholz, H., Vigano, C., and C. Bormann, "Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures", RFC 8610, DOI 10.17487/RFC8610, June 2019, <https://www.rfc-editor.org/info/rfc8610>.

[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 Using the OAuth 2.0 Framework (ACE-OAuth)", RFC 9200, DOI 10.17487/RFC9200, August 2022, <https://www.rfc-editor.org/info/rfc9200>.

15.2. Informative References

[I-D.bormann-t2trg-stp] Bormann, C. and K. Hartke, "The Series Transfer Pattern (STP)", Work in Progress, Internet-Draft, draft-bormann-t2trg-stp-03, 7 April 2020, <https://datatracker.ietf.org/doc/html/draft-bormann-t2trg-stp-03>.

[I-D.ietf-core-conditional-attributes] Koster, M., Soloway, A., and B. Silverajan, "Conditional Attributes for Constrained RESTful Environments", Work in Progress, Internet-Draft, draft-ietf-core-conditional-attributes-06, 14 January 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-core-conditional-attributes-06>.

[RFC7009] Lodderstedt, T., Ed., Dronia, S., and M. Scurtescu, "OAuth 2.0 Token Revocation", RFC 7009, DOI 10.17487/RFC7009, August 2013, <https://www.rfc-editor.org/info/rfc7009>.

Appendix A. On using the Series Transfer Pattern

Performing a diff query of the TRL as specified in Section 7 is in fact a usage example of the Series Transfer Pattern defined in [I-D.bormann-t2trg-stp].

That is, a diff query enables the transfer of a series of TRL updates, with the AS specifying U <= MAX_N diff entries as the U most recent updates to the portion of the TRL pertaining to a requester, i.e., a registered device or an administrator.

When responding to a diff query request from a requester (see Section 7), 'diff_set' is a subset of the update collection associated with the requester, where each 'diff_entry' record is a series item from that update collection. Note that 'diff_set' specifies the whole current update collection when the value of U is equal to SIZE, i.e., the current number of series items in the update collection.

The value N of the 'diff' query parameter in the GET request allows the requester and the AS to trade the amount of provided information with the latency of the information transfer.

Since the update collection associated with each requester includes up to MAX_N series item, the AS deletes the oldest series item when a new one is generated and added to the end of the update collection, due to a new TRL update pertaining to that requester (see Section 5.1). This addresses the question "When can the server decide to no longer retain older items?" raised in Section 3.2 of [I-D.bormann-t2trg-stp].

Furthermore, performing a diff query of the TRL together with the "Cursor" extension as specified in Section 8 in fact relies on the "Cursor" pattern of the Series Transfer Pattern (see Section 3.3 of [I-D.bormann-t2trg-stp]).

Appendix B. Parameters of the TRL Endpoint

Figure 10 provides an aggregated overview of the parameters used by the TRL endpoint, when the AS supports diff queries (see Section 5) and the "Cursor" extension (see Section 5.1.1).

Except for MAX_N defined in Section 5.1, all the other parameters are defined in Section 5.1.1 and are used only if the AS supports the "Cursor" extension.

For each parameter, the columns of the table specify the following information. Both a registered device and an administrator are referred to as "requester".

- Name: parameter name. A name with letters in uppercase denotes a parameter whose value does not change after its initialization.

- Single instance: "Y", if there is a single parameter instance associated with the TRL resource; or "N", if there is one parameter instance per update collection (i.e., per requester).

- Description: short parameter description.

- Values: the unsigned integer values that the parameter can assume, where LB and UB denote the inclusive lower bound and upper bound, respectively, and "**" is the exponentiation operator.

Name	Single instance	Description	Value
MAX_N	Y	Max number of TRL updates stored per requester	LB = 1 If supporting "Cursor", then UB = (MAX_INDEX+1)

MAX_DIFF_BATCH	N	Max number of diff entries included in a diff query response when using "Cursor"	LB = 1
			UB = MAX_N

MAX_INDEX	Y	Max value of each instance of the 'index' parameter	LB = (MAX_N-1)
			UB = ((2**64)-1)

index	N	Value associated with a series item of an updated collection	LB = 0
			UB = MAX_INDEX

last_index	N	The 'index' value of the most recently added series item in an update collection	LB = 0
			UB = MAX_INDEX

Figure 10: Parameters of the TRL Endpoint

Appendix C. Interaction Examples

This section provides examples of interactions between an RS as a registered device and an AS. The AS supports both full queries and diff queries of the TRL, as defined in Section 6 and Section 7, respectively.

