ACE Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 6, 2020

F. Palombini
Ericsson AB
M. Tiloca
RISE AB

July 05, 2019

Key Provisioning for Group Communication using ACE
draft-ietf-ace-key-groupcomm-02

Abstract

This document defines message formats and procedures for requesting and distributing group keying material using the ACE framework, to protect communications between group members.

Status of This Memo

Copyright Notice

Table of Contents

```
   1.  Introduction . . . . . . . . . . . . . . . . . . . . . . . .   3
     1.1.  Terminology . . . . . . . . . . . . . . . . . . . . . . .   3
   2.  Overview  . . . . . . . . . . . . . . . . . . . . . . . . . .   4
   3.  Authorization to Join a Group . . . . . . . . . . . . . . . .   6
     3.1.  Authorization Request . . . . . . . . . . . . . . . . . .   7
     3.2.  Authorization Response  . . . . . . . . . . . . . . . . .   7
     3.3.  Token Post  . . . . . . . . . . . . . . . . . . . . . . .   8
   4.  Key Distribution  . . . . . . . . . . . . . . . . . . . . . .  11
     4.1.  Key Distribution Request  . . . . . . . . . . . . . . . .  12
     4.2.  Key Distribution Response . . . . . . . . . . . . . . . .  13
   5.  Removal of a Node from the Group  . . . . . . . . . . . . . .  16
     5.1.  Expired Authorization . . . . . . . . . . . . . . . . . .  16
     5.2.  Request to Leave the Group  . . . . . . . . . . . . . . .  16
   6.  Retrieval of New or Updated Keying Material . . . . . . . . .  17
     6.1.  Key Re-Distribution Request . . . . . . . . . . . . . . .  18
     6.2.  Key Re-Distribution Response  . . . . . . . . . . . . . .  19
   7.  Retrieval of Public Keys for Group Members  . . . . . . . . .  19
     7.1.  Public Key Request  . . . . . . . . . . . . . . . . . . .  20
     7.2.  Public Key Response . . . . . . . . . . . . . . . . . . .  20
   8.  ACE Groupcomm Parameters  . . . . . . . . . . . . . . . . . .  21
   9.  ACE Groupcomm Request Type  . . . . . . . . . . . . . . . . .  22
   10. Security Considerations . . . . . . . . . . . . . . . . . . .  23
     10.1.  Update of Keying Material  . . . . . . . . . . . . . . .  24
     10.2.  Block-Wise Considerations  . . . . . . . . . . . . . . .  24
   11. IANA Considerations . . . . . . . . . . . . . . . . . . . . .  25
     11.1.  ACE Authorization Server Request Creation Hints Registry  25
     11.2.  ACE Public Key Encoding Registry . . . . . . . . . . . .  25
     11.3.  ACE Groupcomm Parameters Registry  . . . . . . . . . . .  26
     11.4.  Ace Groupcomm Request Type Registry  . . . . . . . . . .  26
     11.5.  ACE Groupcomm Key Registry . . . . . . . . . . . . . . .  27
     11.6.  ACE Groupcomm Profile Registry . . . . . . . . . . . . .  28
     11.7.  ACE Groupcomm Policy Registry  . . . . . . . . . . . . .  28
     11.8.  Sequence Number Synchronization Method Registry  . . . .  29
     11.9.  Expert Review Instructions . . . . . . . . . . . . . . .  29
   12. References  . . . . . . . . . . . . . . . . . . . . . . . . .  30
     12.1.  Normative References . . . . . . . . . . . . . . . . . .  30
     12.2.  Informative References . . . . . . . . . . . . . . . . .  31
   Appendix A.  Requirements on Application Profiles . . . . . . . .  33
   Appendix B.  Document Updates . . . . . . . . . . . . . . . . . .  33
     B.1.  Version -01 to -02  . . . . . . . . . . . . . . . . . . .  34
     B.2.  Version -00 to -01  . . . . . . . . . . . . . . . . . . .  34
   Acknowledgments . . . . . . . . . . . . . . . . . . . . . . . . .  35
   Authors' Addresses  . . . . . . . . . . . . . . . . . . . . . . .  35
```

1. Introduction

This document expands the ACE framework [I-D.ietf-ace-oauth-authz] to define the format of messages used to request, distribute and renew the keying material in a group communication scenario, e.g. based on multicast [RFC7390][I-D.dijk-core-groupcomm-bis] or on publishing-subscribing [I-D.ietf-core-coap-pubsub]. The ACE framework is based on CBOR [RFC7049], so CBOR is the format used in this specification. However, using JSON [RFC8259] instead of CBOR is possible, using the conversion method specified in Sections 4.1 and 4.2 of [RFC7049].

Profiles that use group communication can build on this document to specify the selection of the message parameters defined in this document to use and their values. Known applications that can benefit from this document would be, for example, those addressing group communication based on multicast [RFC7390][I-D.dijk-core-groupcomm-bis] or publishing/subscribing [I-D.ietf-core-coap-pubsub] in ACE.

If the application requires backward and forward security, updated keying material is generated and distributed to the group members (rekeying), when membership changes. A key management scheme performs the actual distribution of the updated keying material to the group. In particular, the key management scheme rekeys the current group members when a new node joins the group, and the remaining group members when a node leaves the group. This document provides a message format for group rekeying that allows to fulfill these requirements. Rekeying mechanisms can be based on [RFC2093], [RFC2094] and [RFC2627].

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]. These words may also appear in this document in lowercase, absent their normative meanings.

Readers are expected to be familiar with the terms and concepts described in [I-D.ietf-ace-oauth-authz] and [RFC8152], such as Authorization Server (AS) and Resource Server (RS).

This document additionally uses the following terminology:

o Transport profile, to indicate a profile of ACE as per Section 5.6.4.3 of [I-D.ietf-ace-oauth-authz]. That is, a transport profile specifies the communication protocol and communication security protocol between an ACE Client and Resource

      Server, as well as proof-of-possession methods, if it supports
      proof-of-possession access tokens. Tranport profiles of ACE
      include, for instance, [I-D.ietf-ace-oscore-profile],
      [I-D.ietf-ace-dtls-authorize] and [I-D.ietf-ace-mqtt-tls-profile].

   o Application profile, to indicate a profile of ACE that defines how
      applications enforce and use supporting security services they
      require. These services include, for instance, provisioning,
      revocation and (re-)distribution of keying material. An
      application profile may define specific procedures and message
      formats.

