ACE Working Group
Internet-Draft
Intended status: Standards Track
Expires: 7 September 2023

M. Tiloca
RISE AB
J. Park
Universitaet Duisburg-Essen
F. Palombini
Ericsson AB

6 March 2023

Key Management for OSCORE Groups in ACE
draft-ietf-ace-key-groupcomm-oscore-16

Abstract

This document defines an application profile of the ACE framework for Authentication and Authorization, to request and provision keying material in group communication scenarios that are based on CoAP and are secured with Group Object Security for Constrained RESTful Environments (Group OSCORE). This application profile delegates the authentication and authorization of Clients, that join an OSCORE group through a Resource Server acting as Group Manager for that group. This application profile leverages protocol-specific transport profiles of ACE to achieve communication security, server authentication and proof-of-possession for a key owned by the Client and bound to an OAuth 2.0 Access Token.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Object Security for Constrained RESTful Environments (OSCORE) [RFC8613] is a method for application-layer protection of the Constrained Application Protocol (CoAP) [RFC7252], using CBOR Object Signing and Encryption (COSE) [RFC9052][RFC9053] and enabling end-to-end security of CoAP payload and options.

As described in [I-D.ietf-core-oscore-groupcomm], Group OSCORE is used to protect CoAP group communication [I-D.ietf-core-groupcomm-bis], which can employ, for example, IP multicast as underlying data transport. This relies on a Group Manager, which is responsible for managing an OSCORE group and enables the group members to exchange CoAP messages secured with Group OSCORE. The Group Manager can be responsible for multiple groups, coordinates the joining process of new group members, and is entrusted with the distribution and renewal of group keying material.

This document is an application profile of [I-D.ietf-ace-key-groupcomm], which itself builds on the ACE framework for Authentication and Authorization [RFC9200]. Message exchanges among the participants as well as message formats and processing follow what specified in [I-D.ietf-ace-key-groupcomm] for provisioning and renewing keying material in group communication scenarios, where Group OSCORE is used to protect CoAP group communication.

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] and in the Authorization Information Format (AIF) [RFC9237] to express authorization information. The terminology for entities in the considered architecture is defined in OAuth 2.0 [RFC6749]. This includes Client (C), Resource Server (RS), and Authorization Server (AS).

* The terms and concepts related to the message formats and processing specified in [I-D.ietf-ace-key-groupcomm], for provisioning and renewing keying material in group communication scenarios. These include the abbreviations REQx and OPTx denoting the numbered mandatory-to-address and optional-to-address requirements, respectively.

* The terms and concepts described in CBOR [RFC8949] and COSE [RFC9052][RFC9053].

* The terms and concepts described in CoAP [RFC7252] and group communication for CoAP [I-D.ietf-core-groupcomm-bis]. 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".

* The terms and concepts for protection and processing of CoAP messages through OSCORE [RFC8613] and through Group OSCORE [I-D.ietf-core-oscore-groupcomm] in group communication scenarios. These especially include:

  - Group Manager, as the entity responsible for a set of groups where communications are secured with Group OSCORE. In this document, the Group Manager acts as Resource Server.

  - Authentication credential, as the set of information associated with an entity, including that entity's public key and parameters associated with the public key. Examples of authentication credentials are CBOR Web Tokens (CWTs) and CWT Claims Sets (CCSs) [RFC8392], X.509 certificates [RFC7925] and C509 certificates [I-D.ietf-cose-cbor-encoded-cert].

Additionally, this document makes use of the following terminology.

- Requester: member of an OSCORE group that sends request messages to other members of the group.

- Responder: member of an OSCORE group that receives request messages from other members of the group. A responder may reply back, by sending a response message to the requester which has sent the request message.

- Monitor: member of an OSCORE group that is configured as responder and never replies back to requesters after receiving request messages. This corresponds to the term "silent server" used in [I-D.ietf-core-oscore-groupcomm].

- Signature verifier: entity external to the OSCORE group and intended to verify the signature of messages exchanged in the group (see Sections 3.1 and 8.5 of [I-D.ietf-core-oscore-groupcomm]). An authorized signature verifier does not join the OSCORE group as an actual member, yet it can retrieve the authentication credentials of the current group members from the Group Manager.

- Signature-only group: an OSCORE group that uses only the group mode (see Section 8 of [I-D.ietf-core-oscore-groupcomm]).

- Pairwise-only group: an OSCORE group that uses only the pairwise mode (see Section 9 of [I-D.ietf-core-oscore-groupcomm]).

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

2. Protocol Overview

Group communication for CoAP has been enabled in [I-D.ietf-core-groupcomm-bis] and can be secured with Group Object Security for Constrained RESTful Environments (Group OSCORE) as specified in [I-D.ietf-core-oscore-groupcomm]. A network node joins an OSCORE group by interacting with the responsible Group Manager. Once registered in the group, the new node can securely exchange messages with other group members.

This document describes how to use [I-D.ietf-ace-key-groupcomm] and [RFC9200] to perform a number of authentication, authorization and key distribution actions as overviewed in Section 2 of [I-D.ietf-ace-key-groupcomm], when the considered group is specifically an OSCORE group.

With reference to [I-D.ietf-ace-key-groupcomm]:

* The node wishing to join the OSCORE group, i.e., the joining node, is the Client.

* The Group Manager is the Key Distribution Center (KDC), acting as a Resource Server.

* The Authorization Server associated with the Group Manager is the AS.

A node performs the steps described in Sections 3 and 4.3.1.1 of [I-D.ietf-ace-key-groupcomm] in order to obtain an authorization for joining an OSCORE group and then to join that group. The format and processing of messages exchanged during such steps are further specified in Section 5 and Section 6 of this document.

All communications between the involved entities MUST be secured.

In particular, communications between the Client and the Group Manager leverage protocol-specific transport profiles of ACE to achieve communication security, proof-of-possession and server authentication. It is expected that, in the commonly referred base-case of this document, the transport profile to use is pre-configured and well-known to nodes participating in constrained applications.

With respect to what is defined in [I-D.ietf-ace-key-groupcomm]:

* The interface provided by the Group Manager extends the original interface defined in Section 4.1 of [I-D.ietf-ace-key-groupcomm] for the KDC, as specified in Section 8 of this document.

* In addition to those defined in Section 8 of [I-D.ietf-ace-key-groupcomm], additional parameters are defined in this document and summarized in Section 12.

* In addition to those defined in Section 9 of [I-D.ietf-ace-key-groupcomm], additional error identifiers are defined in this document and summarized in Section 13.

Finally, Appendix A lists the specifications on this application profile of ACE, based on the requirements defined in Appendix A of [I-D.ietf-ace-key-groupcomm].

3. Format of Scope

Building on Section 3.1 of [I-D.ietf-ace-key-groupcomm], this section defines the exact format and encoding of scope used in this profile.

To this end, this profile uses the Authorization Information Format (AIF) [RFC9237]. With reference to the generic AIF model

```
AIF-Generic<Toid, Tperm> = [* [Toid, Tperm]]
```

the value of the CBOR byte string used as scope encodes the CBOR array [* [Toid, Tperm]], where each [Toid, Tperm] element corresponds to one scope entry.

Furthermore, this document defines the new AIF specific data model AIF-OSCORE-GROUPCOMM, that this profile MUST use to format and encode scope entries.

In particular, the following holds for each scope entry.

* The object identifier ("Toid") is specialized as a CBOR item specifying the name of the groups pertaining to the scope entry.

* The permission set ("Tperm") is specialized as a CBOR unsigned integer with value R, specifying the permissions that the Client wishes to have in the groups indicated by "Toid".

More specifically, the following applies when, as defined in this document, a scope entry includes as set of permissions the set of roles to take in an OSCORE group.

* The object identifier ("Toid") is a CBOR text string, specifying the group name for the scope entry.

* The permission set ("Tperm") is a CBOR unsigned integer with value R, specifying the role(s) that the Client wishes to take in the group (REQ1). The value R is computed as follows.

  - Each role in the permission set is converted into the corresponding numeric identifier X from the "Value" column of the "Group OSCORE Roles" registry, for which this document defines the entries in Figure 1.

   - The set of N numbers is converted into the single value R, by taking two to the power of each numeric identifier X_1, X_2, ..., X_N, and then computing the inclusive OR of the binary representations of all the power values.

Name	Value	Description
Reserved	0	This value is reserved
Requester	1	Send requests; receive responses
Responder	2	Send responses; receive requests
Monitor	3	Receive requests; never send requests/responses
Verifier	4	Verify signature of intercepted messages

Figure 1: Numeric identifier of roles in an OSCORE group

The following CDDL [RFC8610] notation defines a scope entry that uses the AIF-OSCORE-GROUPCOMM data model and expresses a set of Group OSCORE roles from those in Figure 1.

```
AIF-OSCORE-GROUPCOMM = AIF-Generic<oscore-gname, oscore-gperm>

oscore-gname = tstr  ; Group name
oscore-gperm = uint .bits group-oscore-roles

group-oscore-roles = &(
   Requester: 1,
   Responder: 2,
   Monitor: 3,
   Verifier: 4
)

scope_entry = [oscore-gname, oscore-gperm]
```

Future specifications that define new Group OSCORE roles MUST register a corresponding numeric identifier in the "Group OSCORE Roles" registry defined in Section 16.10 of this document.

Note that the value 0 is not available to use as numeric identifier to specify a Group OSCORE role. It follows that, when expressing Group OSCORE roles to take in a group as per this document, a scope entry has the least significant bit of "Tperm" always set to 0.

This is an explicit feature of the AIF-OSCORE-GROUPCOMM data model. That is, for each scope entry, the least significant bit of "Tperm" set to 0 explicitly identifies the scope entry as exactly expressing a set of Group OSCORE roles ("Tperm"), pertaining to a single group whose name is specified by the string literal in "Toid".

Instead, by relying on the same AIF-OSCORE-GROUPCOMM data model, [I-D.ietf-ace-oscore-gm-admin] defines the format of scope entries for Administrator Clients that wish to access an admin interface at the Group Manager. In such scope entries, the least significant bit of "Tperm" is always set to 1.

4. Authentication Credentials

Source authentication of a message sent within the group and protected with Group OSCORE is ensured by means of a digital signature embedded in the message (in group mode), or by integrity-protecting the message with pairwise keying material derived from the asymmetric keys of sender and recipient (in pairwise mode).

Therefore, group members must be able to retrieve each other’s authentication credential from a trusted repository, in order to verify source authenticity of incoming group messages.

As also discussed in [I-D.ietf-core-oscore-groupcomm], the Group Manager acts as trusted repository of the authentication credentials of the group members, and provides those authentication credentials to group members if requested to. Upon joining an OSCORE group, a joining node is thus expected to provide its own authentication credential to the Group Manager.

In particular, the following applies when a node joins an OSCORE group.

* The joining node is going to join the group exclusively as monitor, i.e., it is not going to send messages to the group. In this case, the joining node is not required to provide its own authentication credential to the Group Manager, which thus does not have to perform any check related to the format of the authentication credential, to a signature or ECDH algorithm, and to possible parameters associated with the algorithm and the public key. In case the joining node still provides an authentication credential in the ’client_cred’ parameter of the Join Request (see Section 6.1), the Group Manager silently ignores that parameter, as well as the related parameters ’cnonce’ and ’client_cred_verify’.

- The Group Manager already acquired the authentication credential of the joining node during a past joining process. In this case, the joining node MAY choose not to provide again its own authentication credential to the Group Manager, in order to limit the size of the Join Request. The joining node MUST provide its own authentication credential again if it has provided the Group Manager with multiple authentication credentials during past joining processes, intended for different OSCORE groups. If the joining node provides its own authentication credential, the Group Manager performs consistency checks as per Section 6.2 and, in case of success, considers it as the authentication credential associated with the joining node in the OSCORE group.

- The joining node and the Group Manager use an asymmetric proof-of-possession key to establish a secure communication association. Then, two cases can occur.

  1. When establishing the secure communication association, the Group Manager obtained from the joining node the joining node’s authentication credential, in the format used in the OSCORE group and including the asymmetric proof-of-possession key as public key. Also, such authentication credential and the proof-of-possession key are compatible with the signature or ECDH algorithm, and possible associated parameters used in the OSCORE group.

     In this case, the Group Manager considers the authentication credential as the one associated with the joining node in the OSCORE group. If the joining node is aware that the authentication credential and the public key included thereof are also valid for the OSCORE group, then the joining node MAY choose to not provide again its own authentication credential to the Group Manager.

     The joining node MUST provide again its own authentication credential if it has provided the Group Manager with multiple authentication credentials during past joining processes, intended for different OSCORE groups. If the joining node provides its own authentication credential in the ’client_cred’ parameter of the Join Request (see Section 6.1), the Group Manager performs consistency checks as per Section 6.2 and, in case of success, considers it as the authentication credential associated with the joining node in the OSCORE group.

2. The authentication credential is not in the format used in the OSCORE group, or else the authentication credential and the proof-of-possession key included as public key are not compatible with the signature or ECDH algorithm, and possible associated parameters used in the OSCORE group.

   In this case, the joining node MUST provide a different compatible authentication credential and public key included thereof to the Group Manager in the 'client_cred' parameter of the Join Request (see Section 6.1). Then, the Group Manager performs consistency checks on this latest provided authentication credential as per Section 6.2 and, in case of success, considers it as the authentication credential associated with the joining node in the OSCORE group.

* The joining node and the Group Manager use a symmetric proof-of-possession key to establish a secure communication association. In this case, upon performing a joining process with that Group Manager for the first time, the joining node specifies its own authentication credential in the 'client_cred' parameter of the Join Request (see Section 6.1).

5. Authorization to Join a Group

This section builds on Section 3 of [I-D.ietf-ace-key-groupcomm] and is organized as follows.

First, Section 5.1 and Section 5.2 describe how the joining node interacts with the AS, in order to be authorized to join an OSCORE group under a given Group Manager and to obtain an Access Token. Then, Section 5.3 describes how the joining node transfers the obtained Access Token to the Group Manager. The following considers a joining node that intends to contact the Group Manager for the first time.

Note that what is defined in Section 3 of [I-D.ietf-ace-key-groupcomm] applies, and only additions or modifications to that specification are defined in this document.

5.1. Authorization Request

The Authorization Request message is as defined in Section 3.1 of [I-D.ietf-ace-key-groupcomm], with the following additions.

* If the 'scope' parameter is present:

- The value of the CBOR byte string encodes a CBOR array, whose format MUST follow the data model AIF-OSCORE-GROUPCOMM defined in Section 3. For each OSCORE group to join:

  o The group name is encoded as a CBOR text string.

  o The set of requested roles is expressed as a single CBOR unsigned integer. This is computed as defined in Section 3, from the numerical abbreviations of each requested role defined in the "Group OSCORE Roles" registry, for which this document defines the entries in Figure 1 (REQ1).

5.2. Authorization Response

The Authorization Response message is as defined in Section 3.2 of [I-D.ietf-ace-key-groupcomm], with the following additions:

* The AS MUST include the 'expires_in' parameter. Other means for the AS to specify the lifetime of Access Tokens are out of the scope of this document.

* The AS MUST include the 'scope' parameter, when the value included in the Access Token differs from the one specified by the joining node in the Authorization Request. In such a case, the second element of each scope entry MUST be present, and specifies the set of roles that the joining node is actually authorized to take in the OSCORE group for that scope entry, encoded as specified in Section 5.1.

Furthermore, the AS MAY use the extended format of scope defined in Section 7 of [I-D.ietf-ace-key-groupcomm] for the 'scope' claim of the Access Token. In such a case, the AS MUST use the CBOR tag with tag number TAG_NUMBER, associated with the CoAP Content-Format CF_ID for the media type application/aif+cbor registered in Section 16.9 of this document (REQ28).

Note to RFC Editor: In the previous paragraph, please replace "TAG_NUMBER" with the CBOR tag number computed as TN(ct) in Section 4.3 of [RFC9277], where ct is the ID assigned to the CoAP Content-Format registered in Section 16.9 of this document. Then, please replace "CF_ID" with the ID assigned to that CoAP Content-Format. Finally, please delete this paragraph.

This indicates that the binary encoded scope, as conveying the actual access control information, follows the scope semantics defined for this application profile in Section 3 of this document.

5.3. Token Transferring

The exchange of Token Transfer Request and Token Transfer Response is defined in Section 3.3 of [I-D.ietf-ace-key-groupcomm]. In addition to that, the following applies.

* The Token Transfer Request MAY additionally contain the following parameters, which, if included, MUST have the corresponding values defined below (OPT2):

  - 'ecdh_info' defined in Section 5.3.1 of this document, with value the CBOR simple value "null" (0xf6) to request information about the ECDH algorithm, the ECDH algorithm parameters, the ECDH key parameters and the exact format of authentication credentials used in the groups that the Client has been authorized to join. This is relevant in case the joining node supports the pairwise mode of Group OSCORE [I-D.ietf-core-oscore-groupcomm].