The details of the registration process are omitted, but it is assumed that the RS sends an unspecified payload to the AS, which replies with a 2.01 (Created) response.

The payload of the registration response is a CBOR map, which includes the following entries:

- a 'trl_path' parameter, specifying the path of the TRL resource;

- a 'trl_hash' parameter, specifying the hash function used to computed token hashes as defined in Section 3;
- an 'max_n' parameter, specifying the value of MAX_N, i.e., the maximum number of TRL updates pertaining to each registered device that the AS retains for that device (see Section 7);
- possible further parameters related to the registration process.

Furthermore, 'h(x)' refers to the hash function used to compute the token hashes, as defined in Section 3 of this specification and according to [RFC6920]. Assuming the usage of CWTs transported in CBOR, 'bstr.h(t1)' and 'bstr.h(t2)' denote the byte-string representations of the token hashes for the Access Tokens t1 and t2, respectively.

C.1. Full Query with Observe

Figure 11 shows an interaction example considering a CoAP observation and a full query of the TRL.

In this example, the AS does not support the "Cursor" extension. Hence the 'cursor' parameter is not included in the payload of the responses to a full query request.

```
RS                                                  AS
|                                                    |
|  Registration: POST                                |
+--------------------------------------------------->|
|                                                    |
|<---------------------------------------------------+
|                    2.01 CREATED                    |
|                      Payload: {                    |
|                        ...                         |
|                        "trl_path" : "revoke/trl",  |
|                        "trl_hash" : "sha-256",     |
|                           "max_n" : 10             |
|                      }                             |
|                                                    |
|  GET Observe: 0                                    |
|    coap://as.example.com/revoke/trl/               |
+--------------------------------------------------->|
|                                                    |
|<---------------------------------------------------+
|      2.05 CONTENT Observe: 42                      |
|        Content-Format: "application/ace-trl+cbor"  |
|        Payload: {                                  |
|          "full_set" : []                           |
|        }                                           |
|                         .                          |
|                         .                          |
|                         .                          |
|                                                    |
|          (Access Tokens t1 and t2 issued           |
|          and successfully submitted to RS)         |
|                         .                          |
|                         .                          |
|                         .                          |
|                                                    |
|                                                    |
|             (Access Token t1 is revoked)           |
|                                                    |
|<---------------------------------------------------+
|      2.05 CONTENT Observe: 53                      |
|        Content-Format: "application/ace-trl+cbor"  |
|        Payload: {                                  |
|          "full_set" : [bstr.h(t1)]                 |
|        }                                           |
|                         .                          |
|                         .                          |
|                         .                          |
|                                                    |
|             (Access Token t2 is revoked)           |
|                                                    |
|<---------------------------------------------------+
|      2.05 CONTENT Observe: 64                      |
|        Content-Format: "application/ace-trl+cbor"  |
|        Payload: {                                  |
|          "full_set" : [bstr.h(t1), bstr.h(t2)]     |
|        }                                           |
|                                                    |
|                         .                          |
|                         .                          |
|                         .                          |
|                                                    |
|             (Access Token t1 expires)              |
|                                                    |
|<---------------------------------------------------+
|      2.05 CONTENT Observe: 75                      |
|        Content-Format: "application/ace-trl+cbor"  |
|        Payload: {                                  |
|          "full_set" : [bstr.h(t2)]                 |
|        }                                           |
|                         .                          |
|                         .                          |
|                         .                          |
|                                                    |
|             (Access Token t2 expires)              |
|                                                    |
|<---------------------------------------------------+
|      2.05 CONTENT Observe: 86                      |
|        Content-Format: "application/ace-trl+cbor"  |
|        Payload: {                                  |
|          "full_set" : []                           |
|        }                                           |
|                                                    |
```

Figure 11: Interaction for Full Query with Observe

C.2. Diff Query with Observe

Figure 12 shows an interaction example considering a CoAP observation and a diff query of the TRL.

The RS indicates N=3 as value of the 'diff' query parameter, i.e., as the maximum number of diff entries to be specified in a response from the AS.