2. Overview

```
       +------------+                  +-----------+
       |     AS     |                  |    KDC    |
       |            |        .-------->|           |
       +------------+       /          +-----------+
             ^             /
             |            /
             v           /                           +------------+
       +------------+   /      +------------+        |+------------+
       |   Client   |<-'       | Dispatcher |        ||+------------+
       |            |<-------->|    (RS)    |<------->||   Group    |
       +------------+          +------------+         +|  members   |
                                                       +------------+
```

Figure 1: Key Distribution Participants

   The following participants (see Figure 1) take part in the
   authorization and key distribution.

   o Client (C): node that wants to join the group communication. It
      can request write and/or read rights.

   o Authorization Server (AS): same as AS in the ACE Framework; it
      enforces access policies, and knows if a node is allowed to join
      the group with write and/or read rights.

   o Key Distribution Center (KDC): maintains the keying material to
      protect group communications, and provides it to Clients
      authorized to join the group. During the first part of the
      exchange (Section 3), it takes the role of the RS in the ACE
      Framework. During the second part (Section 4), which is not based
      on the ACE Framework, it distributes the keying material. In
      addition, it provides the latest keying material to group members
      when requested. If required by the application, the KDC renews

and re-distributes the keying material in the group when membership changes.

- o Dispatcher: entity through which the Clients communicate with the group and which distributes messages to the group members. Examples of dispatchers are: the Broker node in a pub-sub setting; a relayer node for group communication that delivers group messages as multiple unicast messages to all group members; an implicit entity as in a multicast communication setting, where messages are transmitted to a multicast IP address and delivered on the transport channel.

This document specifies the message flows and formats for:

- o Authorizing a new node to join the group (Section 3), and providing it with the group keying material to communicate with the other group members (Section 4).

- o Removing of a current member from the group (Section 5).

- o Retrieving keying material as a current group member (Section 6 and Section 7).

- o Renewing and re-distributing the group keying material (rekeying) upon a membership change in the group (Section 4.2 and Section 5).

Figure 2 provides a high level overview of the message flow for a node joining a group communication setting.

```
  C                              AS     KDC   Dispatcher          Group
  |                              |       |        |               Member
  |                              |       |        | \               |
  |     Authorization Request    |       |        | | Defined       |
  |----------------------------->|       |        | | in the ACE    |
  |                              |       |        | | framework     |
  |     Authorization Response   |       |        | |               |
  |<-----------------------------|       |        | |               |
  |                              |       |        | |               |
  |--------- Token Post ---------------->|        | /               |
  |                                      |        |                 |
  |---- Key Distribution Request ------->|        |                 |
  |                                      |        |                 |
  |<--- Key Distribution Response ------ | --- Group Rekeying ----->|
  |                                               |                 |
  |<================== Protected communication ===|================>|
  |                                               |                 |
```

Figure 2: Message Flow Upon New Node's Joining

The exchange of Authorization Request and Authorization Response between Client and AS MUST be secured, as specified by the transport profile of ACE used between Client and KDC.

The exchange of Key Distribution Request and Key Distribution Response between Client and KDC MUST be secured, as a result of the transport profile of ACE used between Client and KDC.

All further communications between the Client and the KDC MUST be secured, for instance with the same security mechanism used for the Key Distribution exchange.

All communications between a Client and the other group members MUST be secured using the keying material provided in Section 4.

3. Authorization to Join a Group

This section describes in detail the format of messages exchanged by the participants when a node requests access to a group. The first part of the exchange is based on ACE [I-D.ietf-ace-oauth-authz].

As defined in [I-D.ietf-ace-oauth-authz], the Client requests from the AS an authorization to join the group through the KDC (see Section 3.1). If the request is approved and authorization is granted, the AS provides the Client with a proof-of-possession access token and parameters to securely communicate with the KDC (see Section 3.2). Communications between the Client and the AS MUST be secured, according to the transport profile of ACE used. The Content-Format used in the messages is the one specified by the used transport profile of ACE (e.g. application/ace+cbor for the first two messages and application/cwt for the third message, depending on the format of the access token).

Figure 3 gives an overview of the exchange described above.

```
      Client                                            AS  KDC
         |                                               |   |
         |---- Authorization Request: POST /token ------>|   |
         |                                               |   |
         |<--- Authorization Response: 2.01 (Created) ---|   |
         |                                               |   |
         |----- POST Token: POST /authz-info --------------->|
         |                                                   |
```

Figure 3: Message Flow of Join Authorization

3.1. Authorization Request

The Authorization Request sent from the Client to the AS is as defined in Section 5.6.1 of [I-D.ietf-ace-oauth-authz] and MUST contain the following parameters:

o 'grant_type', with value "client_credentials".

Additionally, the Authorization Request MAY contain the following parameters, which, if included, MUST have the corresponding values:

o 'scope', containing the identifier of the specific group (or topic in the case of pub-sub) that the Client wishes to access, and optionally the role(s) that the Client wishes to take. This value is a CBOR array encoded as a byte string, which contains:

  * As first element, the identifier of the specific group or topic.

  * Optionally, as second element, the role (or CBOR array of roles) the Client wishes to take in the group.

  The encoding of the group or topic identifier and of the role identifiers is application specific.

o 'audience', with an identifier of a KDC.

o 'req_cnf', as defined in Section 3.1 of [I-D.ietf-ace-oauth-params], optionally containing the public key or a reference to the public key of the Client, if it wishes to communicate that to the AS.

o Other additional parameters as defined in [I-D.ietf-ace-oauth-authz], if necessary.

3.2. Authorization Response

The Authorization Response sent from the AS to the Client is as defined in Section 5.6.2 of [I-D.ietf-ace-oauth-authz] and MUST contain the following parameters:

o 'access_token', containing the proof-of-possession access token.

o 'cnf' if symmetric keys are used, not present if asymmetric keys are used. This parameter is defined in Section 3.2 of [I-D.ietf-ace-oauth-params] and contains the symmetric proof-of-possession key that the Client is supposed to use with the KDC.

- 'rs_cnf' if asymmetric keys are used, not present if symmetric keys are used. This parameter is as defined in Section 3.2 of [I-D.ietf-ace-oauth-params] and contains information about the public key of the KDC.

- 'exp', contains the lifetime in seconds of the access token. This parameter MAY be omitted if the application defines how the expiration time is communicated to the Client via other means, or if it establishes a default value.

Additionally, the Authorization Response MAY contain the following parameters, which, if included, MUST have the corresponding values:

- 'scope', which mirrors the 'scope' parameter in the Authorization Request (see Section 3.1). Its value is a CBOR array encoded as a byte string, containing:

  * As first element, the identifier of the specific group or topic the Client is authorized to access.