  - 'kdc_dh_creds' defined in Section 5.3.2 of this document, with value the CBOR simple value "null" (0xf6) to request the Diffie-Hellman authentication credentials of the Group Manager for the groups that the Client has been authorized to join. That is, each of such authentication credentials includes a Diffie-Hellman public key of the Group Manager. This is relevant in case the joining node supports the pairwise mode of Group OSCORE [I-D.ietf-core-oscore-groupcomm].

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

* The 'kdcchallenge' parameter contains a dedicated nonce N_S generated by the Group Manager. For the N_S value, it is RECOMMENDED to use an 8-byte long random nonce. The joining node can use this nonce in order to prove the possession of its own private key, upon joining the group (see Section 6.1).

  The 'kdcchallenge' parameter MAY be omitted from the Token Transfer Response, if the 'scope' of the Access Token specifies only the role "monitor" or only the role "verifier" or only the two roles combined, for each and every of the specified groups.

* If the 'sign_info' parameter is present in the response, the following applies for each element 'sign_info_entry'.

  - 'id' MUST NOT refer to OSCORE groups that are pairwise-only groups.

- 'sign_alg' takes value from the "Value" column of the "COSE Algorithms" registry [COSE.Algorithms].

- 'sign_parameters' has the same format and value of the COSE capabilities array for the algorithm indicated in 'sign_alg', as specified for that algorithm in the "Capabilities" column of the "COSE Algorithms" registry [COSE.Algorithms] (REQ4).

- 'sign_key_parameters' has the same format and value of the COSE capabilities array for the COSE key type of the keys used with the algorithm indicated in 'sign_alg', as specified for that key type in the "Capabilities" column of the "COSE Key Types" registry [COSE.Key.Types] (REQ5).

- 'cred_fmt' takes value from the "Label" column of the "COSE Header Parameters" registry [COSE.Header.Parameters] (REQ6). Consistently with Section 2.3 of [I-D.ietf-core-oscore-groupcomm], acceptable values denote a format of authentication credential that MUST explicitly provide the public key as well as the comprehensive set of information related to the public key algorithm, including, e.g., the used elliptic curve (when applicable).

  At the time of writing this specification, acceptable formats of authentication credentials are CBOR Web Tokens (CWTs) and CWT Claims Sets (CCSs) [RFC8392], X.509 certificates [RFC7925] and C509 certificates [I-D.ietf-cose-cbor-encoded-cert]. Further formats may be available in the future, and would be acceptable to use as long as they comply with the criteria defined above.

  [As to CWTs and CCSs, the COSE Header Parameters 'kcwt' and 'kccs' are under pending registration requested by draft-ietf-lake-edhoc.]

  [As to C509 certificates, the COSE Header Parameters 'c5b' and 'c5c' are under pending registration requested by draft-ietf-cose-cbor-encoded-cert.]

This format is consistent with every signature algorithm currently considered in [RFC9053], i.e., with algorithms that have only the COSE key type as their COSE capability. Appendix B of [I-D.ietf-ace-key-groupcomm] describes how the format of each 'sign_info_entry' can be generalized for possible future registered algorithms having a different set of COSE capabilities.

* If 'ecdh_info' is included in the Token Transfer Request, the Group Manager SHOULD include the 'ecdh_info' parameter in the Token Transfer Response, as per the format defined in Section 5.3.1. Note that the field 'id' of each 'ecdh_info_entry' specifies the name, or array of group names, for which that 'ecdh_info_entry' applies to.

  As an exception, the KDC MAY omit the 'ecdh_info' parameter in the Token Transfer Response even if 'ecdh_info' is included in the Token Transfer Request, in case all the groups that the Client is authorized to join are signature-only groups.

* If 'kdc_dh_creds' is included in the Token Transfer Request and any of the groups that the Client has been authorized to join is a pairwise-only group, then the Group Manager MUST include the 'kdc_dh_creds' parameter in the Token Transfer Response, as per the format defined in Section 5.3.2. Otherwise, if 'kdc_dh_creds' is included in the Token Transfer Request, the Group Manager MAY include the 'kdc_dh_creds' parameter in the Token Transfer Response. Note that the field 'id' specifies the group name, or array of group names, for which the corresponding 'kdc_dh_creds' applies to.

Note that, other than through the above parameters as defined in Section 3.3 of [I-D.ietf-ace-key-groupcomm], the joining node may have obtained such information by alternative means. For example, information conveyed in the 'sign_info' and 'ecdh_info' parameters may have been pre-configured, or the joining node may early retrieve it, e.g., by using the approach described in [I-D.tiloca-core-oscore-discovery] to discover the OSCORE group and the link to the associated group-membership resource at the Group Manager (OPT3).

5.3.1. 'ecdh_info' Parameter

The 'ecdh_info' parameter is an OPTIONAL parameter of the request and response messages exchanged between the Client and the authz-info endpoint at the RS (see Section 5.10.1. of [RFC9200]).

This parameter allows the Client and the RS to exchange information about an ECDH algorithm as well as about the authentication credentials and public keys to accordingly use for deriving Diffie-Hellman secrets. Its exact semantics and content are application specific.

In this application profile, this parameter is used to exchange information about the ECDH algorithm as well as about the authentication credentials and public keys to be used with it, in the groups indicated by the transferred Access Token as per its 'scope' claim (see Section 3.2 of [I-D.ietf-ace-key-groupcomm]).

When used in the Token Transfer Request sent to the Group Manager, the 'ecdh_info' parameter has value the CBOR simple value "null" (0xf6). This is done to ask for information about the ECDH algorithm as well as about the authentication credentials and public keys to be used to compute static-static Diffie-Hellman shared secrets [NIST-800-56A], in the OSCORE groups that the Client has been authorized to join and that use the pairwise mode of Group OSCORE [I-D.ietf-core-oscore-groupcomm].

When used in the following Token Transfer Response from the Group Manager, the 'ecdh_info' parameter is a CBOR array of one or more elements. The number of elements is at most the number of OSCORE groups that the Client has been authorized to join.

Each element contains information about ECDH parameters as well as about authentication credentials and public keys, for one or more OSCORE groups that use the pairwise mode of Group OSCORE and that the Client has been authorized to join. Each element is formatted as follows.

* The first element 'id' is the group name of the OSCORE group or an array of group names for the OSCORE groups for which the specified information applies. In the following, each specified group name is referred to as 'gname'. The 'id' element MUST NOT refer to OSCORE groups that are signature-only groups.

* The second element 'ecdh_alg' is a CBOR integer or a CBOR text string indicating the ECDH algorithm used in the OSCORE group identified by 'gname'. Values are taken from the "Value" column of the "COSE Algorithms" registry [COSE.Algorithms].

* The third element 'ecdh_parameters' is a CBOR array indicating the parameters of the ECDH algorithm used in the OSCORE group identified by 'gname'. Its format and value are the same of the COSE capabilities array for the algorithm indicated in 'ecdh_alg', as specified for that algorithm in the "Capabilities" column of the "COSE Algorithms" registry [COSE.Algorithms].

* The fourth element 'ecdh_key_parameters' is a CBOR array indicating the parameters of the keys used with the ECDH algorithm in the OSCORE group identified by 'gname'. Its content depends on the value of 'ecdh_alg'. In particular, its format and value are

the same of the COSE capabilities array for the COSE key type of the keys used with the algorithm indicated in 'ecdh_alg', as specified for that key type in the "Capabilities" column of the "COSE Key Types" registry [COSE.Key.Types].

* The fifth element 'cred_fmt' is a CBOR integer indicating the format of authentication credentials used in the OSCORE group identified by 'gname'. It takes value from the "Label" column of the "COSE Header Parameters" registry [COSE.Header.Parameters] (REQ6). Acceptable values denote a format that MUST explicitly provide the public key as well as a comprehensive set of information related to the public key algorithm. This information includes, e.g., the used elliptic curve (when applicable). The same considerations and guidelines for the 'cred_fmt' element of 'sign_info' apply (see Section 5.3).

The CDDL notation [RFC8610] of the 'ecdh_info' parameter is given below.

```
ecdh_info = ecdh_info_req / ecdh_info_resp

ecdh_info_req = null                  ; in the Token Transfer
                                      ; Request to the
                                      ; Group Manager

ecdh_info_res = [ + ecdh_info_entry ] ; in the Token Transfer
                                      ; Response from the
                                      ; Group Manager

ecdh_info_entry =
[
  id : gname / [ + gname ],
  ecdh_alg : int / tstr,
  ecdh_parameters : [ any ],
  ecdh_key_parameters : [ any ],
  cred_fmt : int
]

gname = tstr
```

This format is consistent with every ECDH algorithm currently defined in [RFC9053], i.e., with algorithms that have only the COSE key type as their COSE capability. Appendix B of this document describes how the format of each 'ecdh_info_entry' can be generalized for possible future registered algorithms having a different set of COSE capabilities.

5.3.2. ’kdc_dh_creds’ Parameter

The ’kdc_dh_creds’ parameter is an OPTIONAL parameter of the request and response messages exchanged between the Client and the authz-info endpoint at the RS (see Section 5.10.1. of [RFC9200]).

This parameter allows the Client to request and retrieve the Diffie-Hellman authentication credentials of the RS, i.e., authentication credentials including a Diffie-Hellman public key of the RS.

In this application profile, this parameter is used to request and retrieve from the Group Manager its Diffie-Hellman authentication credentials to use, in the OSCORE groups that the Client has been authorized to join. The Group Manager has a Diffie-Hellman authentication credential in an OSCORE group if and only if the group is a pairwise-only group. In this case, the early retrieval of the Group Manager’s authentication credential is necessary in order for the joining node to prove the possession of its own private key, upon joining the group (see Section 6.1).

When used in the Token Transfer Request sent to the Group Manager, the ’kdc_dh_creds’ parameter has value the CBOR simple value "null" (0xf6). This is done to ask for the Diffie-Hellman authentication credentials that the Group Manager uses in the OSCORE groups that the Client has been authorized to join.

When used in the following Token Transfer Response from the Group Manager, the ’kdc_dh_creds’ parameter is a CBOR array of one or more elements. The number of elements is at most the number of OSCORE groups that the Client has been authorized to join.

Each element ’kdc_dh_creds_entry’ contains information about the Group Manager’s Diffie-Hellman authentication credentials, for one or more OSCORE groups that are pairwise-only groups and that the Client has been authorized to join. Each element is formatted as follows.

* The first element ’id’ is the group name of the OSCORE group or an array of group names for the OSCORE groups for which the specified information applies. In particular, ’id’ MUST refer exclusively to OSCORE groups that are pairwise-only groups.

* The second element ’cred_fmt’ is a CBOR integer indicating the format of authentication credentials used in the OSCORE group identified by ’gname’. It takes value from the "Label" column of the "COSE Header Parameters" registry [COSE.Header.Parameters] (REQ6). Acceptable values denote a format that MUST explicitly provide the public key as well as a comprehensive set of information related to the public key algorithm. This information

includes, e.g., the used elliptic curve (when applicable). The same considerations and guidelines for the 'cred_fmt' element of 'sign_info' apply (see Section 5.3).

- The third element 'cred' is a CBOR byte string, which encodes the Group Manager's Diffie-Hellman authentication credential in its original binary representation made available to other endpoints in the group. That is, the original binary representation complies with the format specified by the 'cred_fmt' element. Note that the authentication credential provides the comprehensive set of information related to its public key algorithm, i.e., the ECDH algorithm used in the OSCORE group as pairwise key agreement algorithm.

The CDDL notation [RFC8610] of the 'kdc_dh_creds' parameter is given below.

```
kdc_dh_creds = kdc_dh_creds_req / kdc_dh_creds_resp

kdc_dh_creds_req = null                     ; in the Token Transfer
                                            ; Request to the
                                            ; Group Manager

kdc_dh_creds_res = [ + kdc_dh_creds_entry ] ; in the Token Transfer
                                            ; Response from the
                                            ; Group Manager

kdc_dh_creds_entry =
[
  id : gname / [ + gname ],
  cred_fmt : int,
  cred : bstr
]

gname = tstr
```

6. Group Joining

This section describes the interactions between the joining node and the Group Manager to join an OSCORE group. The message exchange between the joining node and the Group Manager consists of the messages defined in Section 4.3.1.1 of [I-D.ietf-ace-key-groupcomm]. Note that what is defined in [I-D.ietf-ace-key-groupcomm] applies, and only additions or modifications to that specification are defined in this document.

6.1. Send the Join Request

The joining node requests to join the OSCORE group by sending a Join Request message to the related group-membership resource at the Group Manager, as per Section 4.3.1.1 of [I-D.ietf-ace-key-groupcomm]. Additionally to what is defined in Section 4.3.1 of [I-D.ietf-ace-key-groupcomm], the following applies.

* The 'scope' parameter MUST be included. Its value encodes one scope entry with the format defined in Section 3, indicating the group name and the role(s) that the joining node wants to take in the group.

  The 'scope' parameter MUST NOT specify any of the following sets of roles: ("requester", "monitor") and ("responder", "monitor"). Future specifications that define a new role for members of OSCORE groups MUST define possible sets of roles (including the new role and existing roles) that are not acceptable to specify in the 'scope' parameter of a Join Request.

* The 'get_creds' parameter is present only if the joining node wants to retrieve the authentication credentials of the group members from the Group Manager during the joining process (see Section 4). Otherwise, this parameter MUST NOT be present.

  If this parameter is present and its value is not the CBOR simple value "null" (0xf6), each element of the inner CBOR array 'role_filter' is encoded as a CBOR unsigned integer, with the same value of a permission set ("Tperm") indicating that role or combination of roles in a scope entry, as defined in Section 3.

* 'cnonce' contains a dedicated nonce N_C generated by the joining node. For the N_C value, it is RECOMMENDED to use an 8-byte long random nonce.

* The proof-of-possession (PoP) evidence included in 'client_cred_verify' is computed as defined below (REQ14). In either case, the N_S used to build the PoP input is as defined in Section 6.1.1.

  - If the group is not a pairwise-only group, the PoP evidence MUST be a signature. The joining node computes the signature by using the same private key and signature algorithm it intends to use for signing messages in the OSCORE group.

- If the group is a pairwise-only group, the PoP evidence MUST be a MAC computed as follows, by using the HKDF Algorithm HKDF SHA-256, which consists of composing the HKDF-Extract and HKDF-Expand steps [RFC5869].

  MAC = HKDF(salt, IKM, info, L)

  The input parameters of HKDF are as follows.

  o salt takes as value the empty byte string.

  o IKM is computed as a cofactor Diffie-Hellman shared secret, see Section 5.7.1.2 of [NIST-800-56A], using the ECDH algorithm used in the OSCORE group. The joining node uses its own Diffie-Hellman private key and the Diffie-Hellman public key of the Group Manager. For X25519 and X448, the procedure is described in Section 5 of [RFC7748].

  o info takes as value the PoP input.

  o L is equal to 8, i.e., the size of the MAC, in bytes.

6.1.1. Value of the N_S Challenge

The value of the N_S challenge is determined as follows.

1. If the joining node has provided the Access Token to the Group Manager by means of a Token Transfer Request to the /authz-info endpoint as in Section 5.3, then N_S takes the same value of the most recent 'kdcchallenge' parameter received by the joining node from the Group Manager. This can be either the one specified in the Token Transfer Response, or the one possibly specified in a 4.00 (Bad Request) error response to a following Join Request (see Section 6.2).

2. If the provisioning of the Access Token to the Group Manager has relied on the DTLS profile of ACE [RFC9202], and the Access Token was specified:

   * in the "psk_identity" field of the ClientKeyExchange message when using DTLS 1.2 [RFC6347]; or

   * in the "identity" field of a PskIdentity within the PreSharedKeyExtension of the ClientHello message when using DTLS 1.3 [RFC9147],

then N_S is an exporter value computed as defined in Section 7.5 of [RFC8446]. Specifically, N_S is exported from the DTLS session between the joining node and the Group Manager, using an empty 'context_value', 32 bytes as 'key_length', and the exporter label "EXPORTER-ACE-Sign-Challenge-coap-group-oscore-app" defined in Section 16.7 of this document.

It is up to applications to define how N_S is computed in further alternative settings.

Section 15.3 provides security considerations on the reusage of the N_S challenge.

6.2. Receive the Join Request

The Group Manager processes the Join Request as defined in Section 4.3.1 of [I-D.ietf-ace-key-groupcomm], with the following additions.

The Group Manager verifies the PoP evidence contained in 'client_cred_verify' as follows:

- As PoP input, the Group Manager uses the value of the 'scope' parameter from the Join Request as a CBOR byte string, concatenated with N_S encoded as a CBOR byte string, concatenated with N_C encoded as a CBOR byte string. The value of N_S is determined as described in Section 6.1.1, while N_C is the nonce provided in the 'cnonce' parameter of the Join Request.