In this example, the AS does not support the "Cursor" extension. Hence the 'cursor' parameter and the 'more' parameter are not included in the payload of the responses to a diff query request.

```
RS                                                  AS
|                                                    |
|  Registration: POST                                |
+--------------------------------------------------->|
|                                                    |
|<---------------------------------------------------+
|                   2.01 CREATED                     |
|                     Payload: {                     |
|                       ...                          |
|                       "trl_path" : "revoke/trl",   |
|                       "trl_hash" : "sha-256",      |
|                          "max_n" : 10              |
|                     }                              |
|                                                    |
|  GET Observe: 0                                    |
|    coap://as.example.com/revoke/trl?diff=3         |
+--------------------------------------------------->|
|                                                    |
|<---------------------------------------------------+
|      2.05 CONTENT Observe: 42                      |
|        Content-Format: "application/ace-trl+cbor"  |
|        Payload: {                                  |
|          "diff_set" : []                           |
|        }                                           |
|                         .                          |
|                         .                          |
|                         .                          |
|                                                    |
|          (Access Tokens t1 and t2 issued           |
|          and successfully submitted to RS)         |
|                         .                          |
|                         .                          |
|                         .                          |
|                                                    |
|            (Access Token t1 is revoked)            |
|                                                    |
|<---------------------------------------------------+
|      2.05 CONTENT Observe: 53                      |
|        Content-Format: "application/ace-trl+cbor"  |
|        Payload: {                                  |
|          "diff_set" : [                            |
|                         [ [], [bstr.h(t1)] ]       |
|                       ]                            |
|        }                                           |
|                         .                          |
|                         .                          |
|                         .                          |
|                                                    |
|            (Access Token t2 is revoked)            |
|                                                    |
|<---------------------------------------------------+
|      2.05 CONTENT Observe: 64                      |
|        Content-Format: "application/ace-trl+cbor"  |
|        Payload: {                                  |
|          "diff_set" : [                            |
|                         [ [], [bstr.h(t2)] ],      |
|                         [ [], [bstr.h(t1)] ]       |
|                       ]                            |
|        }                                           |
|                         .                          |
|                         .                          |
|                         .                          |
|                                                    |
|              (Access Token t1 expires)             |
|                                                    |
|<---------------------------------------------------+
|      2.05 CONTENT Observe: 75                      |
|        Content-Format: "application/ace-trl+cbor"  |
|        Payload: {                                  |
|          "diff_set" : [                            |
|                         [ [bstr.h(t1)], [] ],      |
|                         [ [], [bstr.h(t2)] ],      |
|                         [ [], [bstr.h(t1)] ]       |
|                       ]                            |
|        }                                           |
|                         .                          |
|                         .                          |
|                         .                          |
|                                                    |
|              (Access Token t2 expires)             |
|                                                    |
|<---------------------------------------------------+
|      2.05 CONTENT Observe: 86                      |
|        Content-Format: "application/ace-trl+cbor"  |
|        Payload: {                                  |
|          "diff_set" : [                            |
|                         [ [bstr.h(t2)], [] ],      |
|                         [ [bstr.h(t1)], [] ],      |
|                         [ [], [bstr.h(t2)] ]       |
|                       ]                            |
|        }                                           |
|                                                    |
```

Figure 12: Interaction for Diff Query with Observe

C.3. Full Query with Observe plus Diff Query

Figure 13 shows an interaction example considering a CoAP observation and a full query of the TRL.

The example also considers one of the notifications from the AS to get lost in transmission, and thus not reaching the RS.

When this happens, and after a waiting time defined by the application has elapsed, the RS sends a GET request with no Observe Option to the AS, to perform a diff query of the TRL. The RS indicates N=8 as value of the 'diff' query parameter, i.e., as the maximum number of diff entries to be specified in a response from the AS.