  * Optionally, as second element, the role (or CBOR array of roles) the Client is authorized to take in the group.

  The encoding of the group or topic identifier and of the role identifiers is application specific.

- Other additional parameters as defined in [I-D.ietf-ace-oauth-authz], if necessary.

The access token MUST contain all the parameters defined above (including the same 'scope' as in this message, if present, or the 'scope' of the Authorization Request otherwise), and additionally other optional parameters that the transport profile of ACE requires.

When receiving an Authorization Request from a Client that was previously authorized, and which still owns a valid non expired access token, the AS replies with an Authorization Response with a new access token.

3.3. Token Post

The Client sends a CoAP POST request including the access token to the KDC, as specified in Section 5.8.1 of [I-D.ietf-ace-oauth-authz]. If the specific transport profile of ACE defines it, the Client MAY use a different endpoint than /authz-info at the KDC to post the access token to.

Optionally, the Client might need to request necessary information concerning the public keys in the group, as well as concerning the algorithm and related parameters for computing signatures in the group. In such a case, the joining node MAY ask for that information to the KDC in this same request. To this end, it sends the CoAP POST request to the /authz-info endpoint using the Content-Format "application/ace+cbor" defined in Section 8.14 of [I-D.ietf-ace-oauth-authz], and includes also the following parameters:

- 'sign_info' defined in Section 3.3.1, encoding the CBOR simple value Null, to require information and parameters on the signature algorithm and on the public keys used in the group.

- 'pub_key_enc' defined in Section 3.3.2, encoding the CBOR simple value Null, to require information on the exact encoding of public keys used in the group.

The CDDL notation of the 'sign_info' and 'pub_key_enc' parameters formatted as in the request is given below.

```
   sign_info_req = nil

   pub_key_enc_req = nil
```

Alternatively, the joining node may retrieve this information by other means.

After successful verification, the Client is authorized to receive the group keying material from the KDC and join the group. In particular, the KDC replies to the Client with a 2.01 (Created) response, using Content-Format "application/ace+cbor" defined in Section 8.14 of [I-D.ietf-ace-oauth-authz].

The payload of the 2.01 response is a CBOR map, which MUST include a nonce N generated by the KDC. The Client may use this nonce for proving the possession of its own private key (see the 'client_cred_verify' parameter in Section 4).

Optionally, if they were included in the request, the AS MAY include the 'sign_info' parameter as well as the 'pub_key_enc' parameter defined in Section 3.3.1 and Section 3.3.2 of this specification, respectively.

The 'sign_info' parameter MUST be present if the POST request included the 'sign_info' parameter with value Null. If present, the 'sign_info' parameter of the 2.01 (Created) response is a CBOR array formatted as follows.

- o The first element 'sign_alg' is an integer or a text string, indicating the signature algorithm used in the group. It is required of the application profiles to define specific values for this parameter.

- o The second element 'sign_parameters' indicates the parameters of the signature algorithm. Its structure depends on the value of 'sign_alg'. It is required of the application profiles to define specific values for this parameter. If no parameters of the signature algorithm are specified, 'sign_parameters' MUST be encoding the CBOR simple value Null.

- o The third element 'sign_key_parameters' indicates the parameters of the key used with the signature algorithm. Its structure depends on the value of 'sign_alg'. It is required of the application profiles to define specific values for this parameter. If no parameters of the key used with the signature algorithm are specified, 'sign_key_parameters' MUST be encoding the CBOR simple value Null.

The 'pub_key_enc' parameter MUST be present if the POST request included the 'pub_key_enc' parameter with value Null. If present, the 'pub_key_enc' parameter of the 2.01 (Created) response is a CBOR integer, indicating the encoding of public keys used in the group. The values of this field are registered in the "ACE Public Key Encoding" Registry, defined in Section 11.2. It is required of the application profiles to define specific values to use for this parameter.

The CDDL notation of the 'sign_info' and 'pub_key_enc' parameters formatted as in the response is given below.

```
sign_info_res = [
  sign_alg : int / tstr,
  sign_parameters : any / nil,
  sign_key_parameters : any / nil
]

pub_key_enc_res = int
```

Note that the CBOR map specified as payload of the 2.01 (Created) response may include further parameters, e.g. according to the signalled transport profile of ACE.

Note that this step could be merged with the following message from the Client to the KDC, namely Key Distribution Request.

3.3.1. 'sign_info' Parameter

The 'sign_info' parameter is an OPTIONAL parameter of the AS Request Creation Hints message defined in Section 5.1.2. of [I-D.ietf-ace-oauth-authz]. This parameter contains information and parameters about the signature algorithm and the public keys to be used between the Client and the RS. Its exact content is application specific.

3.3.2. 'pub_key_enc' Parameter

The 'pub_key_enc' parameter is an OPTIONAL parameter of the AS Request Creation Hints message defined in Section 5.1.2. of [I-D.ietf-ace-oauth-authz]. This parameter contains information about the exact encoding of public keys to be used between the Client and the RS. Its exact content is application specific.

4. Key Distribution

This section defines how the keying material used for group communication is distributed from the KDC to the Client, when joining the group as a new member.

If not previously established, the Client and the KDC MUST first establish a pairwise secure communication channel using ACE. The exchange of Key Distribution Request-Response MUST occur over that secure channel. The Client and the KDC MAY use that same secure channel to protect further pairwise communications, that MUST be secured.

During this exchange, the Client sends a request to the AS, specifying the group it wishes to join (see Section 4.1). Then, the KDC verifies the access token and that the Client is authorized to join that group; if so, it provides the Client with the keying material to securely communicate with the member of the group (see Section 4.2). The Content-Format used in the messages is set to application/cbor.

Figure 4 gives an overview of the exchange described above.

```
      Client                                              KDC
         |                                                  |
         |---- Key Distribution Request: POST /group-id --->|
         |                                                  |
         |<--- Key Distribution Response: 2.01 (Created) ---|
         |                                                  |
```

Figure 4: Message Flow of Key Distribution to a New Group Member

The same set of message can also be used for the following cases, when the Client is already a group member:

- o The Client wishes to (re-)get the current keying material, for cases such as expiration, loss or suspected mismatch, due to e.g. reboot or missed group rekeying. This is further discussed in Section 6.

- o The Client wishes to (re-)get the public keys of other group members, e.g. if it is aware of new nodes joining the group after itself. This is further discussed in Section 7.