- As public key of the joining node, the Group Manager uses either the one included in the authentication credential retrieved from the 'client_cred' parameter of the Join Request, or the one from the already stored authentication credential as acquired from previous interactions with the joining node (see Section 4).

- If the group is not a pairwise-only group, the PoP evidence is a signature. The Group Manager verifies it by using the public key of the joining node, as well as the signature algorithm used in the OSCORE group and possible corresponding parameters.

* If the group is a pairwise-only group, the PoP evidence is a MAC. The Group Manager recomputes the MAC through the same process taken by the joining node when preparing the value of the 'client_cred_verify' parameter for the Join Request (see Section 6.1), with the difference that the Group Manager uses its own Diffie-Hellman private key and the Diffie-Hellman public key of the joining node. The verification succeeds if and only if the recomputed MAC is equal to the MAC conveyed as PoP evidence in the Join Request.

The Group Manager MUST reply with a 5.03 (Service Unavailable) error response in the following cases:

* There are currently no OSCORE Sender IDs available to assign in the OSCORE group and, at the same time, the joining node is not going to join the group exclusively as monitor. The response MUST have Content-Format set to application/ace-groupcomm+cbor and is formatted as defined in Section 4.1.2 of [I-D.ietf-ace-key-groupcomm]. The value of the 'error' field MUST be set to 4 ("No available node identifiers").

* The OSCORE group that the joining node has been trying to join is currently inactive (see Section 8.1). The response MUST have Content-Format set to application/ace-groupcomm+cbor and is formatted as defined in Section 4.1.2 of [I-D.ietf-ace-key-groupcomm]. The value of the 'error' field MUST be set to 9 ("Group currently not active").

The Group Manager MUST reply with a 4.00 (Bad Request) error response in the following cases:

* The 'client_cred' parameter is present in the Join Request and its value is not an eligible authentication credential (e.g., it is not of the format accepted in the group).

* The 'client_cred' parameter is not present in the Join Request while the joining node is not going to join the group exclusively as monitor, and any of the following conditions holds:

  - The Group Manager does not store an eligible authentication credential (e.g., of the format accepted in the group) for the joining node.

  - The Group Manager stores multiple eligible authentication credentials (e.g., of the format accepted in the group) for the joining node.

- The 'scope' parameter is not present in the Join Request, or it is present and specifies any of the following sets of roles: ("requester", "monitor") and ("responder", "monitor").

- The Join Request includes the 'client_cred' parameter but does not include both the 'cnonce' and 'client_cred_verify' parameters.

In order to prevent the acceptance of Ed25519 and Ed448 public keys that cannot be successfully converted to Montgomery coordinates, and thus cannot be used for the derivation of pairwise keys (see Section 2.4.1 of [I-D.ietf-core-oscore-groupcomm]), the Group Manager MAY reply with a 4.00 (Bad Request) error response in case all the following conditions hold:

- The OSCORE group uses the pairwise mode of Group OSCORE.

- The OSCORE group uses EdDSA public keys [RFC8032].

- The authentication credential of the joining node from the 'client_cred' parameter includes a public key which:

  - Is for the elliptic curve Ed25519 and has its Y coordinate equal to -1 or 1 (mod p), with p = (2^255 - 19), see Section 4.1 of [RFC7748]; or

  - Is for the elliptic curve Ed448 and has its Y coordinate equal to -1 or 1 (mod p), with p = (2^448 - 2^224 - 1), see Section 4.2 of [RFC7748].

A 4.00 (Bad Request) error response from the Group Manager to the joining node MUST have content format application/ace-groupcomm+cbor. The response payload is a CBOR map formatted as follows:

- If the group uses (also) the group mode of Group OSCORE, the CBOR map MUST contain the 'sign_info' parameter, whose CBOR label is defined in Section 8 of [I-D.ietf-ace-key-groupcomm]. This parameter has the same format of 'sign_info_res' defined in Section 3.3.1 of [I-D.ietf-ace-key-groupcomm] and includes a single element 'sign_info_entry', pertaining to the OSCORE group that the joining node has tried to join with the Join Request.

- If the group uses (also) the pairwise mode of Group OSCORE, the CBOR map MUST contain the 'ecdh_info' parameter, whose CBOR label is defined in Section 16.3. This parameter has the same format of 'ecdh_info_res' defined in Section 5.3.1 and includes a single element 'ecdh_info_entry', pertaining to the OSCORE group that the joining node has tried to join with the Join Request.

- If the group is a pairwise-only group, the CBOR map MUST contain the ’kdc_dh_creds’ parameter, whose CBOR label is defined in Section 16.3. This parameter has the same format of ’kdc_dh_creds_res’ defined in Section 5.3.2 and includes a single element ’kdc_dh_creds_entry’, pertaining to the OSCORE group that the joining node has tried to join with the Join Request.

- The CBOR map MAY include the ’kdcchallenge’ parameter, whose CBOR label is defined in Section 8 of [I-D.ietf-ace-key-groupcomm]. If present, this parameter is a CBOR byte string, which encodes a newly generated ’kdcchallenge’ value that the Client can use when preparing a Join Request (see Section 6.1). In such a case the Group Manager MUST store the newly generated value as the ’kdcchallenge’ value associated with the joining node, replacing the currently stored value (if any).

6.2.1. Follow-up to a 4.00 (Bad Request) Error Response

When receiving a 4.00 (Bad Request) error response, the joining node MAY send a new Join Request to the Group Manager. In such a case:

- The ’cnonce’ parameter MUST include a new dedicated nonce N_C generated by the joining node.

- The ’client_cred’ parameter MUST include an authentication credential in the format indicated by the Group Manager. Also, the authentication credential as well as the included public key MUST be compatible with the signature or ECDH algorithm, and possible associated parameters.

- The ’client_cred_verify’ parameter MUST include a PoP evidence computed as described in Section 6.1, by using the private key associated with the authentication credential specified in the current ’client_cred’ parameter, with the signature or ECDH algorithm, and possible associated parameters indicated by the Group Manager. If the error response from the Group Manager includes the ’kdcchallenge’ parameter, the joining node MUST use its content as new N_S challenge to compute the PoP evidence.

6.3. Send the Join Response

If the processing of the Join Request described in Section 6.2 is successful, the Group Manager updates the group membership by registering the joining node NODENAME as a new member of the OSCORE group GROUPNAME, as described in Section 4.3.1 of [I-D.ietf-ace-key-groupcomm].

If the joining node has not taken exclusively the role of monitor, the Group Manager performs also the following actions.

* The Group Manager selects an available OSCORE Sender ID in the OSCORE group, and exclusively assigns it to the joining node. The Group Manager MUST NOT assign an OSCORE Sender ID to the joining node if this joins the group exclusively with the role of monitor, according to what is specified in the Access Token (see Section 5.2).

  Consistently with Section 3.2.1 of [I-D.ietf-core-oscore-groupcomm], the Group Manager MUST assign an OSCORE Sender ID that has not been used in the OSCORE group since the latest time when the current Gid value was assigned to the group.

  If the joining node is recognized as a current group member, e.g., through the ongoing secure communication association, the following also applies.

  - The Group Manager MUST assign a new OSCORE Sender ID different than the one currently used by the joining node in the OSCORE group.

  - The Group Manager MUST add the old, relinquished OSCORE Sender ID of the joining node to the set of stale Sender IDs associated with the current version of the group keying material for the group (see Section 7.1).

* The Group Manager stores the association between i) the authentication credential of the joining node; and ii) the Group Identifier (Gid), i.e., the OSCORE ID Context, associated with the OSCORE group together with the OSCORE Sender ID assigned to the joining node in the group. The Group Manager MUST keep this association updated over time.

Then, the Group Manager replies to the joining node, providing the updated security parameters and keying material necessary to participate in the group communication. This success Join Response is formatted as defined in Section 4.3.1 of [I-D.ietf-ace-key-groupcomm], with the following additions:

* The 'gkty' parameter identifies a key of type "Group_OSCORE_Input_Material object", defined in Section 16.4 of this document.

* The ’key’ parameter includes what the joining node needs in order to set up the Group OSCORE Security Context as per Section 2 of [I-D.ietf-core-oscore-groupcomm].

  This parameter has as value a Group_OSCORE_Input_Material object, which is defined in this document and extends the OSCORE_Input_Material object encoded in CBOR as defined in Section 3.2.1 of [RFC9203]. In particular, it contains the additional parameters ’group_senderId’, ’cred_fmt’, ’sign_enc_alg’, ’sign_alg’, ’sign_params’, ’ecdh_alg’ and ’ecdh_params’ defined in Section 16.6 of this document.

  More specifically, the ’key’ parameter is composed as follows.

  - The ’hkdf’ parameter, if present, specifies the HKDF Algorithm used in the OSCORE group. The HKDF Algorithm is specified by the HMAC Algorithm value. This parameter MAY be omitted, if the HKDF Algorithm used in the group is HKDF SHA-256. Otherwise, this parameter MUST be present.

  - The ’salt’ parameter, if present, has as value the OSCORE Master Salt used in the OSCORE group. This parameter MAY be omitted, if the Master Salt used in the group is the empty byte string. Otherwise, this parameter MUST be present.

  - The ’ms’ parameter includes the OSCORE Master Secret value used in the OSCORE group. This parameter MUST be present.

  - The ’contextId’ parameter has as value the Group Identifier (Gid), i.e., the OSCORE ID Context of the OSCORE group. This parameter MUST be present.

  - The ’group_senderId’ parameter has as value the OSCORE Sender ID assigned to the joining node by the Group Manager, as described above. This parameter MUST be present if and only if the node does not join the OSCORE group exclusively with the role of monitor, according to what is specified in the Access Token (see Section 5.2).

- The 'cred_fmt' parameter specifies the format of authentication credentials used in the OSCORE group. This parameter MUST be present and it takes value from the "Label" column of the "COSE Header Parameters" registry [COSE.Header.Parameters] (REQ6). Consistently with Section 2.3 of [I-D.ietf-core-oscore-groupcomm], acceptable values denote a format that MUST explicitly provide the public key as well as a comprehensive set of information related to the public key algorithm. This information includes, e.g., the used elliptic curve (when applicable).

  At the time of writing this specification, acceptable formats of authentication credentials are CBOR Web Tokens (CWTs) and CWT Claims Sets (CCSs) [RFC8392], X.509 certificates [RFC7925] and C509 certificates [I-D.ietf-cose-cbor-encoded-cert]. Further formats may be available in the future, and would be acceptable to use as long as they comply with the criteria defined above.

  [As to CWTs and CCSs, the COSE Header Parameters 'kcwt' and 'kccs' are under pending registration requested by draft-ietf-lake-edhoc.]

  [As to C509 certificates, the COSE Header Parameters 'c5b' and 'c5c' are under pending registration requested by draft-ietf-cose-cbor-encoded-cert.]

The 'key' parameter MUST also include the following parameters, if and only if the OSCORE group is not a pairwise-only group.

- The 'sign_enc_alg' parameter, specifying the Signature Encryption Algorithm used in the OSCORE group to encrypt messages protected with the group mode. This parameter takes values from the "Value" column of the "COSE Algorithms" registry [COSE.Algorithms].

- The 'sign_alg' parameter, specifying the Signature Algorithm used to sign messages in the OSCORE group. This parameter takes values from the "Value" column of the "COSE Algorithms" registry [COSE.Algorithms].

- The 'sign_params' parameter, specifying the parameters of the Signature Algorithm. This parameter is a CBOR array, which includes the following two elements:

  o 'sign_alg_capab': a CBOR array, with the same format and value of the COSE capabilities array for the Signature Algorithm indicated in 'sign_alg', as specified for that algorithm in the "Capabilities" column of the "COSE Algorithms" registry [COSE.Algorithms].

  o 'sign_key_type_capab': a CBOR array, with the same format and value of the COSE capabilities array for the COSE key type of the keys used with the Signature Algorithm indicated in 'sign_alg', as specified for that key type in the "Capabilities" column of the "COSE Key Types" registry [COSE.Key.Types].

The 'key' parameter MUST also include the following parameters, if and only if the OSCORE group is not a signature-only group.

- The 'alg' parameter, specifying the AEAD Algorithm used in the OSCORE group to encrypt messages protected with the pairwise mode.

- The 'ecdh_alg' parameter, specifying the Pairwise Key Agreement Algorithm used in the OSCORE group. This parameter takes values from the "Value" column of the "COSE Algorithms" registry [COSE.Algorithms].

- The 'ecdh_params' parameter, specifying the parameters of the Pairwise Key Agreement Algorithm. This parameter is a CBOR array, which includes the following two elements:

  o 'ecdh_alg_capab': a CBOR array, with the same format and value of the COSE capabilities array for the algorithm indicated in 'ecdh_alg', as specified for that algorithm in the "Capabilities" column of the "COSE Algorithms" registry [COSE.Algorithms].

  o 'ecdh_key_type_capab': a CBOR array, with the same format and value of the COSE capabilities array for the COSE key type of the keys used with the algorithm indicated in 'ecdh_alg', as specified for that key type in the "Capabilities" column of the "COSE Key Types" registry [COSE.Key.Types].

The format of ’key’ defined above is consistent with every signature algorithm and ECDH algorithm currently considered in [RFC9053], i.e., with algorithms that have only the COSE key type as their COSE capability. Appendix B of this document describes how the format of the ’key’ parameter can be generalized for possible future registered algorithms having a different set of COSE capabilities.

Furthermore, the following applies.

* The ’exp’ parameter MUST be present.

* The ’ace_groupcomm_profile’ parameter MUST be present and has value coap_group_oscore_app (PROFILE_TBD), which is defined in Section 16.5 of this document.

* The ’creds’ parameter, if present, includes the authentication credentials requested by the joining node by means of the ’get_creds’ parameter in the Join Request.

  If the joining node has asked for the authentication credentials of all the group members, i.e., ’get_creds’ had value the CBOR simple value "null" (0xf6) in the Join Request, then the Group Manager provides only the authentication credentials of the group members that are relevant to the joining node. That is, in such a case, ’creds’ includes only: i) the authentication credentials of the responders currently in the OSCORE group, in case the joining node is configured (also) as requester; and ii) the authentication credentials of the requesters currently in the OSCORE group, in case the joining node is configured (also) as responder or monitor.

* The ’peer_identifiers’ parameter includes the OSCORE Sender ID of each group member whose authentication credential is specified in the ’creds’ parameter. That is, a group member’s Sender ID is used as identifier for that group member (REQ25).

* The ’group_policies’ parameter SHOULD be present, and SHOULD include the following elements:

  - "Key Update Check Interval" defined in Section 4.3.1 of [I-D.ietf-ace-key-groupcomm], with default value 3600;

  - "Expiration Delta" defined in Section 4.3.1 of [I-D.ietf-ace-key-groupcomm], with default value 0.

* The 'kdc_cred' parameter MUST be present, specifying the Group Manager's authentication credential in its original binary representation (REQ8). The Group Manager's authentication credential MUST be in the format used in the OSCORE group. Also, the authentication credential as well as the included public key MUST be compatible with the signature or ECDH algorithm, and possible associated parameters used in the OSCORE group.

* The 'kdc_nonce' parameter MUST be present, specifying the dedicated nonce N_KDC generated by the Group Manager. For N_KDC, it is RECOMMENDED to use an 8-byte long random nonce.

* The 'kdc_cred_verify' parameter MUST be present, specifying the proof-of-possession (PoP) evidence computed by the Group Manager. The PoP evidence is computed over the nonce N_KDC, which is specified in the 'kdc_nonce' parameter and taken as PoP input. The PoP evidence is computed as defined below (REQ21).

  - If the group is not a pairwise-only group, the PoP evidence MUST be a signature. The Group Manager computes the signature by using the signature algorithm used in the OSCORE group, as well as its own private key associated with the authentication credential specified in the 'kdc_cred' parameter.

  - If the group is a pairwise-only group, the PoP evidence MUST be a MAC computed as follows, by using the HKDF Algorithm HKDF SHA-256, which consists of composing the HKDF-Extract and HKDF-Expand steps [RFC5869].

    MAC = HKDF(salt, IKM, info, L)

    The input parameters of HKDF are as follows.

    o salt takes as value the empty byte string.

    o IKM is computed as a cofactor Diffie-Hellman shared secret, see Section 5.7.1.2 of [NIST-800-56A], using the ECDH algorithm used in the OSCORE group. The Group Manager uses its own Diffie-Hellman private key and the Diffie-Hellman public key of the joining node. For X25519 and X448, the procedure is described in Section 5 of [RFC7748].

    o info takes as value the PoP input.

    o L is equal to 8, i.e., the size of the MAC, in bytes.

* The 'group_rekeying' parameter MAY be omitted, if the Group Manager uses the "Point-to-Point" group rekeying scheme registered in Section 11.12 of [I-D.ietf-ace-key-groupcomm] as rekeying scheme in the OSCORE group (OPT9). Its detailed use for this profile is defined in Section 11 of this document. In any other case, the 'group_rekeying' parameter MUST be included.