In this example, the AS does not support the "Cursor" extension. Hence, the 'cursor' parameter is not included in the payload of the responses to a full query request. Also, the 'cursor' parameter and the 'more' parameter are not included in the payload of the responses to a diff query request.

```
RS                                                  AS
|                                                    |
|  Registration: POST                                |
+--------------------------------------------------->|
|                                                    |
|<---------------------------------------------------+
|                   2.01 CREATED                     |
|                     Payload: {                     |
|                       ...                          |
|                       "trl_path" : "revoke/trl",   |
|                       "trl_hash" : "sha-256",      |
|                          "max_n" : 10              |
|                     }                              |
|                                                    |
|  GET Observe: 0                                    |
|    coap://as.example.com/revoke/trl/               |
+--------------------------------------------------->|
|                                                    |
|<---------------------------------------------------+
|      2.05 CONTENT Observe: 42                      |
|        Content-Format: "application/ace-trl+cbor"  |
|        Payload: {                                  |
|          "full_set" : []                           |
|        }                                           |
|                         .                          |
|                         .                          |
|                         .                          |
|                                                    |
|          (Access Tokens t1 and t2 issued           |
|          and successfully submitted to RS)         |
|                         .                          |
|                         .                          |
|                         .                          |
|                                                    |
|            (Access Token t1 is revoked)            |
|                                                    |
|<---------------------------------------------------+
|      2.05 CONTENT Observe: 53                      |
|        Content-Format: "application/ace-trl+cbor"  |
|        Payload: {                                  |
|          "full_set" : [bstr.h(t1)]                 |
|        }                                           |
|                         .                          |
|                         .                          |
|                         .                          |
|                                                    |
|            (Access Token t2 is revoked)            |
|                                                    |
|<---------------------------------------------------+
|      2.05 CONTENT Observe: 64                      |
|        Content-Format: "application/ace-trl+cbor"  |
|        Payload: {                                  |
|          "full_set" : [bstr.h(t1), bstr.h(t2)]     |
|        }                                           |
|                         .                          |
|                         .                          |
|                         .                          |
|                                                    |
|             (Access Token t1 expires)              |
|                                                    |
|<---------------------------------------------------+
|      2.05 CONTENT Observe: 75                      |
|        Content-Format: "application/ace-trl+cbor"  |
|        Payload: {                                  |
|          "full_set" : [bstr.h(t2)]                 |
|        }                                           |
|                         .                          |
|                         .                          |
|                         .                          |
|                                                    |
|             (Access Token t2 expires)              |
|                                                    |
|  X<------------------------------------------------+
|      2.05 CONTENT Observe: 86                      |
|        Content-Format: "application/ace-trl+cbor"  |
|        Payload: {                                  |
|          "full_set" : []                           |
|        }                                           |
|                         .                          |
|                         .                          |
|                         .                          |
|                                                    |
|           (Enough time has passed since            |
|         the latest received notification)          |
|                                                    |
|  GET                                               |
|    coap://as.example.com/revoke/trl?diff=8         |
+--------------------------------------------------->|
|                                                    |
|<---------------------------------------------------+
|      2.05 CONTENT                                  |
|        Content-Format: "application/ace-trl+cbor"  |
|        Payload: {                                  |
|          "diff_set" : [                            |
|                         [ [bstr.h(t2)], [] ],      |
|                         [ [bstr.h(t1)], [] ],      |
|                         [ [], [bstr.h(t2)] ],      |
|                         [ [], [bstr.h(t1)] ]       |
|                   ]                       |
|      }                                    |
|                                           |
```

Figure 13: Interaction for Full Query with Observe plus Diff Query

C.4. Diff Query with Observe and "Cursor"

In this example, the AS supports the "Cursor" extension. Hence, the CBOR map conveyed as payload of the registration response additionally includes a "max_diff_batch" parameter. This specifies the value of MAX_DIFF_BATCH, i.e., the maximum number of diff entries that can be included in a response to a diff query from this RS.

Figure 14 shows an interaction example considering a CoAP observation and a diff query of the TRL.

The RS specifies the query parameter 'diff' with value 3, i.e., the maximum number of diff entries to be specified in a response from the AS.

After the RS has not received a notification from the AS for a waiting time defined by the application, the RS sends a GET request with no Observe Option to the AS, to perform a diff query of the TRL.