Additionally, the format of the payload of the Key Distribution Response (Section 4.2) can be reused for messages sent by the KDC to distribute updated group keying material, in case of a new node joining the group or of a current member leaving the group. The key management scheme used to send such messages could rely on, e.g., multicast in case of a new node joining or unicast in case of a node leaving the group.

Note that proof-of-possession to bind the access token to the Client is performed by using the proof-of-possession key bound to the access token for establishing secure communication between the Client and the KDC.

If the application requires backward security, the KDC SHALL generate new group keying material and securely distribute it to all the current group members, using the message format defined in this section. Application profiles may define alternative message formats.

4.1. Key Distribution Request

The Client sends a Key Distribution Request to the KDC. This corresponds to a CoAP POST request to the endpoint in the KDC associated to the group to join. The endpoint in the KDC is associated to the 'scope' value of the Authorization Request/ Response. The payload of this request is a CBOR map which MUST contain the following fields:

- o 'type', encoded as a CBOR int, with value 1 ("key distribution").

Additionally, the CBOR map in the payload MAY contain the following fields, which, if included, MUST have the corresponding values:

- o 'scope', with value the specific resource that the Client is authorized to access (i.e. group or topic identifier) and role(s), encoded as in Section 3.1.

- 'get_pub_keys', if the Client wishes to receive the public keys of the other nodes in the group from the KDC. The value is an empty CBOR array. This parameter may be present if the KDC stores the public keys of the nodes in the group and distributes them to the Client; it is useless to have here if the set of public keys of the members of the group is known in another way, e.g. it was provided by the AS.

- 'client_cred', with value the public key or certificate of the Client, encoded as a CBOR byte string. If the KDC is managing (collecting from/distributing to the Client) the public keys of the group members, this field contains the public key of the Client. The default encoding for public keys is COSE Keys. Alternative specific encodings of this parameter MAY be defined in applications of this specification.

- 'client_cred_verify', encoded as a CBOR byte string. This parameter contains a signature computed by the Client over the nonce N received from the KDC in the 2.01 (Created) response to the token POST request (see Section 3.3). The Client computes the signature by using its own private key, whose corresponding public key is either directly specified in the 'client_cred' parameter or included in the certificate specified in the 'client_cred' parameter. This parameter MUST be present if the 'client_cred' parameter is present.

- 'pub_keys_repos', can be present if a certificate is present in the 'client_cred' field, with value a list of public key repositories storing the certificate of the Client. This parameter is encoded as a CBOR array of CBOR text strings, each of which specifies the URI of a key repository.

4.2. Key Distribution Response

The KDC verifies that the 'scope' received in the Key Distribution Request, if present, is a subset of the 'scope' stored in the access token associated to this client. If verification fails, the KDC MUST respond with a 4.01 (Unauthorized) error message.

If the Key Distribution Request is not formatted correctly (e.g. no 'scope' field present while expected, or unknown fields present), the KDC MUST respond with 4.00 (Bad Request) error message.

If verification succeeds, the KDC sends a Key Distribution success Response to the Client. The Key Distribution success Response corresponds to a 2.01 Created message. The payload of this response is a CBOR map, which MUST contain:

- 'kty', identifying the key type of the 'key' parameter. The set of values can be found in the "Key Type" column of the "ACE Groupcomm Key" Registry. Implementations MUST verify that the key type matches the application profile being used, if present, as registered in the "ACE Groupcomm Key" registry.

- 'key', containing the keying material for the group communication, or information required to derive it.

The exact format of the 'key' value MUST be defined in applications of this specification. Additionally, documents specifying the key format MUST register it in the "ACE Groupcomm Key" registry, including its name, type and application profile to be used with, as defined in the "ACE Groupcomm Key" registry, defined in Section 11.5.

Name	Key Type Value	Profile	Description
Reserved	0		This value is reserved

Figure 5: Key Type Values

Optionally, the Key Distribution Response MAY contain the following parameters, which, if included, MUST have the corresponding values:

- 'profile', with value a CBOR integer that MUST be used to uniquely identify the application profile for group communication. The value MUST be registered in the "ACE Groupcomm Profile" Registry.

- 'exp', with value the expiration time of the keying material for the group communication, encoded as a CBOR unsigned integer or floating-point number. This field contains a numeric value representing the number of seconds from 1970-01-01T00:00:00Z UTC until the specified UTC date/time, ignoring leap seconds, analogous to what specified in Section 2 of [RFC7519].

- 'pub_keys', may only be present if 'get_pub_keys' was present in the Key Distribution Request. This parameter is a CBOR byte string, which encodes the public keys of all the group members paired with the respective member identifiers. The default encoding for public keys is COSE Keys, so the default encoding for 'pub_keys' is a CBOR byte string wrapping a COSE_KeySet (see [RFC8152]), which contains the public keys of all the members of the group. In particular, each COSE Key in the COSE_KeySet includes the identifier of the corresponding group member as value of its 'kid' key parameter. Alternative specific encodings of

this parameter MAY be defined in applications of this specification.

- o 'group_policies', with value a CBOR map, whose entries specify how the group handles specific management aspects. These include, for instance, approaches to achieve synchronization of sequence numbers among group members. The elements of this field are registered in the "ACE Groupcomm Policy" Registry. This specification defines the two elements "Sequence Number Synchronization Method" and "Key Update Check Interval", which are summarized in Figure 6. Application profiles that build on this document MUST specify the exact content format of included map entries.

Name	CBOR label	CBOR type	Description	Reference
Sequence Number Synchronization Method	TBD1	tstr/int	Method for a recipient node to synchronize with sequence numbers of a sender node. Its value is taken from the 'Value' column of the Sequence Number Synchronization Method registry	[[this document]]
Key Update Check Interval	TBD2	int	Polling interval in seconds, to check for new keying material at the KDC	[[this document]]

Figure 6: ACE Groupcomm Policies

- o 'mgt_key_material', encoded as a CBOR byte string and containing the administrative keying material to participate in the group rekeying performed by the KDC. The exact format and content depend on the specific rekeying scheme used in the group, which may be specified in the application profile.

Specific application profiles that build on this document need to specify how exactly the keying material is used to protect the group communication.

5. Removal of a Node from the Group

This section describes at a high level how a node can be removed from the group.

If the application requires forward security, the KDC SHALL generate new group keying material and securely distribute it to all the current group members but the leaving node, using the message format defined in Section 4.2. Application profiles may define alternative message formats.