As a last action, if the Group Manager reassigns Gid values during the group's lifetime (see Section 3.2.1.1 of [I-D.ietf-core-oscore-groupcomm]), then the Group Manager MUST store the Gid specified in the 'contextId' parameter of the 'key' parameter, as the Birth Gid of the joining node in the joined group (see Section 3 of [I-D.ietf-core-oscore-groupcomm]). This applies also in case the joining node is in fact re-joining the group; in such a case, the newly determined Birth Gid overwrites the one currently stored.

6.4. Receive the Join Response

Upon receiving the Join Response, the joining node retrieves the Group Manager's authentication credential from the 'kdc_cred' parameter. The joining node MUST verify the proof-of-possession (PoP) evidence specified in the 'kdc_cred_verify' parameter of the Join Response as defined below (REQ21).

* If the group is not a pairwise-only group, the PoP evidence is a signature. The joining node verifies it by using the public key of the Group Manager from the received authentication credential, as well as the signature algorithm used in the OSCORE group and possible corresponding parameters.

* If the group is a pairwise-only group, the PoP evidence is a MAC. The joining node recomputes the MAC through the same process taken by the Group Manager when computing the value of the 'kdc_cred_verify' parameter (see Section 6.3), with the difference that the joining node uses its own Diffie-Hellman private key and the Diffie-Hellman public key of the Group Manager from the received authentication credential. The verification succeeds if and only if the recomputed MAC is equal to the MAC conveyed as PoP evidence in the Join Response.

In case of failed verification of the PoP evidence, the joining node MUST stop processing the Join Response and MAY send a new Join Request to the Group Manager (see Section 6.1).

In case of successful verification of the PoP evidence, the joining node uses the information received in the Join Response to set up the Group OSCORE Security Context, as described in Section 2 of

[I-D.ietf-core-oscore-groupcomm]. If the following parameters were not included in the 'key' parameter of the Join Response, the joining node considers the default values specified below, consistently with Section 3.2 of [RFC8613].

* Absent the 'hkdf' parameter, the joining node considers HKDF SHA-256 as HKDF Algorithm to use in the OSCORE group.

* Absent the 'salt' parameter, the joining node considers the empty byte string as Master Salt to use in the OSCORE group.

* Absent the 'group_rekeying' parameter, the joining node considers the "Point-to-Point" group rekeying scheme registered in Section 11.12 of [I-D.ietf-ace-key-groupcomm] as the rekeying scheme used in the group (OPT9). Its detailed use for this profile is defined in Section 11 of this document.

In addition, the joining node maintains an association between each authentication credential retrieved from the 'creds' parameter and the role(s) that the corresponding group member has in the OSCORE group.

From then on, the joining node can exchange group messages secured with Group OSCORE as described in [I-D.ietf-core-oscore-groupcomm]. When doing so:

* The joining node MUST NOT process an incoming request message, if protected by a group member whose authentication credential is not associated with the role "Requester".

* The joining node MUST NOT process an incoming response message, if protected by a group member whose authentication credential is not associated with the role "Responder".

* The joining node MUST NOT use the pairwise mode of Group OSCORE to process messages in the group, if the Join Response did not include the 'ecdh_alg' parameter.

If the application requires backward security, the Group Manager MUST generate updated security parameters and group keying material, and provide it to the current group members, upon the new node's joining (see Section 11). In such a case, the joining node is not able to access secure communication in the OSCORE group that occurred prior to its joining.

7. Overview of the Group Rekeying Process

In a number of cases, the Group Manager has to generate new keying material and distribute it to the group (rekeying), as also discussed in Section 3.2 of [I-D.ietf-core-oscore-groupcomm].

To this end the Group Manager MUST support the Group Rekeying Process described in Section 11 of this document, as an instance of the "Point-to-Point" rekeying scheme defined in Section 6.1 of [I-D.ietf-ace-key-groupcomm] and registered in Section 11.12 of [I-D.ietf-ace-key-groupcomm]. Future documents may define the use of alternative group rekeying schemes for this application profile, together with the corresponding rekeying message formats. The resulting group rekeying process MUST comply with the functional steps defined in Section 3.2 of [I-D.ietf-core-oscore-groupcomm].

Upon generating the new group keying material and before starting its distribution, the Group Manager MUST increment the version number of the group keying material. When rekeying a group, the Group Manager MUST preserve the current value of the OSCORE Sender ID of each member in that group.

The data distributed to a group through a rekeying MUST include:

* The new version number of the group keying material for the group.

* A new Group Identifier (Gid) for the group as introduced in [I-D.ietf-ace-key-groupcomm], used as ID Context parameter of the Group OSCORE Common Security Context of that group (see Section 2 of [I-D.ietf-core-oscore-groupcomm]).

  Note that the Gid differs from the group name also introduced in [I-D.ietf-ace-key-groupcomm], which is a plain, stable and invariant identifier, with no cryptographic relevance and meaning.

* A new value for the Master Secret parameter of the Group OSCORE Common Security Context of the group (see Section 2 of [I-D.ietf-core-oscore-groupcomm]).

* A set of stale Sender IDs, which allows each rekeyed node to purge authentication credentials and Recipient Contexts used in the group and associated with those Sender IDs. This in turn allows every group member to rely on stored authentication credentials, in order to confidently assert the group membership of other sender nodes, when receiving protected messages in the group (see Section 3.2 of [I-D.ietf-core-oscore-groupcomm]). More details on the maintenance of stale Sender IDs are provided in Section 7.1.

Also, the data distributed through a group rekeying MAY include a new value for the Master Salt parameter of the Group OSCORE Common Security Context of that group.

The Group Manager MUST rekey the group in the following cases.

* The application requires backward security - In this case, the group is rekeyed when a node joins the group as a new member. Therefore, a joining node cannot access communications in the group prior to its joining.

* One or more nodes leave the group - That is, the group is rekeyed when one or more current members spontaneously request to leave the group (see Section 9.11), or when the Group Manager forcibly evicts them from the group, e.g., due to expired or revoked authorization (see Section 10). Therefore, a leaving node cannot access communications in the group after its leaving, thus ensuring forward security in the group.

  Due to the set of stale Sender IDs distributed through the rekeying, this ensures that a node owning the latest group keying material does not store the authentication credentials of former group members (see Sections 3.2 and 12.1 of [I-D.ietf-core-oscore-groupcomm]).

When the expiration time for the group keying material approaches or has passed, the Group Manager may want to extend the secure group operation, as considered appropriate. If the Group Manager does so, the Group Manager MUST rekey the group.

The Group Manager MAY rekey the group for other reasons, e.g., according to an application-specific rekeying period or scheduling.

7.1. Stale OSCORE Sender IDs

For each OSCORE group, the Group Manager MUST maintain N > 1 sets of "stale" OSCORE Sender IDs. It is up to the application to specify the value of N, possibly on a per-group basis.

Each set is uniquely associated with one version of the group keying material, and includes the OSCORE Sender IDs that have become "stale" in the OSCORE group under that version of the group keying material.

In the following cases, the Group Manager MUST add an element to the set X associated with the current version of the group keying material.

* When a current group member obtains a new Sender ID, its old Sender ID is added to X. This happens when the Group Manager assigns a new Sender ID upon request from the group member (see Section 9.2), or in case the group member re-joins the group (see Section 6.1 and Section 6.3), thus also obtaining a new Sender ID.

* When a current group member leaves the group, its current Sender ID is added to X. This happens when a group member requests to leave the group (see Section 9.11) or is forcibly evicted from the group (see Section 10).

The value of N can change during the lifetime of the group. If the new value N' is smaller than N, the Group Manager MUST preserve the sets associated with the (up to) N' most recent versions of the group keying material.

When performing a group rekeying (see Section 11) for switching from an old version V to a new version V' = (V + 1) of the group keying material, the Group Manager MUST perform the following actions.

* Before creating the new group keying material with version V', if the number of sets of stale Sender IDs for the group is equal to N, then the Group Manager deletes the oldest set.

* The Group Manager rekeys the group. This includes also distributing the set of stale Sender IDs associated with the version V of the group keying material (see Section 7).

* After completing the group rekeying, the Group Manager creates an empty set of stale Sender IDs, as associated with the version V' of the group keying material.

8. Interface at the Group Manager

The Group Manager provides the interface defined in Section 4.1 of [I-D.ietf-ace-key-groupcomm], with the additional sub-resources defined from Section 8.1 to Section 8.3 of this document.

Furthermore, Section 8.4 provides a summary of the CoAP methods admitted to access different resources at the Group Manager, for nodes with different roles in the group or as non members (REQ11).

The GROUPNAME segment of the URI path MUST match with the group name specified in the scope entry of the Access Token scope (i.e., 'gname' in Section 3.1 of [I-D.ietf-ace-key-groupcomm]) (REQ7).

The Resource Type (rt=) Link Target Attribute value "core.osc.gm" is registered in Section 16.11 (REQ10), and can be used to describe group-membership resources and its sub-resources at a Group Manager, e.g., by using a link-format document [RFC6690].

Applications can use this common resource type to discover links to group-membership resources for joining OSCORE groups, e.g., by using the approach described in [I-D.tiloca-core-oscore-discovery].

8.1. ace-group/GROUPNAME/active

This resource implements a GET handler.

8.1.1. GET Handler

The handler expects a GET request.

In addition to what is defined in Section 4.1.2 of [I-D.ietf-ace-key-groupcomm], the handler verifies that the requesting Client is a current member of the group. If the verification fails, the KDC MUST reply with a 4.03 (Forbidden) error response. The response MUST have Content-Format set to application/ace-groupcomm+cbor and is formatted as defined in Section 4.1.2 of [I-D.ietf-ace-key-groupcomm]. The value of the 'error' field MUST be set to 0 ("Operation permitted only to group members").

If all verifications succeed, the handler replies with a 2.05 (Content) response, specifying the current status of the group, i.e., active or inactive. The payload of the response is formatted as defined in Section 9.9.

The method to set the current group status is out of the scope of this document, and is defined for the administrator interface of the Group Manager specified in [I-D.ietf-ace-oscore-gm-admin].

8.2. ace-group/GROUPNAME/verif-data

This resource implements a GET handler.

8.2.1. GET Handler

The handler expects a GET request.

In addition to what is defined in Section 4.1.2 of [I-D.ietf-ace-key-groupcomm], the Group Manager performs the following checks.

If the requesting Client is a current group member, the Group Manager MUST reply with a 4.03 (Forbidden) error response. The response MUST have Content-Format set to application/ace-groupcomm+cbor and is formatted as defined in Section 4.1.2 of [I-D.ietf-ace-key-groupcomm]. The value of the ’error’ field MUST be set to 8 ("Operation permitted only to signature verifiers").

If GROUPNAME denotes a pairwise-only group, the Group Manager MUST reply with a 4.00 (Bad Request) error response. The response MUST have Content-Format set to application/ace-groupcomm+cbor and is formatted as defined in Section 4.1.2 of [I-D.ietf-ace-key-groupcomm]. The value of the ’error’ field MUST be set to 7 ("Signatures not used in the group").

If all verifications succeed, the handler replies with a 2.05 (Content) response, specifying data that allow also an external signature verifier to verify signatures of messages protected with the group mode and sent to the group (see Sections 3.1 and 8.5 of [I-D.ietf-core-oscore-groupcomm]). The response MUST have Content-Format set to application/ace-groupcomm+cbor. The payload of the response is a CBOR map, which is formatted as defined in Section 9.6.

8.3. ace-group/GROUPNAME/stale-sids

This resource implements a FETCH handler.

8.3.1. FETCH Handler

The handler expects a FETCH request, whose payload specifies a version number of the group keying material, encoded as an unsigned CBOR integer.

In addition to what is defined in Section 4.1.2 of [I-D.ietf-ace-key-groupcomm], the handler verifies that the requesting Client is a current member of the group. If the verification fails, the Group Manager MUST reply with a 4.03 (Forbidden) error response. The response MUST have Content-Format set to application/ace-groupcomm+cbor and is formatted as defined in Section 4.1.2 of [I-D.ietf-ace-key-groupcomm]. The value of the ’error’ field MUST be set to 0 ("Operation permitted only to group members").

If all verifications succeed, the handler replies with a 2.05 (Content) response, specifying data that allow the requesting Client to delete the Recipient Contexts and authentication credentials associated with former members of the group (see Section 3.2 of [I-D.ietf-core-oscore-groupcomm]. The payload of the response is formatted as defined in Section 11.3.1.

8.4. Admitted Methods

The table in Figure 2 summarizes the CoAP methods admitted to access different resources at the Group Manager, for (non-)members of a group with group name GROUPNAME, and considering different roles. The last two rows of the table apply to a node with node name NODENAME.

Resource	Type1	Type2	Type3	Type4
ace-group/	F	F	F	F
ace-group/GROUPNAME/	G Po	G Po	Po *	Po
ace-group/GROUPNAME/active	G	G	-	-
ace-group/GROUPNAME/verif-data	-	-	G	-
ace-group/GROUPNAME/creds	G F	G F	G F	-
ace-group/GROUPNAME/kdc-cred	G	G	G	-
ace-group/GROUPNAME/stale-sids	F	F	-	-
ace-group/GROUPNAME/policies	G	G	-	-
ace-group/GROUPNAME/num	G	G	-	-
ace-group/GROUPNAME/nodes/NODENAME	G Pu D	G D	-	-
ace-group/GROUPNAME/nodes/NODENAME/pub-key	Po	-	-	-

CoAP methods: G = GET; F = FETCH; Po = POST; Pu = PUT; D = DELETE

Type1 = Member as Requester and/or Responder
Type2 = Member as Monitor
Type3 = Non-member (authorized to be signature verifier)
        (*) = cannot join the group as signature verifier
Type4 = Non-member (not authorized to be signature verifier)

Figure 2: Admitted CoAP Methods on the Group Manager Resources

8.4.1. Signature Verifiers

Just like any candidate group member, a signature verifier provides the Group Manager with an Access Token, as described in Section 5.3. However, unlike candidate group members, it does not join any OSCORE group, i.e., it does not perform the joining process defined in Section 6.

After successfully transferring an Access Token to the Group Manager, a signature verifier is allowed to perform only some operations as non-member of a group, and only for the OSCORE groups specified in the validated Access Token. These are the operations specified in Section 9.3, Section 9.5, Section 9.6 and Section 9.10.

Consistently, in case a node is not a member of the group with group name GROUPNAME and is authorized to be only signature verifier for that group, the Group Manager MUST reply with a 4.03 (Forbidden) error response if that node attempts to access any other endpoint than: /ace-group; ace-group/GROUPNAME/verif-data; /ace-group/GROUPNAME/creds; and ace-group/GROUPNAME/kdc-cred.

8.5. Operations Supported by Clients

Building on what is defined in Section 4.1.1 of [I-D.ietf-ace-key-groupcomm], and with reference to the resources at the Group Manager newly defined earlier in Section 8 of this document, it is expected that a Client minimally supports also the following set of operations and corresponding interactions with the Group Manager (REQ12).

* GET request to ace-group/GROUPNAME/active, in order to check the current status of the group.

* GET request to ace-group/GROUPNAME/verif-data, in order for a signature verifier to retrieve data required to verify signatures of messages protected with the group mode of Group OSCORE and sent to a group (see Sections 3.1 and 8.5 of [I-D.ietf-core-oscore-groupcomm]). Note that this operation is relevant to support only to signature verifiers.

* FETCH request to ace-group/GROUPNAME/stale-sids, in order to retrieve from the Group Manager the data required to delete some of the stored group members’ authentication credentials and associated Recipient Contexts (see Section 8.3.1). This data is provided as an aggregated set of stale Sender IDs, which are used as specified in Section 11.3.

9. Additional Interactions with the Group Manager

This section defines the possible interactions with the Group Manager, in addition to the group joining specified in Section 6.

9.1. Retrieve Updated Keying Material

At some point, a group member considers the Group OSCORE Security Context invalid and to be renewed. This happens, for instance, after a number of unsuccessful security processing of incoming messages from other group members, or when the Security Context expires as specified by the 'exp' parameter of the Join Response.

When this happens, the group member retrieves updated security parameters and group keying material. This can occur in the two different ways described below.

9.1.1. Get Group Keying Material

If the group member wants to retrieve only the latest group keying material, it sends a Key Distribution Request to the Group Manager.

That is, it sends a CoAP GET request to the endpoint /ace-group/GROUPNAME at the Group Manager.

The Group Manager processes the Key Distribution Request according to Section 4.3.2 of [I-D.ietf-ace-key-groupcomm]. The Key Distribution Response is formatted as defined in Section 4.3.2 of [I-D.ietf-ace-key-groupcomm], with the following additions.