This is followed up by a further diff query request that specifies the query parameter 'cursor'. Note that the payload of the corresponding response differs from the payload of the response to the previous diff query request.

```
RS                                                      AS
|                                                        |
|  Registration: POST                                    |
+------------------------------------------------------->|
|                                                        |
|<-------------------------------------------------------+
|                   2.01 CREATED                         |
|                     Payload: {                         |
|                            ...                         |
|                            "trl_path" : "revoke/trl",  |
|                            "trl_hash" : "sha-256",     |
|                               "max_n" : 10,            |
|                       "max_diff_batch": 5              |
|                     }                                  |
|                                                        |
|  GET Observe: 0                                        |
|    coap://as.example.com/revoke/trl?diff=3             |
+------------------------------------------------------->|
|                                                        |
|<-------------------------------------------------------+
|          2.05 CONTENT Observe: 42                      |
|            Content-Format: "application/ace-trl+cbor"  |
|            Payload: {                                  |
|              "diff_set" : [],                          |
|                "cursor" : null,                        |
|                  "more" : false                        |
|            }                                           |
|                           .                            |
|                           .                            |
|                           .                            |
|                                                        |
|            (Access Tokens t1 and t2 issued             |
|            and successfully submitted to RS)           |
|                           .                            |
|                           .                            |
|                           .                            |
|                                                        |
|              (Access Token t1 is revoked)              |
|                                                        |
|<-------------------------------------------------------+
|          2.05 CONTENT Observe: 53                      |
|            Content-Format: "application/ace-trl+cbor"  |
|            Payload: {                                  |
|              "diff_set" : [                            |
|                             [ [], [bstr.h(t1)] ]       |
|                           ],                           |
|                "cursor" : 0,                           |
|                  "more" : false                        |
|            }                                           |
|                           .                            |
|                           .                            |
|                           .                            |
|                                                        |
|              (Access Token t2 is revoked)              |
|                                                        |
|<-------------------------------------------------------+
|          2.05 CONTENT Observe: 64                      |
|            Content-Format: "application/ace-trl+cbor"  |
|            Payload: {                                  |
|              "diff_set" : [                            |
|                             [ [], [bstr.h(t2)] ],      |
|                             [ [], [bstr.h(t1)] ]       |
|                           ],                           |
|                "cursor" : 1,                           |
|                  "more" : false                        |
|            }                                           |
|                           .                            |
|                           .                            |
|                           .                            |
|                                                        |
|              (Access Token t1 expires)                 |
|                                                        |
|<-------------------------------------------------------+
|          2.05 CONTENT Observe: 75                      |
|            Content-Format: "application/ace-trl+cbor"  |
|            Payload: {                                  |
|              "diff_set" : [                            |
|                             [ [bstr.h(t1)], [] ],      |
|                             [ [], [bstr.h(t2)] ],      |
|                             [ [], [bstr.h(t1)] ]       |
|                           ],                           |
|                "cursor" : 2,                           |
|                  "more" : false                        |
|            }                                           |
|                           .                            |
|                           .                            |
|                           .                            |
|                                                        |
|              (Access Token t2 expires)                 |
|                                                        |
|<-------------------------------------------------------+
|          2.05 CONTENT Observe: 86                      |
|            Content-Format: "application/ace-trl+cbor"  |
|            Payload: {                                  |
|              "diff_set" : [                            |
|                             [ [bstr.h(t2)], [] ],      |
|                             [ [bstr.h(t1)], [] ],      |
|                             [ [], [bstr.h(t2)] ]       |
|                           ],                            |
|                "cursor" : 3,                            |
|                  "more" : false                         |
|            }                                            |
|                           .                             |
|                           .                             |
|                           .                             |
|                                                         |
|            (Enough time has passed since                |
|             the latest received notification)           |
|                                                         |
|  GET                                                    |
|    coap://as.example.com/revoke/trl?diff=3              |
+------------------------------------------------------->|
|                                                         |
|<-------------------------------------------------------+
|          2.05 CONTENT                                   |
|            Content-Format: "application/ace-trl+cbor"   |
|            Payload: {                                   |
|              "diff_set" : [                             |
|                             [ [bstr.h(t2)], [] ],       |
|                             [ [bstr.h(t1)], [] ],       |
|                             [ [], [bstr.h(t2)] ]        |
|                           ],                            |
|                "cursor" : 3,                            |
|                  "more" : false                         |
|            }                                            |
|                                                         |
|  GET                                                    |
|    coap://as.example.com/revoke/trl?diff=3&cursor=3     |
+------------------------------------------------------->|
|                                                         |
|<-------------------------------------------------------+
|          2.05 CONTENT                                   |
|            Content-Format: "application/ace-trl+cbor"   |
|            Payload: {                                   |
|              "diff_set" : [],                           |
|                "cursor" : 3,                            |
|                  "more" : false                         |
|            }                                            |
|                                                         |
```

Figure 14: Interaction for Diff Query with Observe and "Cursor"

C.5. Full Query with Observe plus Diff Query with "Cursor"

In this example, the AS supports the "Cursor" extension. Hence, the CBOR map conveyed as payload of the registration response additionally includes a "max_diff_batch" parameter. This specifies the value of MAX_DIFF_BATCH, i.e., the maximum number of diff entries that can be included in a response to a diff query from this RS.