5.1. Expired Authorization

If the AS provides Token introspection (see Section 5.7 of [I-D.ietf-ace-oauth-authz]), the KDC can optionally use and check whether:

- the node is not authorized anymore;
- the access token is still valid, upon its expiration.

Either case, once aware that a node is not authorized anymore, the KDC has to remove the unauthorized node from the list of group members, if the KDC keeps track of that.

5.2. Request to Leave the Group

A node can actively request to leave the group. In this case, the Client can send a request formatted as follows to the KDC, to abandon the group. The client MUST use the protected channel established with ACE, mentioned in Section 4.

To request to leave a group, the client MUST send a CoAP POST request to the endpoint in the KDC associated to the group to leave (same endpoint used in Section 4.1 for Key Distribution requests). The payload of this Leave Request is a CBOR map which MUST contain:

- 'type', encoded as a CBOR int, with value 2 ("leave").
- 'scope', with value the specific resource that the Client is authorized to access (i.e. group or topic identifier) and wants to leave, encoded as in Section 3.1. The 'role' field is omitted.

Note that the 'role' field is omitted since such a request should only be used to leave a group altogether. If the leaving node wants to be part of a group with fewer roles, it does not need to communicate that to the KDC, and can simply stop acting according to such roles.

If the Leave Request is such that the KDC cannot extract all the necessary information to understand and process it correctly (e.g. no 'scope' field present), the KDC MUST respond with a 4.00 (Bad Request) error message. Otherwise, the KDC MUST remove the leaving node from the list of group members, if the KDC keeps track of that.

Note that, after having left the group, a node may wish to join it again. Then, as long as the node is still authorized to join the group, i.e. it has a still valid access token, it can re-request to join the group directly to the KDC without needing to retrieve a new access token from the AS. This means that the KDC needs to keep track of nodes with valid access tokens, before deleting all information about the leaving node.

6. Retrieval of New or Updated Keying Material

A node stops using the group keying material upon its expiration, according to the 'exp' parameter specified in the retained COSE Key. Then, if it wants to continue participating in the group communication, the node has to request new updated keying material to the KDC. In this case, and depending on what part of the keying material is expired, the client may need to communicate to the KDC its need for that part to be renewed: for example, if the Client uses an individual key to protect outgoing traffic and has to renew it, the node may request a new one, or new input material to derive it, without renewing the whole group keying material.

The Client may perform the same request to the KDC also upon receiving messages from other group members without being able to retrieve the material to correctly decrypt them. This may be due to a previous update of the group keying material (rekeying) triggered by the KDC, that the Client was not able to receive or decrypt.

Note that policies can be set up so that the Client sends a request to the KDC only after a given number of unsuccessfully decrypted incoming messages. It is application dependent and pertaining to the particular message exchange (e.g. [I-D.ietf-core-oscore-groupcomm]) to set up policies that instruct clients to retain unsuccessfully decrypted messages and for how long, so that they can be decrypted after getting updated keying material, rather than just considered non valid messages to discard right away.

The same request could also be sent by the client without being triggered by a failed decryption of a message, if the client wants to confirm that it has the latest group keying material. If that is the case, the client will receive from the KDC the same group keying material it has in memory.

Note that the difference between the keying material renewal request and the keying material update request is that the first one triggers the KDC to produce new keying material for that node, while the second one only triggers distribution (the renewal might have happened independently, because of expiration). Once a node receives new individual keying material, other group members may need to use the update keying material request to retrieve it.

Alternatively, the re-distribution of keying material can be initiated by the KDC, which e.g.:

- o Can maintain an Observable resource to send notifications to Clients when the keying material is updated. Such a notification would have the same payload as the Key Re-Distribution Response defined in Section 6.2.

- o Can send the payload of the Key Re-Distribution Response as one or multiple multicast requests to the members of the group, using secure rekeying schemes such as [RFC2093][RFC2094][RFC2627].

- o Can send unicast requests to each Client over a secure channel, with the Key Re-Distribution Response as payload.

- o Can act as a publisher in a pub-sub scenario, and update the keying material by publishing on a specific topic on a broker, which all the members of the group are subscribed to.

Note that these methods of KDC-initiated key re-distribution have different security properties and require different security associations.

6.1. Key Re-Distribution Request

To request a re-distribution of keying material, the Client sends a shortened Key Distribution Request to the KDC (Section 4.1), formatted as follows. The payload MUST contain the following fields:

- o 'type', encoded as a CBOR int, with value 3 ("update key") if the request is intended to retrieve updated group keying material, and 4 ("new") if the request is intended for the KDC to produce and provide new individual keying material for the Client.

- o 'scope', which contains only the identifier of the specific group or topic, encoded as in Section 3.1. That is, the role field is not present.

6.2. Key Re-Distribution Response

The KDC receiving a Key Re-Distribution Request MUST check that it is storing a valid access token from that client for that scope.

If that is not the case, i.e. it does not store the token or the token is not valid for that client for the scope requested, the KDC MUST respond with a 4.01 (Unauthorized) error message. Analogously to Section 4.2, if the Key Re-Distribution Request is not formatted correctly (e.g. no 'scope' field present, or unknown fields present), the KDC MUST respond with a 4.00 (Bad Request) error message.

Otherwise, the KDC replies to the Client with a Key Distribution Response, which MUST include the 'kty', 'key' and 'exp' parameters specified in Section 4.2. The Key Distribution Response MAY also include the 'profile', 'group_policies' and 'mgt_key_material' parameters specified in Section 4.2.

Note that this response might simply re-provide the same keying material currently owned by the Client, if it has not been renewed.

7. Retrieval of Public Keys for Group Members

In case the KDC maintains the public keys of group members, a node in the group can contact the KDC to request public keys of either all group members or a specified subset, using the messages defined below.

Figure 7 gives an overview of the exchange described above.

```
      Client                                          KDC
         |                                             |
         |---- Public Key Request: POST /group-id --->|
         |                                             |
         |<--- Public Key Response: 2.01 (Created) ---|
         |                                             |
```

Figure 7: Message Flow of Public Key Request-Response

Note that these messages can be combined with the Key Re-Distribution messages in Section 6, to request at the same time the keying material and the public keys. In this case, either a new endpoint at the KDC may be used, or additional information needs to be sent in the request payload, to distinguish these combined messages from the Public Key messages described below, since they would be identical otherwise.

7.1. Public Key Request

To request public keys, the Client sends a shortened Key Distribution Request to the KDC (Section 4.1), formatted as follows. The payload of this request MUST contain the following fields:

- 'type', encoded as a CBOR int, with value 5 ("pub keys").