* The 'key' parameter is formatted as defined in Section 6.3 of this document, with the difference that it does not include the 'group_SenderId' parameter.

* The 'exp' parameter MUST be present.

* The 'ace_groupcomm_profile' parameter MUST be present and has value coap_group_oscore_app.

Upon receiving the Key Distribution Response, the group member retrieves the updated security parameters and group keying material, and, if they differ from the current ones, uses them to set up the new Group OSCORE Security Context as described in Section 2 of [I-D.ietf-core-oscore-groupcomm].

9.1.2. Get Group Keying Material and OSCORE Sender ID

If the group member wants to retrieve the latest group keying material as well as the OSCORE Sender ID that it has in the OSCORE group, it sends a Key Distribution Request to the Group Manager.

That is, it sends a CoAP GET request to the endpoint /ace-group/GROUPNAME/nodes/NODENAME at the Group Manager.

The Group Manager processes the Key Distribution Request according to Section 4.8.1 of [I-D.ietf-ace-key-groupcomm]. The Key Distribution Response is formatted as defined in Section 4.8.1 of [I-D.ietf-ace-key-groupcomm], with the following additions.

* The ’key’ parameter is formatted as defined in Section 6.3 of this document. If the requesting group member has exclusively the role of monitor, then the ’key’ parameter does not include the ’group_SenderId’.

  Note that, in any other case, the current Sender ID of the group member is not specified as a separate parameter, but rather specified by ’group_SenderId’ within the ’key’ parameter.

* The ’exp’ parameter MUST be present.

Upon receiving the Key Distribution Response, the group member retrieves the updated security parameters, group keying material and Sender ID, and, if they differ from the current ones, uses them to set up the new Group OSCORE Security Context as described in Section 2 of [I-D.ietf-core-oscore-groupcomm].

9.2. Request to Change Individual Keying Material

As discussed in Section 2.5.2 of [I-D.ietf-core-oscore-groupcomm], a group member may at some point exhaust its Sender Sequence Numbers in the OSCORE group.

When this happens, the group member MUST send a Key Renewal Request message to the Group Manager, as per Section 4.8.2.1 of [I-D.ietf-ace-key-groupcomm]. That is, it sends a CoAP PUT request to the endpoint /ace-group/GROUPNAME/nodes/NODENAME at the Group Manager.

Upon receiving the Key Renewal Request, the Group Manager processes it as defined in Section 4.8.2 of [I-D.ietf-ace-key-groupcomm], with the following additions.

The Group Manager MUST return a 5.03 (Service Unavailable) response in case the OSCORE group identified by GROUPNAME is currently inactive (see Section 8.1). The response MUST have Content-Format set to application/ace-groupcomm+cbor and is formatted as defined in Section 4.1.2 of [I-D.ietf-ace-key-groupcomm]. The value of the 'error' field MUST be set to 9 ("Group currently not active").

Otherwise, the Group Manager performs one of the following actions.

1. If the requesting group member has exclusively the role of monitor, the Group Manager replies with a 4.03 (Forbidden) error response. The response MUST have Content-Format set to application/ace-groupcomm+cbor and is formatted as defined in Section 4.1.2 of [I-D.ietf-ace-key-groupcomm]. The value of the 'error' field MUST be set to 1 ("Request inconsistent with the current roles").

2. Otherwise, the Group Manager takes one of the following actions.

   * The Group Manager rekeys the OSCORE group. That is, the Group Manager generates new group keying material for that group (see Section 11), and replies to the group member with a group rekeying message as defined in Section 11, providing the new group keying material. Then, the Group Manager rekeys the rest of the OSCORE group, as discussed in Section 11.

     The Group Manager SHOULD perform a group rekeying only if already scheduled to occur shortly, e.g., according to an application-specific rekeying period or scheduling, or as a reaction to a recent change in the group membership. In any other case, the Group Manager SHOULD NOT rekey the OSCORE group when receiving a Key Renewal Request (OPT12).

   * The Group Manager determines and assigns a new OSCORE Sender ID for that group member, and replies with a Key Renewal Response formatted as defined in Section 4.8.2 of [I-D.ietf-ace-key-groupcomm]. The CBOR Map in the response payload includes a single parameter 'group_SenderId' defined in Section 16.3 of this document, specifying the new Sender ID of the group member encoded as a CBOR byte string.

     Consistently with Section 2.5.3.1 of [I-D.ietf-core-oscore-groupcomm], the Group Manager MUST assign a new Sender ID that has not been used in the OSCORE group since the latest time when the current Gid value was assigned to the group.

Furthermore, the Group Manager MUST add the old, relinquished Sender ID of the group member to the most recent set of stale Sender IDs for the group (see Section 7.1).

The Group Manager MUST return a 5.03 (Service Unavailable) response in case there are currently no Sender IDs available to assign in the OSCORE group. The response MUST have Content-Format set to application/ace-groupcomm+cbor and is formatted as defined in Section 4.1.2 of [I-D.ietf-ace-key-groupcomm]. The value of the 'error' field MUST be set to 4 ("No available node identifiers").

9.3. Retrieve Authentication Credentials of Group Members

A group member or a signature verifier may need to retrieve the authentication credentials of (other) group members. To this end, the group member or signature verifier sends an Authentication Credential Request message to the Group Manager, as per Sections 4.4.1.1 and 4.4.2.1 of [I-D.ietf-ace-key-groupcomm]. That is, it sends the request to the endpoint /ace-group/GROUPNAME/creds at the Group Manager.

If the Authentication Credential Request uses the method FETCH, the Authentication Credential Request is formatted as defined in Section 4.4.1 of [I-D.ietf-ace-key-groupcomm]. That is:

- Each element (if any) of the inner CBOR array 'role_filter' is formatted as in the inner CBOR array 'role_filter' of the 'get_creds' parameter of the Join Request when the parameter value is not the CBOR simple value "null" (0xf6) (see Section 6.1).

- Each element (if any) of the inner CBOR array 'id_filter' is a CBOR byte string, which encodes the OSCORE Sender ID of the group member for which the associated authentication credential is requested (REQ25).

Upon receiving the Authentication Credential Request, the Group Manager processes it as per Section 4.4.1 or Section 4.4.2 of [I-D.ietf-ace-key-groupcomm], depending on the request method being FETCH or GET, respectively. Additionally, if the Authentication Credential Request uses the method FETCH, the Group Manager silently ignores node identifiers included in the 'get_creds' parameter of the request that are not associated with any current group member (REQ26).

The success Authentication Credential Response is formatted as defined in Section 4.4.1 or Section 4.4.2 of [I-D.ietf-ace-key-groupcomm], depending on the request method being FETCH or GET, respectively.

9.4. Upload a New Authentication Credential

A group member may need to provide the Group Manager with its new authentication credential to use in the group from then on, hence replacing the current one. This can be the case, for instance, if the signature or ECDH algorithm and possible associated parameters used in the OSCORE group have been changed, and the current authentication credential is not compatible with them.

To this end, the group member sends an Authentication Credential Update Request message to the Group Manager, as per Section 4.9.1.1 of [I-D.ietf-ace-key-groupcomm], with the following addition.

- The group member computes the proof-of-possession (PoP) evidence included in 'client_cred_verify' in the same way taken when preparing a Join Request for the OSCORE group in question, as defined in Section 6.1 (REQ14).

That is, the group member sends a CoAP POST request to the endpoint /ace-group/GROUPNAME/nodes/NODENAME/cred at the Group Manager.

Upon receiving the Authentication Credential Update Request, the Group Manager processes it as per Section 4.9.1 of [I-D.ietf-ace-key-groupcomm], with the following additions.

- The N_S challenge used to build the proof-of-possession input is computed as defined in Section 6.1.1 (REQ15).

- The Group Manager verifies the PoP challenge included in 'client_cred_verify' in the same way as when processing a Join Request for the OSCORE group in question, as defined in Section 6.2 (REQ14).

- The Group Manager MUST return a 5.03 (Service Unavailable) response in case the OSCORE group identified by GROUPNAME is currently inactive (see Section 8.1). The response MUST have Content-Format set to application/ace-groupcomm+cbor and is formatted as defined in Section 4.1.2 of [I-D.ietf-ace-key-groupcomm]. The value of the 'error' field MUST be set to 9 ("Group currently not active").

* If the requesting group member has exclusively the role of monitor, the Group Manager replies with a 4.00 (Bad request) error response. The response MUST have Content-Format set to application/ace-groupcomm+cbor and is formatted as defined in Section 4.1.2 of [I-D.ietf-ace-key-groupcomm]. The value of the 'error' field MUST be set to 1 ("Request inconsistent with the current roles").

* If the request is successfully processed, the Group Manager stores the association between i) the new authentication credential of the group member; and ii) the Group Identifier (Gid), i.e., the OSCORE ID Context, associated with the OSCORE group together with the OSCORE Sender ID assigned to the group member in the group. The Group Manager MUST keep this association updated over time.

9.5. Retrieve the Group Manager's Authentication Credential

A group member or a signature verifier may need to retrieve the authentication credential of the Group Manager. To this end, the requesting Client sends a KDC Authentication Credential Request message to the Group Manager.

That is, it sends a CoAP GET request to the endpoint /ace-group/GROUPNAME/kdc-cred at the Group Manager defined in Section 4.5.1.1 of [I-D.ietf-ace-key-groupcomm], where GROUPNAME is the name of the OSCORE group.

In addition to what is defined in Section 4.5.1 of [I-D.ietf-ace-key-groupcomm], the Group Manager MUST respond with a 4.00 (Bad Request) error response, if the requesting Client is not a current group member and GROUPNAME denotes a pairwise-only group. The response MUST have Content-Format set to application/ace-groupcomm+cbor and is formatted as defined in Section 4.1.2 of [I-D.ietf-ace-key-groupcomm]. The value of the 'error' field MUST be set to 7 ("Signatures not used in the group").

The payload of the 2.05 (Content) KDC Authentication Credential Response is a CBOR map, which is formatted as defined in Section 4.5.1 of [I-D.ietf-ace-key-groupcomm]. The Group Manager specifies the parameters 'kdc_cred', 'kdc_nonce' and 'kdc_challenge' as defined for the Join Response in Section 6.3 of this document. This especially applies to the computing of the proof-of-possession (PoP) evidence included in 'kdc_cred_verify' (REQ21).

Upon receiving a 2.05 (Content) KDC Authentication Credential Response, the requesting Client retrieves the Group Manager's authentication credential from the 'kdc_cred' parameter, and proceeds as defined in Section 4.5.1.1 of [I-D.ietf-ace-key-groupcomm]. The requesting Client verifies the PoP evidence included in 'kdc_cred_verify' by means of the same method used when processing the Join Response, as defined in Section 6.3 of this document (REQ21).

Note that a signature verifier would not receive a successful response from the Group Manager, in case GROUPNAME denotes a pairwise-only group.

9.6. Retrieve Signature Verification Data

A signature verifier may need to retrieve data required to verify signatures of messages protected with the group mode and sent to a group (see Sections 3.1 and 8.5 of [I-D.ietf-core-oscore-groupcomm]). To this end, the signature verifier sends a Signature Verification Data Request message to the Group Manager.

That is, it sends a CoAP GET request to the endpoint /ace-group/GROUPNAME/verif-data at the Group Manager defined in Section 8.2 of this document, where GROUPNAME is the name of the OSCORE group.

The payload of the 2.05 (Content) Signature Verification Data Response is a CBOR map, which has the format used for the Join Response message in Section 6.3, with the following differences.

* From the Join Response message, only the parameters 'gkty', 'key', 'num', 'exp' and 'ace_groupcomm_profile' are present. The 'key' parameter includes only the following data.

  - The parameters 'hkdf', 'contextId', 'cred_fmt', 'sign_enc_alg', 'sign_alg', 'sign_params'. These parameters MUST be present.

  - The parameters 'alg' and 'ecdh_alg'. These parameters MUST NOT be present if the group is a signature-only group. Otherwise, they MUST be present.

* The parameter 'group_enc_key' is also included, with CBOR label defined in Section 16.3. This parameter specifies the Group Encryption Key of the OSCORE Group, encoded as a CBOR byte string. The Group Manager derives the Group Encryption Key from the group keying material, as per Section 2.1.6 of [I-D.ietf-core-oscore-groupcomm]. This parameter MUST be present.

In order to verify signatures in the group (see Section 8.5 of [I-D.ietf-core-oscore-groupcomm]), the signature verifier relies on: the data retrieved from the 2.05 (Content) Signature Verification Data Response; the public keys of the group members signing the messages to verify, retrieved from those members' authentication credentials that can be obtained as defined in Section 9.3; and the public key of the Group Manager, retrieved from the Group Manager's authentication credential that can be obtained as defined in Section 9.5.

Figure 3 gives an overview of the exchange described above, while Figure 4 shows an example of Signature Verification Data Request-Response.

```
Signature                                                     Group
Verifier                                                     Manager
  |                                                             |
  |              Signature Verification Data Request            |
  |------------------------------------------------------------>|
  |              GET ace-group/GROUPNAME/verif-data             |
  |                                                             |
  |<--- Signature Verification Data Response: 2.05 (Content) ---|
  |                                                             |
```

Figure 3: Message Flow of Signature Verification Data Request-Response

```
   Request:

   Header: GET (Code=0.01)
   Uri-Host: "kdc.example.com"
   Uri-Path: "ace-group"
   Uri-Path: "g1"
   Uri-Path: "verif-data"
   Payload: -

   Response:

   Header: Content (Code=2.05)
   Content-Format: "application/ace-groupcomm+cbor"
   Payload (in CBOR diagnostic notation, with GROUPCOMM_KEY_TBD
            and PROFILE_TBD being CBOR integers, while GROUP_ENC_KEY
            being a CBOR byte string):
    {
      "gkty": GROUPCOMM_KEY_TBD,
      "key": {
        "hkdf": 5,                     ; HMAC 256/256
        "contextId": h'37fc',
        "cred_fmt": 33,                ; x5chain
        "sign_enc_alg": 10,            ; AES-CCM-16-64-128
        "sign_alg": -8,                ; EdDSA
        "sign_params": [[1], [1, 6]]   ; [[OKP], [OKP, Ed25519]]
      },
      "num": 12,
      "exp": 1609459200,
      "ace_groupcomm_profile": PROFILE_TBD,
      "group_enc_key": GROUP_ENC_KEY
    }
```

Figure 4: Example of Signature Verification Data Request-Response

9.7. Retrieve the Group Policies

A group member may request the current policies used in the OSCORE group. To this end, the group member sends a Policies Request, as per Section 4.6.1.1 of [I-D.ietf-ace-key-groupcomm]. That is, it sends a CoAP GET request to the endpoint /ace-group/GROUPNAME/policies at the Group Manager, where GROUPNAME is the name of the OSCORE group.

Upon receiving the Policies Request, the Group Manager processes it as per Section 4.6.1 of [I-D.ietf-ace-key-groupcomm]. The success Policies Response is formatted as defined in Section 4.6.1 of [I-D.ietf-ace-key-groupcomm].

9.8. Retrieve the Keying Material Version

A group member may request the current version of the keying material used in the OSCORE group. To this end, the group member sends a Version Request, as per Section 4.7.1.1 of [I-D.ietf-ace-key-groupcomm]. That is, it sends a CoAP GET request to the endpoint /ace-group/GROUPNAME/num at the Group Manager, where GROUPNAME is the name of the OSCORE group.

Upon receiving the Version Request, the Group Manager processes it as per Section 4.7.1 of [I-D.ietf-ace-key-groupcomm]. The success Version Response is formatted as defined in Section 4.7.1 of [I-D.ietf-ace-key-groupcomm].

9.9. Retrieve the Group Status

A group member may request the current status of the OSCORE group, i.e., active or inactive. To this end, the group member sends a Group Status Request to the Group Manager.

That is, the group member sends a CoAP GET request to the endpoint /ace-group/GROUPNAME/active at the Group Manager defined in Section 8.1 of this document, where GROUPNAME is the name of the OSCORE group.

The payload of the 2.05 (Content) Group Status Response includes the CBOR simple value "true" (0xf5) if the group is currently active, or the CBOR simple value "false" (0xf4) otherwise. The group is considered active if it is set to allow new members to join, and if communication within the group is fine to happen.

Upon learning from a 2.05 (Content) response that the group is currently inactive, the group member SHOULD stop taking part in communications within the group, until it becomes active again.

Upon learning from a 2.05 (Content) response that the group has become active again, the group member can resume taking part in communications within the group.