Figure 15 shows an interaction example considering a CoAP observation and a full query of the TRL.

The example also considers some of the notifications from the AS to get lost in transmission, and thus not reaching the RS.

When this happens, and after a waiting time defined by the application has elapsed, the RS sends a GET request with no Observe Option to the AS, to perform a diff query of the TRL. In particular, the RS specifies:

- The query parameter 'diff' with value 8, i.e., the maximum number of diff entries to be specified in a response from the AS.

- The query parameter 'cursor' with value 2, thus requesting from the update collection the series items following the one with 'index' value equal to 2 (i.e., following the last series item that the RS successfully received in an earlier notification response).

The response from the AS conveys a first batch of MAX_DIFF_BATCH=5 series items from the update collection corresponding to the RS. The AS indicates that further series items are actually available in the update collection, by setting the 'more' parameter of the response to "true". Also, the 'cursor' parameter of the response is set to 7, i.e., to the 'index' value of the most recent series item included in the response.

After that, the RS follows up with a further diff query request specifying the query parameter 'cursor' with value 7, in order to retrieve the next and last batch of series items from the update collection.

```
RS                                                             AS
|                                                               |
|  Registration: POST                                           |
+-------------------------------------------------------------->|
|                                                               |
|<--------------------------------------------------------------+
|                          2.01 CREATED                         |
|                            Payload: {                         |
|                                   ...                         |
|                                   "trl_path" : "revoke/trl",  |
|                                   "trl_hash" : "sha-256",     |
|                                      "max_n" : 10,            |
|                              "max_diff_batch": 5              |
|                            }                                  |
|                                                               |
|  GET Observe: 0                                               |
|    coap://as.example.com/revoke/trl/                          |
+-------------------------------------------------------------->|
|                                                               |
|<--------------------------------------------------------------+
|                 2.05 CONTENT Observe: 42                      |
|                   Content-Format: "application/ace-trl+cbor"  |
|                   Payload: {                                  |
|                     "full_set" : [],                          |
|                       "cursor" : null                         |
|                   }                                           |
|                               .                               |
|                               .                               |
|                               .                               |
|                                                               |
|               (Access Tokens t1, t2, t3 issued                |
|                and successfully submitted to RS)              |
|                               .                               |
|                               .                               |
|                               .                               |
|                                                               |
|               (Access Tokens t4, t5, t6 issued                |
|               and successfully submitted to RS)               |
|                               .                               |
|                               .                               |
|                               .                               |
|                                                               |
|                  (Access Token t1 is revoked)                 |
|                                                               |
|<--------------------------------------------------------------+
|                 2.05 CONTENT Observe: 53                      |
|                   Content-Format: "application/ace-trl+cbor"  |
|                   Payload: {                                  |
|                     "full_set" : [bstr.h(t1)],                |
|                       "cursor" : 0                            |
|                   }                                           |
|                               .                               |
|                               .                               |
|                               .                               |
|                                                               |
|                  (Access Token t2 is revoked)                 |
|                                                               |
|<--------------------------------------------------------------+
|                 2.05 CONTENT Observe: 64                      |
|                   Content-Format: "application/ace-trl+cbor"  |
|                   Payload: {                                  |
|                     "full_set" : [bstr.h(t1), bstr.h(t2)],    |
|                       "cursor" : 1                            |
|                   }                                           |
|                               .                               |
|                               .                               |
|                               .                               |
|                                                               |
|                   (Access Token t1 expires)                   |
|                                                               |
|<--------------------------------------------------------------+
|                 2.05 CONTENT Observe: 75                      |
|                   Content-Format: "application/ace-trl+cbor"  |
|                   Payload: {                                  |
|                     "full_set" : [bstr.h(t2)],                |
|                     "cursor"   : 2                            |
|                   }                                           |
|                               .                               |
|                               .                               |
|                               .                               |
|                                                               |
|                   (Access Token t2 expires)                   |
|                                                               |
|  X<-----------------------------------------------------------+
|                 2.05 CONTENT Observe: 86                      |
|                   Content-Format: "application/ace-trl+cbor"  |
|                   Payload: {                                  |
|                     "full_set" : [],                          |
|                       "cursor" : 3                            |
|                   }                                           |
|                               .                               |
|                               .                               |
|                               .                               |
|                                                               |
|                  (Access Token t3 is revoked)                 |
|                                                               |
|  X<-----------------------------------------------------------+
|                 2.05 CONTENT Observe: 88                      |
|                   Content-Format: "application/ace-trl+cbor"   |
|                   Payload: {                                   |
|                     "full_set" : [bstr.h(t3)],                 |