- 'get_pub_keys', which has as value a CBOR array including either:

  * no elements, i.e. an empty array, in order to request the public key of all current group members; or

  * N elements, each of which is the identifier of a group member encoded as a CBOR byte string, in order to request the public key of the specified nodes.

- 'scope', which contains only the identifier of the specific group or topic, encoded as in Section 3.1. That is, the role field is not present.

7.2. Public Key Response

The KDC replies to the Client with a Key Distribution Response containing only the 'pub_keys' parameter, as specified in Section 4.2. The payload of this response contains the following field:

- 'pub_keys', which contains either:

  * the public keys of all the members of the group, if the 'get_pub_keys' parameter of the Public Key request was an empty array; or

  * the public keys of the group members with the identifiers specified in the 'get_pub_keys' parameter of the Public Key request.

The KDC may enforce one of the following policies, in order to handle possible identifiers that are included in the 'get_pub_keys' parameter of the Public Key request but are not associated to any current group member.

- The KDC silently ignores those identifiers.

- The KDC retains public keys of group members for a given amount of time after their leaving, before discarding them. As long as such

public keys are retained, the KDC provides them to a requesting Client.

Either case, a node that has left the group should not expect any of its outgoing messages to be successfully processed, if received after its leaving, due to a possible group rekeying occurred before the message reception.

8. ACE Groupcomm Parameters

This specification defines a number of fields used during the message exchange. The table below summarizes them, and specifies the CBOR key to use instead of the full descriptive name.

Name	CBOR Key	CBOR Type
scope	TBD	array
get_pub_keys	TBD	array
client_cred	TBD	byte string
client_cred_verify	TBD	byte string
pub_keys_repos	TBD	array
kty	TBD	int / byte string
key	TBD	see "ACE Groupcomm Key" Registry
profile	TBD	int
exp	TBD	int / float
pub_keys	TBD	byte string
group_policies	TBD	map
mgt_key_material	TBD	byte string
type	TBD	int

9. ACE Groupcomm Request Type

This specification defines a number of types of requests. The table below summarizes them.

Name	Value
key distribution	1
leave	2
update key	3
new	4
pub keys	5

10. Security Considerations

When a Client receives a message from a sender for the first time, it needs to have a mechanism in place to avoid replay, e.g. Appendix B.2 of [I-D.ietf-core-object-security].

The KDC must renew the group keying material upon its expiration.

The KDC should renew the keying material upon group membership change, and should provide it to the current group members through the rekeying scheme used in the group.

The KDC may enforce a rekeying policy that takes into account the overall time required to rekey the group, as well as the expected rate of changes in the group membership.

That is, the KDC may not rekey the group at every membership change, for instance if members' joining and leaving occur frequently and performing a group rekeying takes too long. Instead, the KDC may rekey the group after a minum number of group members have joined or left within a given time interval, or during predictable network inactivity periods.

However, this would result in the KDC not constantly preserving backward and forward security. In fact, newly joining group members could be able to access the keying material used before their joining, and thus could access past group communications. Also, until the KDC performs a group rekeying, the newly leaving nodes would still be able to access upcoming group communications that are protected with the keying material that has not yet been updated.

10.1. Update of Keying Material

A group member can receive a message shortly after the group has been rekeyed, and new keying material has been distributed by the KDC. In the following two cases, this may result in misaligned keying material between the group members.

In the first case, the sender protects a message using the old keying material. However, the recipient receives the message after having received the new keying material, hence not being able to correctly process it. A possible way to ameliorate this issue is to preserve the old, recent, keying material for a maximum amount of time defined by the application. By doing so, the recipient can still try to process the received message using the old retained keying material as second attempt. Note that a former (compromised) group member can take advantage of this by sending messages protected with the old retained keying material. Therefore, a conservative application policy should not admit the storage of old keying material.

In the second case, the sender protects a message using the new keying material, but the recipient receives that request before having received the new keying material. Therefore, the recipient would not be able to correctly process the request and hence discards it. If the recipient receives the new keying material shortly after that and the sender endpoint uses CoAP retransmissions, the former will still be able to receive and correctly process the message. In any case, the recipient should actively ask the KDC for an updated keying material according to an application-defined policy, for instance after a given number of unsuccessfully decrypted incoming messages.

10.2. Block-Wise Considerations

If the block-wise options [RFC7959] are used, and the keying material is updated in the middle of a block-wise transfer, the sender of the blocks just changes the keying material to the updated one and continues the transfer. As long as both sides get the new keying material, updating the keying material in the middle of a transfer will not cause any issue. Otherwise, the sender will have to transmit the message again, when receiving an error message from the recipient.

Compared to a scenario where the transfer does not use block-wise, depending on how fast the keying material is changed, the nodes might consume a larger amount of the network resending the blocks again and again, which might be problematic.

11. IANA Considerations

This document has the following actions for IANA.

11.1. ACE Authorization Server Request Creation Hints Registry

IANA is asked to register the following entries in the "ACE Authorization Server Request Creation Hints" Registry defined in Section 8.1 of [I-D.ietf-ace-oauth-authz].

- Name: sign_info
- CBOR Key: TBD (range -256 to 255)
- Value Type: any
- Reference: [[This specification]]
- Name: pub_key_enc
- CBOR Key: TBD (range -256 to 255)
- Value Type: integer
- Reference: [[This specification]]

11.2. ACE Public Key Encoding Registry

This specification establishes the "ACE Public Key Encoding" IANA Registry. The Registry has been created to use the "Expert Review Required" registration procedure [RFC8126]. Expert review guidelines are provided in Section 11.9. It should be noted that, in addition to the expert review, some portions of the Registry require a specification, potentially a Standards Track RFC, be supplied as well.

The columns of this Registry are:

- Name: This is a descriptive name that enables easier reference to the item. The name MUST be unique. It is not used in the encoding.
- Value: The value to be used to identify this public key encoding. This value MUST be unique. The value can be a positive or a negative integer. Integer values between 0 and 255 are designated as Standards Track Document required. Integer values from 256 to 65535 are designated as Specification Required. Integer values of

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 for this public key encoding.

- Reference: This field contains a pointer to the public specification providing the public key encoding, if one exists.

The value 0 is to be marked as "Reserved".

11.3. ACE Groupcomm Parameters Registry

This specification establishes the "ACE Groupcomm Parameters" IANA Registry. The Registry has been created to use the "Expert Review Required" registration procedure [RFC8126]. Expert review guidelines are provided in Section 11.9.