Figure 5 gives an overview of the exchange described above, while Figure 6 shows an example of Group Status Request-Response.

```
Group                                                          Group
Member                                                        Manager
  |                                                              |
  |--- Group Status Request: GET ace-group/GROUPNAME/active --->|
  |                                                              |
  |<---------- Group Status Response: 2.05 (Content) -----------|
  |                                                              |
```

Figure 5: Message Flow of Group Status Request-Response

```
Request:

Header: GET (Code=0.01)
Uri-Host: "kdc.example.com"
Uri-Path: "ace-group"
Uri-Path: "g1"
Uri-Path: "active"
Payload: -

Response:

Header: Content (Code=2.05)
Payload (in CBOR diagnostic notation):
  true
```

Figure 6: Example of Group Status Request-Response

9.10. Retrieve Group Names

A node may want to retrieve from the Group Manager the group name and the URI of the group-membership resource of a group. This is relevant in the following cases.

* Before joining a group, a joining node may know only the current Group Identifier (Gid) of that group, but not the group name and the URI to the group-membership resource.

* As current group member in several groups, the node has missed a previous group rekeying in one of them (see Section 11). Hence, it retains stale keying material and fails to decrypt received messages exchanged in that group.

Such messages do not provide a direct hint to the correct group name, that the node would need in order to retrieve the latest keying material and authentication credentials from the Group Manager (see Section 9.1.1, Section 9.1.2 and Section 9.3). However, such messages may specify the current Gid of the group, as value of the 'kid_context' field of the OSCORE CoAP option (see Section 6.1 of [RFC8613] and Section 4.2 of [I-D.ietf-core-oscore-groupcomm]).

* As signature verifier, the node also refers to a group name for retrieving the required authentication credentials from the Group Manager (see Section 9.3). As discussed above, intercepted messages do not provide a direct hint to the correct group name, while they may specify the current Gid of the group, as value of the 'kid_context' field of the OSCORE CoAP option. In such a case, upon intercepting a message in the group, the node requires to correctly map the Gid currently used in the group with the invariant group name.

  Furthermore, since it is not a group member, the node does not take part to a possible group rekeying. Thus, following a group rekeying and the consequent change of Gid in a group, the node would retain the old Gid value and cannot correctly associate intercepted messages to the right group, especially if acting as signature verifier in several groups. This in turn prevents the efficient verification of signatures, and especially the retrieval of required, new authentication credentials from the Group Manager.

In either case, the node only knows the current Gid of the group, as learned from received or intercepted messages exchanged in the group. As detailed below, the node can contact the Group Manager, and request the group name and URI to the group-membership resource corresponding to that Gid. Then, it can use that information to join the group, or get the latest keying material as a current group member, or retrieve authentication credentials used in the group as a signature verifier. To this end, the node sends a Group Name and URI Retrieval Request, as per Section 4.2.1.1 of [I-D.ietf-ace-key-groupcomm].

That is, the node sends a CoAP FETCH request to the endpoint /ace-group at the Group Manager formatted as defined in Section 4.2.1 of [I-D.ietf-ace-key-groupcomm]. Each element of the CBOR array 'gid' is a CBOR byte string (REQ13), which encodes the Gid of the group for which the group name and the URI to the group-membership resource are requested.

Upon receiving the Group Name and URI Retrieval Request, the Group Manager processes it as per Section 4.2.1 of [I-D.ietf-ace-key-groupcomm]. The success Group Name and URI Retrieval Response is formatted as defined in Section 4.2.1 of [I-D.ietf-ace-key-groupcomm]. Each element of the CBOR array 'gid' is a CBOR byte string (REQ13), which encodes the Gid of the group for which the group name and the URI to the group-membership resource are provided.

For each of its groups, the Group Manager maintains an association between the group name and the URI to the group-membership resource on one hand, and only the current Gid for that group on the other hand. That is, the Group Manager does not maintain an association between the former pair and any other Gid for that group than the current, most recent one.

Figure 7 gives an overview of the exchanges described above, while Figure 8 shows an example of Group Name and URI Retrieval Request-Response.

```
                                                                 Group
 Node                                                           Manager
   |                                                                 |
   |---- Group Name and URI Retrieval Request: FETCH ace-group/ --->|
   |                                                                 |
   |<--- Group Name and URI Retrieval Response: 2.05 (Content) -----|
   |                                                                 |
```

Figure 7: Message Flow of Group Name and URI Retrieval Request-Response

```
      Request:

      Header: FETCH (Code=0.05)
      Uri-Host: "kdc.example.com"
      Uri-Path: "ace-group"
      Content-Format: "application/ace-groupcomm+cbor"
      Payload (in CBOR diagnostic notation):
        {
          "gid": [h'37fc', h'84bd']
        }

      Response:

      Header: Content (Code=2.05)
      Content-Format: "application/ace-groupcomm+cbor"
      Payload (in CBOR diagnostic notation):
        {
          "gid": [h'37fc', h'84bd'],
          "gname": ["g1", "g2"],
          "guri": ["ace-group/g1", "ace-group/g2"]
        }
```

Figure 8: Example of Group Name and URI Retrieval Request-Response

9.11. Leave the Group

A group member may request to leave the OSCORE group. To this end, the group member sends a Group Leaving Request, as per Section 4.8.3.1 of [I-D.ietf-ace-key-groupcomm]. That is, it sends a CoAP DELETE request to the endpoint /ace-group/GROUPNAME/nodes/NODENAME at the Group Manager.

Upon receiving the Group Leaving Request, the Group Manager processes it as per Section 4.8.3 of [I-D.ietf-ace-key-groupcomm]. Then, the Group Manager performs the follow-up actions defined in Section 10 of this document.

10. Removal of a Group Member

Other than after a spontaneous request to the Group Manager as described in Section 9.11, a node may be forcibly removed from the OSCORE group, e.g., due to expired or revoked authorization.

In either case, if the Group Manager reassigns Gid values during the group’s lifetime (see Section 3.2.1.1 of [I-D.ietf-core-oscore-groupcomm]), the Group Manager "forgets" the Birth Gid currently associated with the leaving node in the OSCORE group. This was stored following the Join Response sent to that node, after its latest (re-)joining of the OSCORE group (see Section 6.3).

If any of the two conditions below holds, the Group Manager MUST inform the leaving node of its eviction as follows. If both conditions hold, the Group Manager MUST inform the leaving node by using only the method corresponding to one of either conditions.

* If, upon joining the group (see Section 6.1), the leaving node specified a URI in the ’control_uri’ parameter defined in Section 4.3.1 of [I-D.ietf-ace-key-groupcomm], the Group Manager sends a DELETE request targeting the URI specified in the ’control_uri’ parameter (OPT7).

* If the leaving node has been observing the associated resource at ace-group/GROUPNAME/nodes/NODENAME, the Group Manager sends an unsolicited 4.04 (Not Found) error response to the leaving node, as specified in Section 4.3.2 of [I-D.ietf-ace-key-groupcomm].

Furthermore, the Group Manager might intend to evict all the current group members from the group at once. In such a case, if the Join Responses sent by the Group Manager to nodes joining the group (see Section 6.3) specify a URI in the ’control_group_uri’ parameter defined in Section 4.3.1 of [I-D.ietf-ace-key-groupcomm], then the Group Manager MUST additionally send a DELETE request targeting the URI specified in the ’control_group_uri’ parameter (OPT10).

If the leaving node has not exclusively the role of monitor, the Group Manager performs the following actions.

* The Group Manager frees the OSCORE Sender ID value of the leaving node. This value MUST NOT become available for possible upcoming joining nodes in the same group, until the group has been rekeyed and assigned a new Group Identifier (Gid).

* The Group Manager MUST add the relinquished Sender ID of the leaving node to the most recent set of stale Sender IDs for the group (see Section 7.1).

* The Group Manager cancels the association between, on one hand, the authentication credential of the leaving node and, on the other hand, the Gid associated with the OSCORE group together with the freed Sender ID value. The Group Manager deletes the

authentication credential of the leaving node, if that authentication credential has no remaining association with any pair (Gid, Sender ID).

Then, the Group Manager MUST generate updated security parameters and group keying material, and provide it to the remaining group members (see Section 11). As a consequence, the leaving node is not able to acquire the new security parameters and group keying material distributed after its leaving.

The same considerations from Section 5 of [I-D.ietf-ace-key-groupcomm] apply here as well, considering the Group Manager acting as KDC.

11. Group Rekeying Process

In order to rekey the OSCORE group, the Group Manager distributes a new Group Identifier (Gid), i.e., a new OSCORE ID Context; a new OSCORE Master Secret; and, optionally, a new OSCORE Master Salt for that group. When doing so, the Group Manager MUST increment the version number of the group keying material, before starting its distribution.

As per Section 3.2.1.1 of [I-D.ietf-core-oscore-groupcomm], the Group Manager MAY reassign a Gid to the same group over that group’s lifetime, e.g., once the whole space of Gid values has been used for the group in question. If the Group Manager supports reassignment of Gid values and performs it in a group, then the Group Manager additionally takes the following actions.

* Before rekeying the group, the Group Manager MUST check if the new Gid to be distributed coincides with the Birth Gid of any of the current group members (see Section 6.3).

* If any of such "elder members" is found in the group, the Group Manager MUST evict them from the group. That is, the Group Manager MUST terminate their membership and MUST rekey the group in such a way that the new keying material is not provided to those evicted elder members. This also includes adding their relinquished Sender IDs to the most recent set of stale Sender IDs for the group (see Section 7.1), before rekeying the group.

Until a further following group rekeying, the Group Manager MUST store the list of those latest-evicted elder members. If any of those nodes re-joins the group before a further following group rekeying occurs, the Group Manager MUST NOT rekey the group upon their re-joining. When one of those nodes re-joins the group, the Group Manager can rely, e.g., on the ongoing secure communication association to recognize the node as included in the stored list.

Across the rekeying execution, the Group Manager MUST preserve the same unchanged OSCORE Sender IDs for all group members intended to remain in the group. This avoids affecting the retrieval of authentication credentials from the Group Manager and the verification of group messages.

The Group Manager MUST support the "Point-to-Point" group rekeying scheme registered in Section 11.12 of [I-D.ietf-ace-key-groupcomm], as per the detailed use defined in Section 11.1 of this document. Future specifications may define how this application profile can use alternative group rekeying schemes, which MUST comply with the functional steps defined in Section 3.2 of [I-D.ietf-core-oscore-groupcomm]. The Group Manager MUST indicate the use of such an alternative group rekeying scheme to joining nodes, by means of the 'group_rekeying' parameter included in Join Response messages (see Section 6.3).

It is RECOMMENDED that the Group Manager gets confirmation of successful distribution from the group members, and admits a maximum number of individual retransmissions to non-confirming group members. Once completed the group rekeying process, the Group Manager creates a new empty set of stale Sender IDs associated with the version of the newly distributed group keying material (see Section 7.1).

In case the rekeying terminates and some group members have not received the new keying material, they will not be able to correctly process following secured messages exchanged in the group. These group members will eventually contact the Group Manager, in order to retrieve the current keying material and its version.

Some of these group members may be in multiple groups, each associated with a different Group Manager. When failing to correctly process messages secured with the new keying material, these group members may not have sufficient information to determine which exact Group Manager they should contact, in order to retrieve the current keying material they are missing.

If the Gid is formatted as described in Appendix C of [I-D.ietf-core-oscore-groupcomm], the Group Prefix can be used as a hint to determine the right Group Manager, as long as no collisions among Group Prefixes are experienced. Otherwise, a group member needs to contact the Group Manager of each group, e.g., by first requesting only the version of the current group keying material (see Section 9.8) and then possibly requesting the current keying material (see Section 9.1.1).

Furthermore, some of these group members can be in multiple groups, all of which are associated with the same Group Manager. In this case, these group members may also not have sufficient information to determine which exact group they should refer to, when contacting the right Group Manager. Hence, they need to contact a Group Manager multiple times, i.e., separately for each group they belong to and associated with that Group Manager.

Section 11.2 defines the actions performed by a group member upon receiving the new group keying material. Section 11.3 discusses how a group member can realize that it has missed one or more rekeying instances, and the actions it is accordingly required to take.

11.1. Sending Rekeying Messages

When using the "Point-to-Point" group rekeying scheme, the group rekeying messages MUST have Content-Format set to application/ace-groupcomm+cbor and have the same format used for the Join Response message in Section 6.3, with the following differences. Note that this extends the minimal content of a rekeying message as defined in Section 6 of [I-D.ietf-ace-key-groupcomm] (OPT14).

* From the Join Response, only the parameters 'gkty', 'key', 'num', 'exp', and 'ace_groupcomm_profile' are present. The 'key' parameter includes only the following data.

  - The 'ms' parameter, specifying the new OSCORE Master Secret value. This parameter MUST be present.

  - The 'contextId' parameter, specifying the new Gid to use as OSCORE ID Context value. This parameter MUST be present.

  - The 'salt' value, specifying the new OSCORE Master Salt value. This parameter MAY be present.

* The parameter 'stale_node_ids' MUST also be included, with CBOR label defined in Section 16.3. This parameter is encoded as a CBOR array, where each element is encoded as a CBOR byte string. The order of elements in the CBOR array is irrelevant. The parameter is populated as follows.

  - The Group Manager creates an empty CBOR array ARRAY.

   - The Group Manager considers the most recent set of stale Sender IDs for the group (see Section 7.1), i.e., the set X associated with the current version of the group keying material about to be relinquished.

   - For each Sender ID in X, the Group Manager encodes it as a CBOR byte string and adds the result to ARRAY.

   - The parameter 'stale_node_ids' takes ARRAY as value.

* The parameters 'creds', 'peer_roles' and 'peer_identifiers' SHOULD be present, if the group rekeying is performed due to one or multiple Clients that have requested to join the group. Following the same semantics used in the Join Response message (see Section 6.3), the three parameters specify the authentication credential, roles in the group and node identifier of each of the Clients that have requested to join the group. The Group Manager MUST NOT include a non-empty subset of these three parameters.

The Group Manager separately sends a group rekeying message formatted as defined above to each group member to be rekeyed.

Each rekeying message MUST be secured with the pairwise secure communication association between the Group Manager and the group member used during the joining process. Each rekeying message can target the 'control_uri' URI path defined in Section 4.3.1 of [I-D.ietf-ace-key-groupcomm] (OPT7), if provided by the intended recipient upon joining the group (see Section 6.1).

This distribution approach requires group members to act (also) as servers, in order to correctly handle unsolicited group rekeying messages from the Group Manager. If a group member and the Group Manager use OSCORE [RFC8613] to secure their pairwise communications, the group member MUST create a Replay Window in its own Recipient Context upon establishing the OSCORE Security Context with the Group Manager, e.g., by means of the OSCORE profile of ACE [RFC9203].

Group members and the Group Manager SHOULD additionally support alternative distribution approaches that do not require group members to act (also) as servers. A number of such approaches are defined in Section 6 of [I-D.ietf-ace-key-groupcomm]. In particular, a group member may use CoAP Observe [RFC7641] and subscribe for updates to the group-membership resource of the group, at the endpoint /ace-group/GROUPNAME/ of the Group Manager (see Section 6.1 of [I-D.ietf-ace-key-groupcomm]). Alternatively, a full-fledged Pub-Sub model can be considered [I-D.ietf-core-coap-pubsub], where the Group Manager publishes to a rekeying topic hosted at a Broker, while the group members subscribe to such topic (see Section 6.2 of [I-D.ietf-ace-key-groupcomm]).

11.2. Receiving Rekeying Messages

After having received the new group keying material, a group member proceeds as follows. Unless otherwise specified, the following is independent of the specifically used group rekeying scheme.

The group member considers the stale Sender IDs received from the Group Manager. If the "Point-to-Point" group rekeying scheme as detailed in Section 11.1 is used, the stale Sender IDs are specified by the 'stale_node_ids' parameter.

After that, as per Section 3.2 of [I-D.ietf-core-oscore-groupcomm], the group member MUST remove every authentication credential associated with a stale Sender ID from its list of group members' authentication credentials used in the group, and MUST delete each of its Recipient Contexts used in the group whose corresponding Recipient ID is a stale Sender ID.

Then, the following cases can occur, based on the version number V' of the new group keying material distributed through the rekeying process. If the "Point-to-Point" group rekeying scheme as detailed in Section 11.1 is used, this information is specified by the 'num' parameter.

* The group member has not missed any group rekeying. That is, the old keying material stored by the group member has version number V, while the received new keying material has version number V' = (V + 1). In such a case, the group member simply installs the new keying material and derives the corresponding new Security Context.

* The group member has missed one or more group rekeying instances. That is, the old keying material stored by the group member has version number V, while the received new keying material has version number V' > (V + 1). In such a case, the group member MUST proceed as defined in Section 11.3.

* The group member has received keying material not newer than the stored one. That is, the old keying material stored by the group member has version number V, while the received keying material has version number V' < (V + 1). In such a case, the group member MUST ignore the received rekeying messages and MUST NOT install the received keying material.