|                       "cursor" : 4                             |
|                   }                                            |
|                               .                               |
|                               .                               |
|                               .                               |
|                                                               |
|                  (Access Token t4 is revoked)                 |
|                                                               |
|  X<-----------------------------------------------------------+
|                 2.05 CONTENT Observe: 89                      |
|                   Content-Format: "application/ace-trl+cbor"   |
|                   Payload: {                                   |
|                     "full_set" : [bstr.h(t3), bstr.h(t4)],     |
|                       "cursor" : 5                             |
|                   }                                            |
|                               .                               |
|                               .                               |
|                               .                               |
|                                                               |
|                    (Access Token t3 expires)                  |
|                                                               |
|  X<-----------------------------------------------------------+
|                 2.05 CONTENT Observe: 90                      |
|                   Content-Format: "application/ace-trl+cbor"   |
|                   Payload: {                                   |
|                     "full_set" : [bstr.h(t4)],                 |
|                       "cursor" : 6                             |
|                   }                                            |
|                               .                               |
|                               .                               |
|                               .                               |
|                                                               |
|                    (Access Token t4 expires)                  |
|                                                               |
|  X<-----------------------------------------------------------+
|                 2.05 CONTENT Observe: 91                      |
|                   Content-Format: "application/ace-trl+cbor"   |
|                   Payload: {                                   |
|                     "full_set" : [],                           |
|                       "cursor" : 7                             |
|                   }                                            |
|                               .                               |
|                               .                               |
|                               .                               |
|                                                               |
|              (Access Tokens t5 and t6 are revoked)            |
|                                                               |
|  X<-----------------------------------------------------------+
|                 2.05 CONTENT Observe: 92                      |
|                   Content-Format: "application/ace-trl+cbor"  |
|                   Payload: {                                  |
|                     "full_set" : [bstr.h(t5), bstr.h(t6)],    |
|                     "cursor" : 8                              |
|                   }                                           |
|                               .                               |
|                               .                               |
|                               .                               |
|                                                               |
|                    (Access Token t5 expires)                  |
|                                                               |
|  X<-----------------------------------------------------------+
|                 2.05 CONTENT Observe: 93                      |
|                   Content-Format: "application/ace-trl+cbor"  |
|                   Payload: {                                  |
|                     "full_set" : [bstr.h(t6)],                |
|                     "cursor" : 9                              |
|                   }                                           |
|                               .                               |
|                               .                               |
|                               .                               |
|                                                               |
|                    (Access Token t6 expires)                  |
|                                                               |
|  X<-----------------------------------------------------------+
|                 2.05 CONTENT Observe: 94                      |
|                   Content-Format: "application/ace-trl+cbor"  |
|                   Payload: {                                  |
|                     "full_set" : [],                          |
|                       "cursor" : 10                           |
|                   }                                           |
|                               .                               |
|                               .                               |
|                               .                               |
|                                                               |
|                (Enough time has passed since                  |
|                 the latest received notification)             |
|                                                               |
|  GET                                                          |
|    coap://as.example.com/revoke/trl?diff=8&cursor=2           |
+-------------------------------------------------------------->|
|                                                               |
|<--------------------------------------------------------------+
|                 2.05 CONTENT                                  |
|                   Content-Format: "application/ace-trl+cbor"  |
|                   Payload: {                                  |
|                     "diff_set" : [                             |
|                                    [ [bstr.h(t4)], [] ],       |
|                                    [ [bstr.h(t3)], [] ],       |
|                                    [ [], [bstr.h(t4)] ],       |
|                                    [ [], [bstr.h(t3)] ],       |
|                                    [ [bstr.h(t2)], [] ]        |
|                                  ],                            |
|                       "cursor" : 7,                            |
|                         "more" : true                          |
|                   }                                            |
|                                                               |
|  GET                                                          |
|    coap://as.example.com/revoke/trl?diff=8&cursor=7           |
+-------------------------------------------------------------->|
|                                                               |
|<--------------------------------------------------------------+
|        2.05 CONTENT                                           |
|          Content-Format: "application/ace-trl+cbor"           |
|          Payload: {                                           |
|            "diff_set" : [                                     |
|                           [ [bstr.h(t6)], [] ],               |
|                           [ [bstr.h(t5)], [] ],               |
|                           [ [], [bstr.h(t5), bstr.h(t6)] ]    |
|                         ],                                    |
|              "cursor" : 10,                                   |
|                "more" : false                                 |
|          }                                                    |
|                                                               |
```