The columns of this Registry are:

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

- CBOR Key: This is the value used as CBOR key of the item. These values MUST be unique. The value can be a positive integer, a negative integer, or a string.

- CBOR Type: This 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 contains a pointer to the public specification for the format of the item, if one exists.

This Registry has been initially populated by the values in Section 8. The specification column for all of these entries will be this document.

11.4. Ace Groupcomm Request Type Registry

This specification establishes the "ACE Groupcomm Request Type" IANA Registry. The Registry has been created to use the "Expert Review Required" registration procedure [RFC8126]. Expert review guidelines are provided in Section 11.9.

The columns of this Registry are:

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

- o Value: This is the value used to identify the request. These values MUST be unique. The value must be a positive integer.

- o Reference: This contains a pointer to the public specification for the format of the item, if one exists.

This Registry has been initially populated by the values in Section 9. The reference column for all of these entries will be this document. The value 0 is to be marked as "Reserved".

11.5. ACE Groupcomm Key Registry

This specification establishes the "ACE Groupcomm Key" IANA Registry. The Registry has been created to use the "Expert Review Required" registration procedure [RFC8126]. Expert review guidelines are provided in Section 11.9.

The columns of this Registry are:

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

- o Key Type Value: This is the value used to identify the keying material. These values MUST be unique. The value can be a positive integer, a negative integer, or a string.

- o Profile: This field may contain one or more descriptive strings of application profiles to be used with this item. The values should be taken from the Name column of the "ACE Groupcomm Profile" Registry.

- o Description: This field contains a brief description of the keying material.

- o References: This contains a pointer to the public specification for the format of the keying material, if one exists.

This Registry has been initially populated by the values in Figure 5. The specification column for all of these entries will be this document.

11.6. ACE Groupcomm Profile Registry

This specification establishes the "ACE Groupcomm Profile" IANA Registry. The Registry has been created to use the "Expert Review Required" registration procedure [RFC8126]. Expert review guidelines are provided in Section 11.9. It should be noted that, in addition to the expert review, some portions of the Registry require a specification, potentially a Standards Track RFC, be supplied as well.

The columns of this Registry are:

- Name: The name of the application profile, to be used as value of the profile attribute.

- Description: Text giving an overview of the application profile and the context it is developed for.

- CBOR Value: CBOR abbreviation for the name of this application profile. Different ranges of values use different registration policies [RFC8126]. Integer values from -256 to 255 are designated as Standards Action. 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.

- Reference: This contains a pointer to the public specification of the abbreviation for this application profile, if one exists.

11.7. ACE Groupcomm Policy Registry

This specification establishes the "ACE Groupcomm Policy" IANA Registry. The Registry has been created to use the "Expert Review Required" registration procedure [RFC8126]. Expert review guidelines are provided in Section 11.9. It should be noted that, in addition to the expert review, some portions of the Registry require a specification, potentially a Standards Track RFC, be supplied as well.

The columns of this Registry are:

- Name: The name of the group communication policy.

- CBOR label: The value to be used to identify this group communication policy. Key map labels MUST be unique. The label can be a positive integer, a negative integer or a string. Integer values between 0 and 255 and strings of length 1 are

designated as Standards Track Document required. Integer values from 256 to 65535 and strings of length 2 are designated as Specification Required. Integer values of greater than 65535 and strings of length greater than 2 are designated as expert review. Integer values less than -65536 are marked as private use.

- o CBOR type: the CBOR type used to encode the value of this group communication policy.
- o Description: This field contains a brief description for this group communication policy.
- o Reference: This field contains a pointer to the public specification providing the format of the group communication policy, if one exists.

This registry will be initially populated by the values in Figure 6.

11.8. Sequence Number Synchronization Method Registry

This specification establishes the "Sequence Number Synchronization Method" IANA Registry. The Registry has been created to use the "Expert Review Required" registration procedure [RFC8126]. Expert review guidelines are provided in Section 11.9. It should be noted that, in addition to the expert review, some portions of the Registry require a specification, potentially a Standards Track RFC, be supplied as well.

The columns of this Registry are:

- o Name: The name of the sequence number synchronization method.
- o Value: The value to be used to identify this sequence number synchronization method.
- o Description: This field contains a brief description for this sequence number synchronization method.
- o Reference: This field contains a pointer to the public specification describing the sequence number synchronization method.

11.9. 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 track 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 first-come, first-serve 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.

12. References

12.1. Normative References

[I-D.ietf-ace-oauth-authz]
Seitz, L., Selander, G., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "Authentication and Authorization for Constrained Environments (ACE) using the OAuth 2.0 Framework (ACE-OAuth)", draft-ietf-ace-oauth-authz-24 (work in progress), March 2019.

[I-D.ietf-ace-oauth-params]
Seitz, L., "Additional OAuth Parameters for Authorization in Constrained Environments (ACE)", draft-ietf-ace-oauth-params-05 (work in progress), March 2019.

[I-D.ietf-core-oscore-groupcomm]
Tiloca, M., Selander, G., Palombini, F., and J. Park, "Group OSCORE - Secure Group Communication for CoAP", draft-ietf-core-oscore-groupcomm-05 (work in progress), July 2019.

[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>.

[RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", RFC 7049, DOI 10.17487/RFC7049, October 2013, <https://www.rfc-editor.org/info/rfc7049>.

[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>.

[RFC8152] Schaad, J., "CBOR Object Signing and Encryption (COSE)", RFC 8152, DOI 10.17487/RFC8152, July 2017, <https://www.rfc-editor.org/info/rfc8152>.

12.2. Informative References

[I-D.dijk-core-groupcomm-bis]
Dijk, E., Wang, C., and M. Tiloca, "Group Communication for the Constrained Application Protocol (CoAP)", draft-dijk-core-groupcomm-bis-00 (work in progress), March 2019.

[I-D.ietf-ace-dtls-authorize]
Gerdes, S., Bergmann, O., Bormann, C., Selander, G., and L. Seitz, "Datagram Transport Layer Security (DTLS) Profile for Authentication and Authorization for Constrained Environments (ACE)", draft-ietf-ace-dtls-authorize-08 (work in progress), April 2019.

[I-D.ietf-ace-mqtt-tls-profile]
Sengul, C., Kirby, A., and P. Fremantle, "MQTT-TLS profile of ACE", draft-ietf-ace-mqtt-tls-profile-00 (work in progress), May 2019.