11.3. Missed Rekeying Instances

A group member can realize to have missed one or more rekeying instances in one of the ways discussed below. In the following, V denotes the version number of the old keying material stored by the group member, while V' denotes the version number of the latest, possibly just distributed, keying material.

a. The group member has participated to a rekeying process that has distributed new keying material with version number V' > (V + 1), as discussed in Section 11.2.

b. The group member has obtained the latest keying material from the Group Manager, as a response to a Key Distribution Request (see Section 9.1.1) or to a Join Request when re-joining the group (see Section 6.1). That is, V is different than V' specified by the 'num' parameter in the response.

c. The group member has obtained the authentication credentials of other group members, through an Authentication Credential Request-Response exchange with the Group Manager (see Section 9.3). That is, V is different than V' specified by the 'num' parameter in the response.

d. The group member has performed a Version Request-Response exchange with the Group Manager (see Section 9.8). That is, V is different than V' specified by the 'num' parameter in the response.

In either case, the group member MUST delete the stored keying material with version number V.

If case (a) or case (b) applies, the group member MUST perform the following actions.

1. The group member MUST NOT install the latest keying material yet, in case that was already obtained.

2. The group member sends a Stale Sender IDs Request to the Group Manager (see Section 11.3.1), specifying the version number V as payload of the request.

   If the Stale Sender IDs Response from the Group Manager has no payload, the group member MUST remove all the authentication credentials from its list of group members' authentication credentials used in the group, and MUST delete all its Recipient Contexts used in the group.

   Otherwise, the group member considers the stale Sender IDs specified in the Stale Sender IDs Response from the Group Manager. Then, the group member MUST remove every authentication credential associated with a stale Sender ID from its list of group members' authentication credentials used in the group, and MUST delete each of its Recipient Contexts used in the group whose corresponding Recipient ID is a stale Sender ID.

3. The group member installs the latest keying material with version number V' and derives the corresponding new Security Context.

If case (c) or case (d) applies, the group member SHOULD perform the following actions.

1. The group member sends a Stale Sender IDs Request to the Group Manager (see Section 11.3.1), specifying the version number V as payload of the request.

   If the Stale Sender IDs Response from the Group Manager has no payload, the group member MUST remove all the authentication credentials from its list of group members' authentication credentials used in the group, and MUST delete all its Recipient Contexts used in the group.

   Otherwise, the group member considers the stale Sender IDs specified in the Stale Sender IDs Response from the Group Manager. Then, the group member MUST remove every authentication credential associated with a stale Sender ID from its list of group members' authentication credentials used in the group, and MUST delete each of its Recipient Contexts used in the group whose corresponding Recipient ID is a stale Sender ID.

2. The group member obtains the latest keying material with version number V' from the Group Manager. This can happen by sending a Key Distribution Request to the Group Manager (see Section 9.1.1) and Section 9.1.2).

3. The group member installs the latest keying material with version number V' and derives the corresponding new Security Context.

If case (c) or case (d) applies, the group member can alternatively perform the following actions.

1. The group member re-joins the group (see Section 6.1). When doing so, the group member MUST re-join with the same roles it currently has in the group, and MUST request from the Group Manager the authentication credentials of all the current group members. That is, the 'get_creds' parameter of the Join Request MUST be present and MUST be set to the CBOR simple value "null" (0xf6).

2. When receiving the Join Response (see Section 6.4 and Section 6.4), the group member retrieves the set Z of authentication credentials specified in the 'creds' parameter.

   Then, the group member MUST remove every authentication credential which is not in Z from its list of group members' authentication credentials used in the group, and MUST delete each of its Recipient Contexts used in the group that does not include any of the authentication credentials in Z.

3. The group member installs the latest keying material with version number V' and derives the corresponding new Security Context.

11.3.1. Retrieve Stale Sender IDs

When realizing to have missed one or more group rekeying instances (see Section 11.3), a node needs to retrieve from the Group Manager the data required to delete some of its stored group members' authentication credentials and Recipient Contexts (see Section 8.3.1). These data is provided as an aggregated set of stale Sender IDs, which are used as specified in Section 11.3.

That is, the node sends a CoAP FETCH request to the endpoint /ace-group/GROUPNAME/stale-sids at the Group Manager defined in Section 8.3 of this document, where GROUPNAME is the name of the OSCORE group.

The payload of the Stale Sender IDs Request MUST include a CBOR unsigned integer. This encodes the version number V of the most recent group keying material stored and installed by the requesting Client, which is older than the latest, possibly just distributed, keying material with version number V'.

The handler MUST reply with a 4.00 (Bad Request) error response, if the request is not formatted correctly. Also, the handler MUST respond with a 4.00 (Bad Request) error response, if the specified version number V is greater or equal than the version number V' associated with the latest keying material in the group, i.e., in case V >= V'.

Otherwise, the handler responds with a 2.05 (Content) Stale Sender IDs Response. The payload of the response is formatted as defined below, where SKEW = (V' - V + 1).

* The Group Manager considers ITEMS as the current number of sets of stale Sender IDs for the group (see Section 7.1).

* If SKEW > ITEMS, the Stale Sender IDs Response MUST NOT have a payload.

* Otherwise, the payload of the Stale Sender IDs Response MUST include a CBOR array, where each element is encoded as a CBOR byte string. The order of elements in the CBOR array is irrelevant. The Group Manager populates the CBOR array as follows.

  - The Group Manager creates an empty CBOR array ARRAY and an empty set X.

  - The Group Manager considers the SKEW most recent sets of stale Sender IDs for the group. Note that the most recent set is the one associated with the latest version of the group keying material.

  - The Group Manager copies all the Sender IDs from the selected sets into X. When doing so, the Group Manager MUST discard duplicates. That is, the same Sender ID MUST NOT be present more than once in the final content of X.

  - For each Sender ID in X, the Group Manager encodes it as a CBOR byte string and adds the result to ARRAY.

  - Finally, ARRAY is specified as payload of the Stale Sender IDs Response. Note that ARRAY might result in the empty CBOR array.

Figure 9 gives an overview of the exchange described above, while Figure 10 shows an example of Stale Sender IDs Request-Response.

```
                                                           Group
Node                                                      Manager
  |                                                           |
  |                  Stale Sender IDs Request                 |
  |---------------------------------------------------------->|
  |            FETCH ace-group/GROUPNAME/stale-sids           |
  |                                                           |
  |<---------- Stale Sender IDs Response: 2.05 (Content) -------|
  |                                                           |
```

Figure 9: Message Flow of Stale Sender IDs Request-Response

```
Request:

Header: FETCH (Code=0.05)
Uri-Host: "kdc.example.com"
Uri-Path: "ace-group"
Uri-Path: "g1"
Uri-Path: "stale-sids"
Payload (in CBOR diagnostic notation):
  42

Response:

Header: Content (Code=2.05)
Payload (in CBOR diagnostic notation):
  [h'01', h'fc', h'12ab', h'de44', h'ff']
```

Figure 10: Example of Stale Sender IDs Request-Response

12. ACE Groupcomm Parameters

In addition to those defined in Section 8 of [I-D.ietf-ace-key-groupcomm], this application profile defines additional parameters used during the second part of the message exchange with the Group Manager, i.e., after the exchange of Token Transfer Request and Response (see Section 5.3). The table below summarizes them and specifies the CBOR key to use instead of the full descriptive name.

Note that the media type application/ace-groupcomm+cbor MUST be used when these parameters are transported in the respective message fields.

Name	CBOR Key	CBOR Type	Reference
group_senderId	TBD	bstr	[RFC-XXXX]
ecdh_info	TBD	array	[RFC-XXXX]
kdc_dh_creds	TBD	array	[RFC-XXXX]
group_enc_key	TBD	bstr	[RFC-XXXX]
stale_node_ids	TBD	array	[RFC-XXXX]

Figure 11: ACE Groupcomm Parameters

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

The Group Manager is expected to support all the parameters above. Instead, a Client is required to support the new parameters defined in this application profile as specified below (REQ29).

* 'group_senderId' MUST be supported by a Client that intends to join an OSCORE group with the role of Requester and/or Responder.

* 'ecdh_info' MUST be supported by a Client that intends to join a group which uses the pairwise mode of Group OSCORE.

* 'kdc_dh_creds' MUST be supported by a Client that intends to join a group which uses the pairwise mode of Group OSCORE and that does not plan to or cannot rely on an early retrieval of the Group Manager's Diffie-Hellman authentication credential.

* 'group_enc_key' MUST be supported by a Client that intends to join a group which uses the group mode of Group OSCORE or to be signature verifier for that group.

* 'stale_node_ids' MUST be supported.

When the conditional parameters defined in Section 8 of [I-D.ietf-ace-key-groupcomm] are used with this application profile, a Client must, should or may support them as specified below (REQ30).

* 'client_cred', 'cnonce', 'client_cred_verify'. A Client that has an own authentication credential to use in a group MUST support these parameters.

* 'kdcchallenge'. A Client that has an own authentication credential to use in a group and that provides the Access Token to the Group Manager through a Token Transfer Request (see Section 5.3) MUST support this parameter.

* 'creds_repo'. This parameter is not relevant for this application profile, since the Group Manager always acts as repository of the group members' authentication credentials.

* 'group_policies'. A Client that is interested in the specific policies used in a group, but that does not know them or cannot become aware of them before joining that group, SHOULD support this parameter.

* 'peer_roles'. A Client MUST support this parameter, since in this application profile it is relevant for Clients to know the roles of the group member associated with each authentication credential.

* 'kdc_nonce', 'kdc_cred' and 'kdc_cred_verify'. A Client MUST support these parameters, since the Group Manager's authentication credential is required to process messages protected with Group OSCORE (see Section 4.3 of [I-D.ietf-core-oscore-groupcomm]).

* 'mgt_key_material'. A Client that supports an advanced rekeying scheme possibly used in the group, such as based on one-to-many rekeying messages sent by the Group Manager (e.g., over IP multicast), MUST support this parameter.

* 'control_group_uri'. A Client that supports the hosting of local resources each associated with a group (hence acting as CoAP server) and the reception of one-to-many requests sent to those resources by the Group Manager (e.g., over IP multicast) MUST support this parameter.

13. ACE Groupcomm Error Identifiers

In addition to those defined in Section 9 of [I-D.ietf-ace-key-groupcomm], this application profile defines new values that the Group Manager can include as error identifiers, in the 'error' field of an error response with Content-Format application/ace-groupcomm+cbor.

Value	Description
7	Signatures not used in the group
8	Operation permitted only to signature verifiers
9	Group currently not active

Figure 12: ACE Groupcomm Error Identifiers

A Client supporting the 'error' parameter (see Sections 4.1.2 and 8 of [I-D.ietf-ace-key-groupcomm]) and able to understand the specified error may use that information to determine what actions to take next. If it is included in the error response and supported by the Client, the 'error_description' parameter may provide additional context. The following guidelines apply.

* In case of error 7, the Client should stop sending the request in question to the Group Manager. In this application profile, this error is relevant only for a signature verifier, in case it tries to access resources related to a pairwise-only group.

* In case of error 8, the Client should stop sending the request in question to the Group Manager.

* In case of error 9, the Client should wait for a certain (pre-configured) amount of time, before trying to re-send its request to the Group Manager.

14. Default Values for Group Configuration Parameters

This section defines the default values that the Group Manager assumes for the configuration parameters of an OSCORE group, unless differently specified when creating and configuring the group. This can be achieved as specified in [I-D.ietf-ace-oscore-gm-admin].

A possible reason for the Group Manager to consider default values different from those recommended in this section is to ensure that each of those are consistent with what the Group Manager supports, e.g., in terms of signature algorithm and format of authentication credentials used in the OSCORE group.

This ensures that the Group Manager is able to perform the operations defined in this document, e.g., to achieve proof-of-possession of a joining node’s private key (see Section 6.2), as well as to provide a joining node with its own authentication credential and the associated proof-of-possession challenge (see Section 6.3).

14.1. Common

This section always applies, as related to common configuration parameters.

* For the HKDF Algorithm ’hkdf’, the Group Manager SHOULD use HKDF SHA-256, defined as default in Section 3.2 of [RFC8613]. In the ’hkdf’ parameter, this HKDF Algorithm is specified by the HMAC Algorithm HMAC 256/256 (COSE algorithm encoding: 5).

* For the format ’cred_fmt’ used for the authentication credentials in the group, the Group Manager SHOULD use CBOR Web Token (CWT) or CWT Claims Set (CCS) [RFC8392], i.e., the COSE Header Parameter ’kcwt’ and ’kccs’, respectively.

  [These COSE Header Parameters are under pending registration requested by draft-ietf-lake-edhoc.]

* For ’max_stale_sets’, the Group Manager SHOULD consider N = 3 as the maximum number of stored sets of stale Sender IDs for the group (see Section 7.1).

14.2. Group Mode

This section applies if the group uses (also) the group mode of Group OSCORE.

* For the Signature Encryption Algorithm ’sign_enc_alg’ used to encrypt messages protected with the group mode, the Group Manager SHOULD use AES-CCM-16-64-128 (COSE algorithm encoding: 10) as default value.

The Group Manager SHOULD use the following default values for the Signature Algorithm ’sign_alg’ and related parameters ’sign_params’, consistently with the "COSE Algorithms" registry [COSE.Algorithms], the "COSE Key Types" registry [COSE.Key.Types] and the "COSE Elliptic Curves" registry [COSE.Elliptic.Curves].

* For the Signature Algorithm ’sign_alg’ used to sign messages protected with the group mode, the signature algorithm EdDSA [RFC8032].

* For the parameters 'sign_params' of the Signature Algorithm:

  - The array [[OKP], [OKP, Ed25519]], in case EdDSA is assumed or specified for 'sign_alg'. This indicates to use the COSE key type OKP and the elliptic curve Ed25519 [RFC8032].

  - The array [[EC2], [EC2, P-256]], in case ES256 [RFC6979] is specified for 'sign_alg'. This indicates to use the COSE key type EC2 and the elliptic curve P-256.

  - The array [[EC2], [EC2, P-384]], in case ES384 [RFC6979] is specified for 'sign_alg'. This indicates to use the COSE key type EC2 and the elliptic curve P-384.

  - The array [[EC2], [EC2, P-521]], in case ES512 [RFC6979] is specified for 'sign_alg'. This indicates to use the COSE key type EC2 and the elliptic curve P-521.

  - The array [[RSA], [RSA]], in case PS256, PS384 or PS512 [RFC8017] is specified for 'sign_alg'. This indicates to use the COSE key type RSA.

14.3. Pairwise Mode

This section applies if the group uses (also) the pairwise mode of Group OSCORE.

For the AEAD Algorithm 'alg' used to encrypt messages protected with the pairwise mode, the Group Manager SHOULD use the same default value defined in Section 3.2 of [RFC8613], i.e., AES-CCM-16-64-128 (COSE algorithm encoding: 10).

For the Pairwise Key Agreement Algorithm 'ecdh_alg' and related parameters 'ecdh_params', the Group Manager SHOULD use the following default values, consistently with the "COSE Algorithms" registry [COSE.Algorithms], the "COSE Key Types" registry [COSE.Key.Types] and the "COSE Elliptic Curves" registry [COSE.Elliptic.Curves].

* For the Pairwise Key Agreement Algorithm 'ecdh_alg' used to compute static-static Diffie-Hellman shared secrets, the ECDH algorithm ECDH-SS + HKDF-256 specified in Section 6.3.1 of [RFC9053].

* For the parameters 'ecdh_params' of the Pairwise Key Agreement Algorithm:

   - The array [[OKP], [OKP, X25519]], in case EdDSA is assumed or specified for 'sign_alg', or in case the group is a pairwise-only group. This indicates to use the COSE key type OKP and the elliptic curve X25519 [RFC8032].

   - The array [[EC2], [EC2, P-256]], in case ES256 [RFC6979] is specified for 'sign_alg'. This indicates to use the COSE key type EC2 and the elliptic curve P-256.

   - The array [[EC2], [EC2, P-384]], in case ES384 [RFC6979] is specified for 'sign_alg'. This indicates to use the COSE key type EC2 and the elliptic curve P-384.

   - The array [[EC2], [EC2, P-521]], in case ES512 [RFC6979] is specified for 'sign_alg'. This indicates to use the COSE key type EC2 and the elliptic curve P-521.

15. Security Considerations

Security considerations for this profile are inherited from [I-D.ietf-ace-key-groupcomm], the ACE framework for Authentication and Authorization [RFC9200], and the specific transport profile of ACE signalled by the AS, such as [RFC9202] and [RFC9203].

The following security considerations also apply for this profile.

15.1. Management of OSCORE Groups

This profile leverages the following management aspects related to OSCORE groups and discussed in the sections of [I-D.ietf-core-oscore-groupcomm] referred below.

* Management of group keying material (see Section 3.2 of [I-D.ietf-core-oscore-groupcomm]). The Group Manager is responsible for the renewal and re-distribution of the keying material in the groups of its competence (rekeying).

  The Group Manager performs a rekeying when one or more members leave the group, thus preserving forward security and ensuring that the security properties of Group OSCORE are fulfilled. According to the specific application requirements, the Group Manager can also rekey the group upon a new node's joining, in case backward security has also to be preserved.