Figure 15: Interaction for Full Query with Observe plus Diff Query with "Cursor"

Appendix D. Document Updates

RFC EDITOR: Please remove this section.

D.1. Version -03 to -04

- Improved presentation of pre- and post-registration operations.

- Removed moot processing cases with the "Cursor" extension.

- Positive integers as CBOR abbreviations for all parameters.

- Renamed N_MAX as MAX_N.

- Access Tokens are not necessarily uploaded through /authz-info.

- The use of the "c.pmax" conditional attribute is just an example.

- Revised handling of token hashes at the RS.

- Extended and improved security considerations.

- Fixed details in IANA considerations.

- New appendix overviewing parameters of the TRL endpoint.

- Examples of message exchange moved to an appendix.

- Added examples of message exchange with the "Cursor" extension.

- Clarifications and editorial improvements.

D.2. Version -02 to -03

- Definition of MAX_INDEX for the "Cursor" extension.

- Handling wrap-around of 'index' when using the "Cursor" extension.

- Error handling for the case where 'cursor' > MAX_INDEX.

- Improved error handling in case 'index' is out-of-bound.

- Clarified parameter semantics, message content and examples.

- Editorial improvements.

D.3. Version -01 to -02

*Earlier mentioning of error cases.

*Clearer distinction between maintaining the history of TRL updates and preparing the response to a diff query.

*Defined the use of "cursor" in the document body, as an extension of diff queries.

*Both success and error responses have a CBOR map as payload.

*Corner cases of message processing explained more explcitly.

*Clarifications and editorial improvements.

D.4. Version -00 to -01

*Added actions to perform upon receiving responses from the TRL endpoint.

*Fixed off-by-one error when using the "Cursor" pattern.

*Improved error handling, with registered error codes.

*Section restructuring (full- and diff-query as self-standing sections).

*Renamed identifiers and CBOR parameters.

*Clarifications and editorial improvements.

Acknowledgments

The authors sincerely thank Christian Amsüss, Carsten Bormann, Rikard Höglund, Benjamin Kaduk, David Navarro, Marco Rasori, Michael Richardson, Jim Schaad, Göran Selander and Travis Spencer for their comments and feedback.

The work on this document has been partly supported by VINNOVA and the Celtic-Next project CRITISEC; and by the H2020 project SIFIS-Home (Grant agreement 952652).

Authors' Addresses

Marco Tiloca
RISE AB
Isafjordsgatan 22
SE-16440 Kista
Sweden

Email: marco.tiloca@ri.se

Ludwig Seitz
Combitech
Djaeknegatan 31
SE-21135 Malmoe
Sweden

Email: ludwig.seitz@combitech.com

Francesca Palombini
Ericsson AB
Torshamnsgatan 23
SE-16440 Kista
Sweden

Email: francesca.palombini@ericsson.com

Sebastian Echeverria
CMU SEI
4500 Fifth Avenue
Pittsburgh, PA, 15213-2612
United States of America

Email: secheverria@sei.cmu.edu

Grace Lewis
CMU SEI
4500 Fifth Avenue
Pittsburgh, PA, 15213-2612
United States of America

Email: glewis@sei.cmu.edu