[I-D.ietf-ace-oscore-profile]
Palombini, F., Seitz, L., Selander, G., and M. Gunnarsson, "OSCORE profile of the Authentication and Authorization for Constrained Environments Framework", draft-ietf-ace-oscore-profile-07 (work in progress), February 2019.

[I-D.ietf-core-coap-pubsub]
Koster, M., Keranen, A., and J. Jimenez, "Publish-Subscribe Broker for the Constrained Application Protocol (CoAP)", draft-ietf-core-coap-pubsub-08 (work in progress), March 2019.

[I-D.ietf-core-object-security]
Selander, G., Mattsson, J., Palombini, F., and L. Seitz, "Object Security for Constrained RESTful Environments (OSCORE)", draft-ietf-core-object-security-16 (work in progress), March 2019.

[RFC2093] Harney, H. and C. Muckenhirn, "Group Key Management Protocol (GKMP) Specification", RFC 2093, DOI 10.17487/RFC2093, July 1997, <https://www.rfc-editor.org/info/rfc2093>.

[RFC2094] Harney, H. and C. Muckenhirn, "Group Key Management Protocol (GKMP) Architecture", RFC 2094, DOI 10.17487/RFC2094, July 1997, <https://www.rfc-editor.org/info/rfc2094>.

[RFC2627] Wallner, D., Harder, E., and R. Agee, "Key Management for Multicast: Issues and Architectures", RFC 2627, DOI 10.17487/RFC2627, June 1999, <https://www.rfc-editor.org/info/rfc2627>.

[RFC7390] Rahman, A., Ed. and E. Dijk, Ed., "Group Communication for the Constrained Application Protocol (CoAP)", RFC 7390, DOI 10.17487/RFC7390, October 2014, <https://www.rfc-editor.org/info/rfc7390>.

[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>.

[RFC7959] Bormann, C. and Z. Shelby, Ed., "Block-Wise Transfers in the Constrained Application Protocol (CoAP)", RFC 7959, DOI 10.17487/RFC7959, August 2016, <https://www.rfc-editor.org/info/rfc7959>.

[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>.

Appendix A. Requirements on Application Profiles

This section lists the requirements on application profiles of this specification,for the convenience of application profile designers.

- o Specify the communication protocol the members of the group must use (e.g., multicast CoAP).

- o Specify the security protocol the group members must use to protect their communication (e.g., group OSCORE). This must provide encryption, integrity and replay protection.

- o Specify the encoding and value of the identifier of group or topic and role of 'scope' (see Section 3.1).

- o Specify and register the application profile identifier (see Section 4.1).

- o Specify the acceptable values of 'kty' (see Section 4.2).

- o Specify the format and content of 'group_policies' entries (see Section 4.2).

- o Optionally, specify the format and content of 'mgt_key_material' (see Section 4.2).

- o Optionally, specify tranport profile of ACE [I-D.ietf-ace-oauth-authz] to use between Client and KDC.

- o Optionally, specify the encoding of public keys, of 'client_cred', and of 'pub_keys' if COSE_Keys are not used (see Section 4.2).

- o Optionally, specify the acceptable values for parameters related to signature algorithm and signature keys: 'sign_alg', 'sign_parameters', 'sign_key_parameters', 'pub_key_enc' (see Section 3.3).

- o Optionally, specify the negotiation of parameter values for signature algorithm and signature keys, if 'sign_info' and 'pub_key_enc' are not used (see Section 3.3).

Appendix B. Document Updates

RFC EDITOR: PLEASE REMOVE THIS SECTION.

B.1. Version -01 to -02

- Editorial fixes.

- Distinction between transport profile and application profile (Section 1.1).

- New parameters 'sign_info' and 'pub_key_enc' to negotiate parameter values for signature algorithm and signature keys (Section 3.3).

- New parameter 'type' to distinguish different Key Distribution Request messages (Section 4.1).

- New parameter 'client_cred_verify' in the Key Distribution Request to convey a Client signature (Section 4.1).

- Encoding of 'pub_keys_repos' (Section 4.1).

- Encoding of 'mgt_key_material' (Section 4.1).

- Improved description on retrieval of new or updated keying material (Section 6).

- Encoding of 'get_pub_keys' in Public Key Request (Section 7.1).

- Extended security considerations (Sections 10.1 and 10.2).

- New "ACE Public Key Encoding" IANA Registry (Section 11.2).

- New "ACE Groupcomm Parameters" IANA Registry (Section 11.3), populated with the entries in Section 8.

- New "Ace Groupcomm Request Type" IANA Registry (Section 11.4), populated with the values in Section 9.

- New "ACE Groupcomm Policy" IANA Registry (Section 11.7) populated with two entries "Sequence Number Synchronization Method" and "Key Update Check Interval" (Section 4.2).

- Improved list of requirements for application profiles (Appendix A).

B.2. Version -00 to -01

- Changed name of 'req_aud' to 'audience' in the Authorization Request (Section 3.1).

- Defined error handling on the KDC (Sections 4.2 and 6.2).

- Updated format of the Key Distribution Response as a whole (Section 4.2).

- Generalized format of 'pub_keys' in the Key Distribution Response (Section 4.2).

- Defined format for the message to request leaving the group (Section 5.2).

- Renewal of individual keying material and methods for group rekeying initiated by the KDC (Section 6).

- CBOR type for node identifiers in 'get_pub_keys' (Section 7.1).

- Added section on parameter identifiers and their CBOR keys (Section 8).

- Added request types for requests to a Join Response (Section 9).

- Extended security considerations (Section 10).

- New IANA registries "ACE Groupcomm Key Registry", "ACE Groupcomm Profile Registry", "ACE Groupcomm Policy Registry" and "Sequence Number Synchronization Method Registry" (Section 11).

- Added appendix about requirements for application profiles of ACE on group communication (Appendix A).

Acknowledgments

The following individuals were helpful in shaping this document: Ben Kaduk, John Mattsson, Jim Schaad, Ludwig Seitz, Goeran Selander and Peter van der Stok.

The work on this document has been partly supported by VINNOVA and the Celtic-Next project CRITISEC; and by the EIT-Digital High Impact Initiative ACTIVE.

Authors' Addresses

Francesca Palombini
Ericsson AB
Torshamnsgatan 23
Kista SE-16440 Stockholm
Sweden

Email: francesca.palombini@ericsson.com

Marco Tiloca
RISE AB
Isafjordsgatan 22
Kista SE-16440 Stockholm
Sweden

Email: marco.tiloca@ri.se