* Provisioning and retrieval of authentication credentials (see Section 3 of [I-D.ietf-core-oscore-groupcomm]). The Group Manager acts as repository of authentication credentials of group members, and provides them upon request.

* Synchronization of sequence numbers (see Section 6.3 of [I-D.ietf-core-oscore-groupcomm]). This concerns how a responder node that has just joined an OSCORE group can synchronize with the sequence number of requesters in the same group.

Before sending the Join Response, the Group Manager MUST verify that the joining node actually owns the associated private key. To this end, the Group Manager can rely on the proof-of-possession challenge-response defined in Section 6.

Alternatively, when establishing a secure communication association with the Group Manager, the joining node can provide the Group Manager with its own authentication credential, and use the public key included thereof as asymmetric proof-of-possession key. For example, this is the case when the joining node relies on Section 3.2.2 of [RFC9202] and authenticates itself during the DTLS handshake with the Group Manager. However, this requires the authentication credential to be in the format used in the OSCORE group, and that both the authentication credential of the joining node and the included public key are compatible with the signature or ECDH algorithm, and possible associated parameters used in the OSCORE group.

A node may have joined multiple OSCORE groups under different non-synchronized Group Managers. Therefore, it can happen that those OSCORE groups have the same Group Identifier (Gid). It follows that, upon receiving a Group OSCORE message addressed to one of those groups, the node would have multiple Security Contexts matching with the Gid in the incoming message. It is up to the application to decide how to handle such collisions of Group Identifiers, e.g., by trying to process the incoming message using one Security Context at the time until the right one is found.

15.2. Size of Nonces as Proof-of-Possesion Challenge

With reference to the Join Request message in Section 6.1, the proof-of-possession (PoP) evidence included in 'client_cred_verify' is computed over an input including also N_C | N_S, where | denotes concatenation.

For the N_C challenge, it is RECOMMENDED to use an 8-byte long random nonce. Furthermore, N_C is always conveyed in the 'cnonce' parameter of the Join Request, which is always sent over the secure communication association between the joining node and the Group Manager.

As defined in Section 6.1.1, the way the N_S value is computed depends on the particular way the joining node provides the Group Manager with the Access Token, as well as on following interactions between the two.

* If the Access Token has not been provided to the Group Manager by means of a Token Transfer Request to the /authz-info endpoint as in Section 5.3, then N_S is computed as a 32-byte long challenge. For an example, see point (2) of Section 6.1.1.

* If the Access Token has been provided to the Group Manager by means of a Token Transfer Request to the /authz-info endpoint as in Section 5.3, then N_S takes the most recent value provided to the Client by the Group Manager in the 'kdcchallenge' parameter, as specified in point (1) of Section 6.1.1. This value is provided either in the Token Transfer Response (see Section 5.3), or in a 4.00 (Bad Request) error response to a following Join Request (see Section 6.2). In either case, it is RECOMMENDED to use an 8-byte long random challenge as value for N_S.

If we consider both N_C and N_S to take 8-byte long values, the following considerations hold.

* Let us consider both N_C and N_S as taking random values, and the Group Manager to never change the value of the N_S provided to a Client during the lifetime of an Access Token. Then, as per the birthday paradox, the average collision for N_S will happen after 2^32 new transferred Access Tokens, while the average collision for N_C will happen after 2^32 new Join Requests. This amounts to considerably more token provisionings than the expected new joinings to OSCORE groups under a same Group Manager, as well as to considerably more requests to join OSCORE groups from a same Client using a same Access Token under a same Group Manager.

* Section 7 of [RFC9203] as well Appendix B.2 of [RFC8613] recommend the use of 8-byte random values as well. Unlike in those cases, the values of N_C and N_S considered in this document are not used for as sensitive operations as the derivation of a Security Context, and thus do not have possible implications in the security of AEAD ciphers.

15.3. Reusage of Nonces for Proof-of-Possession Input

As long as the Group Manager preserves the same N_S value currently associated with an Access Token, i.e., the latest value provided to a Client in a 'kdcchallenge' parameter, the Client is able to successfully reuse the same proof-of-possession (PoP) input for multiple Join Requests to that Group Manager.

In particular, the Client can reuse the same N_C value for every Join Request to the Group Manager, and combine it with the same unchanged N_S value. This results in reusing the same PoP input for producing the PoP evidence to include in the 'client_cred_verify' parameter of the Join Requests.

Unless the Group Manager maintains a list of N_C values already used by that Client since the latest update to the N_S value associated with the Access Token, the Group Manager can be forced to falsely believe that the Client possesses its own private key at that point in time, upon verifying the PoP evidence in the 'client_cred_verify' parameter.

16. 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.

16.1. OAuth Parameters

IANA is asked to register the following entries to the "OAuth Parameters" registry, as per the procedure specified in Section 11.2 of [RFC6749].

* Parameter name: ecdh_info

* Parameter usage location: client-rs request, rs-client response

* Change Controller: IESG

* Specification Document(s): [RFC-XXXX]

* Parameter name: kdc_dh_creds

* Parameter usage location: client-rs request, rs-client response

* Change Controller: IESG

* Specification Document(s): [RFC-XXXX]

16.2. OAuth Parameters CBOR Mappings

IANA is asked to register the following entries to the "OAuth Parameters CBOR Mappings" registry, as per the procedure specified in Section 8.10 of [RFC9200].

- Name: ecdh_info
- CBOR Key: TBD (range -256 to 255)
- Value Type: Simple value "null" / Array
- Reference: [RFC-XXXX]

- Name: kdc_dh_creds
- CBOR Key: TBD (range -256 to 255)
- Value Type: Simple value "null" / Array
- Reference: [RFC-XXXX]

16.3. ACE Groupcomm Parameters

IANA is asked to register the following entries to the "ACE Groupcomm Parameters" registry defined in Section 11.6 of [I-D.ietf-ace-key-groupcomm].

- Name: group_senderId
- CBOR Key: TBD
- CBOR Type: Byte string
- Reference: [RFC-XXXX] (Section 9.2)

- Name: ecdh_info
- CBOR Key: TBD
- CBOR Type: Array
- Reference: [RFC-XXXX] (Section 6.2)

- Name: kdc_dh_creds
- CBOR Key: TBD
- CBOR Type: Array
- Reference: [RFC-XXXX] (Section 6.2)

* Name: group_enc_key

* CBOR Key: TBD

* CBOR Type: Byte string

* Reference: [RFC-XXXX] (Section 8.2.1)

* Name: stale_node_ids

* CBOR Key: TBD

* CBOR Type: Array

* Reference: [RFC-XXXX] (Section 11)

16.4. ACE Groupcomm Key Types

IANA is asked to register the following entry to the "ACE Groupcomm Key Types" registry defined in Section 11.7 of [I-D.ietf-ace-key-groupcomm].

* Name: Group_OSCORE_Input_Material object

* Key Type Value: GROUPCOMM_KEY_TBD

* Profile: "coap_group_oscore_app", defined in Section 16.5 of this document.

* Description: A Group_OSCORE_Input_Material object encoded as described in Section 6.3 of this document.

* Reference: [RFC-XXXX] (Section 6.3)

16.5. ACE Groupcomm Profiles

IANA is asked to register the following entry to the "ACE Groupcomm Profiles" registry defined in Section 11.8 of [I-D.ietf-ace-key-groupcomm].

* Name: coap_group_oscore_app

* Description: Application profile to provision keying material for participating in group communication protected with Group OSCORE as per [I-D.ietf-core-oscore-groupcomm].

* CBOR Value: PROFILE_TBD

- Reference: [RFC-XXXX] (Section 6.3)

16.6. OSCORE Security Context Parameters

IANA is asked to register the following entries in the "OSCORE Security Context Parameters" registry defined in Section 9.4 of [RFC9203].

- Name: group_SenderId
- CBOR Label: TBD
- CBOR Type: Byte string
- Registry: -
- Description: OSCORE Sender ID assigned to a member of an OSCORE group
- Reference: [RFC-XXXX] (Section 6.3)

- Name: cred_fmt
- CBOR Label: TBD
- CBOR Type: Integer
- Registry: COSE Header Parameters
- Description: Format of authentication credentials to be used in the OSCORE group
- Reference: [RFC-XXXX] (Section 6.3)

- Name: sign_enc_alg
- CBOR Label: TBD
- CBOR Type: Text string / Integer
- Registry: COSE Algorithms
- Description: OSCORE Signature Encryption Algorithm Value
- Reference: [RFC-XXXX] (Section 6.3)

* Name: sign_alg
* CBOR Label: TBD
* CBOR Type: Text string / Integer
* Registry: COSE Algorithms
* Description: OSCORE Signature Algorithm Value
* Reference: [RFC-XXXX] (Section 6.3)

* Name: sign_params
* CBOR Label: TBD
* CBOR Type: Array
* Registry: COSE Algorithms, COSE Key Types, COSE Elliptic Curves
* Description: OSCORE Signature Algorithm Parameters
* Reference: [RFC-XXXX] (Section 6.3)

* Name: ecdh_alg
* CBOR Label: TBD
* CBOR Type: Text string / Integer
* Registry: COSE Algorithms
* Description: OSCORE Pairwise Key Agreement Algorithm Value
* Reference: [RFC-XXXX] (Section 6.3)

* Name: ecdh_params
* CBOR Label: TBD
* CBOR Type: Array
* Registry: COSE Algorithms, COSE Key Types, COSE Elliptic Curves
* Description: OSCORE Pairwise Key Agreement Algorithm Parameters

* Reference: [RFC-XXXX] (Section 6.3)

16.7. TLS Exporter Labels

IANA is asked to register the following entry to the "TLS Exporter Labels" registry defined in Section 6 of [RFC5705] and updated in Section 12 of [RFC8447].

* Value: EXPORTER-ACE-Sign-Challenge-coap-group-oscore-app

* DTLS-OK: Y

* Recommended: N

* Reference: [RFC-XXXX] (Section 6.1.1)

16.8. AIF Media-Type Sub-Parameters

For the media-types application/aif+cbor and application/aif+json defined in Section 5.1 of [RFC9237], IANA is requested to register the following entries for the two media-type parameters Toid and Tperm, in the respective sub-registry defined in Section 5.2 of [RFC9237] within the "MIME Media Type Sub-Parameter" registry group.

* Parameter: Toid

* Name: oscore-gname

* Description/Specification: OSCORE group name

* Reference: [RFC-XXXX]

* Parameter: Tperm

* Name: oscore-gperm

* Description/Specification: permissions pertaining OSCORE groups

* Reference: [RFC-XXXX]

16.9. CoAP Content-Format

IANA is asked to register the following entries to the "CoAP Content-Formats" registry within the "Constrained RESTful Environments (CoRE) Parameters" registry group.

- Media Type: application/aif+cbor;Toid="oscore-gname",Tperm="oscore-gperm"

- Encoding: -

- ID: 292 (suggested)

- Reference: [RFC-XXXX]

- Media Type: application/aif+json;Toid="oscore-gname",Tperm="oscore-gperm"

- Encoding: -

- ID: 293 (suggested)

- Reference: [RFC-XXXX]

16.10. Group OSCORE Roles

This document establishes the IANA "Group OSCORE Roles" registry. The registry has been created to use the "Expert Review" registration procedure [RFC8126]. Expert review guidelines are provided in Section 16.13.

This registry includes the possible roles that nodes can take in an OSCORE group, each in combination with a numeric identifier. These numeric identifiers are used to express authorization information about joining OSCORE groups, as specified in Section 3 of [RFC-XXXX].

The columns of this registry are:

- Name: A value that can be used in documents for easier comprehension, to identify a possible role that nodes can take in an OSCORE group.

- Value: The numeric identifier for this role. Integer values greater than 65535 are marked as "Private Use", all other values use the registration policy "Expert Review" [RFC8126].

- Description: This field contains a brief description of the role.

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

This registry will be initially populated by the values in Figure 1.

The Reference column for all of these entries will be [RFC-XXXX].

16.11. CoRE Resource Type

IANA is asked to register the following entry in the "Resource Type (rt=) Link Target Attribute Values" registry within the "Constrained Restful Environments (CoRE) Parameters" registry group.

- Value: "core.osc.gm"

- Description: Group-membership resource of an OSCORE Group Manager.

- Reference: [RFC-XXXX]

Client applications can use this resource type to discover a group membership resource at an OSCORE Group Manager, where to send a request for joining the corresponding OSCORE group.

16.12. ACE Groupcomm Errors

IANA is asked to register the following entries in the "ACE Groupcomm Errors" registry defined in Section 11.11 of [I-D.ietf-ace-key-groupcomm].

- Value: 7

- Description: Signatures not used in the group.

- Reference: [RFC-XXXX]

- Value: 8

- Description: Operation permitted only to signature verifiers.

- Reference: [RFC-XXXX]

- Value: 9

- Description: Group currently not active.

- Reference: [RFC-XXXX]

16.13. Expert Review Instructions

The IANA registry established in this document is 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:

* Clarity and correctness of registrations. Experts are expected to check the clarity of purpose and use of the requested entries. Experts should inspect the entry for the considered role, to verify the correctness of its description against the role as intended in the specification that defined it. Experts should consider requesting an opinion on the correctness of registered parameters from the Authentication and Authorization for Constrained Environments (ACE) Working Group and the Constrained RESTful Environments (CoRE) Working Group.

  Entries that do not meet these objectives of clarity and completeness should not be registered.

* Duplicated registration and 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.

* Experts should take into account the expected usage of roles when approving point assignments. Given a 'Value' V as code point, the length of the encoding of (2^(V+1) - 1) should be weighed against the usage of the entry, considering the resources and capabilities of devices it will be used on. Additionally, given a 'Value' V as code point, the length of the encoding of (2^(V+1) - 1) should be weighed against how many code points resulting in that encoding length are left, and the resources and capabilities of devices it will be used on.

* Specifications are recommended. When specifications are not provided, the description provided needs to have sufficient information to verify the points above.

17. References

17.1. Normative References

[COSE.Algorithms]
IANA, "COSE Algorithms", <https://www.iana.org/assignments/cose/cose.xhtml#algorithms>.

[COSE.Elliptic.Curves]
IANA, "COSE Elliptic Curves", <https://www.iana.org/assignments/cose/cose.xhtml#elliptic-curves>.

[COSE.Header.Parameters]
IANA, "COSE Header Parameters", <https://www.iana.org/assignments/cose/cose.xhtml#header-parameters>.

[COSE.Key.Types]
IANA, "COSE Key Types", <https://www.iana.org/assignments/cose/cose.xhtml#key-type>.

[I-D.ietf-ace-key-groupcomm]
Palombini, F. and M. Tiloca, "Key Provisioning for Group Communication using ACE", Work in Progress, Internet-Draft, draft-ietf-ace-key-groupcomm-16, 5 September 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-ace-key-groupcomm-16>.

[I-D.ietf-core-oscore-groupcomm]
Tiloca, M., Selander, G., Palombini, F., Mattsson, J. P., and J. Park, "Group OSCORE - Secure Group Communication for CoAP", Work in Progress, Internet-Draft, draft-ietf-core-oscore-groupcomm-17, 20 December 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-core-oscore-groupcomm-17>.

[NIST-800-56A]
Barker, E., Chen, L., Roginsky, A., Vassilev, A., and R. Davis, "Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography - NIST Special Publication 800-56A, Revision 3", April 2018, <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar3.pdf>.

[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>.

[RFC5705] Rescorla, E., "Keying Material Exporters for Transport Layer Security (TLS)", RFC 5705, DOI 10.17487/RFC5705, March 2010, <https://www.rfc-editor.org/info/rfc5705>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/info/rfc6347>.

[RFC6979] Pornin, T., "Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA)", RFC 6979, DOI 10.17487/RFC6979, August 2013, <https://www.rfc-editor.org/info/rfc6979>.

[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>.

[RFC7748] Langley, A., Hamburg, M., and S. Turner, "Elliptic Curves for Security", RFC 7748, DOI 10.17487/RFC7748, January 2016, <https://www.rfc-editor.org/info/rfc7748>.

[RFC8017] Moriarty, K., Ed., Kaliski, B., Jonsson, J., and A. Rusch, "PKCS #1: RSA Cryptography Specifications Version 2.2", RFC 8017, DOI 10.17487/RFC8017, November 2016, <https://www.rfc-editor.org/info/rfc8017>.

[RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, DOI 10.17487/RFC8032, January 2017, <https://www.rfc-editor.org/info/rfc8032>.

[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>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8447] Salowey, J. and S. Turner, "IANA Registry Updates for TLS and DTLS", RFC 8447, DOI 10.17487/RFC8447, August 2018, <https://www.rfc-editor.org/info/rfc8447>.

[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>.

[RFC8613] Selander, G., Mattsson, J., Palombini, F., and L. Seitz, "Object Security for Constrained RESTful Environments (OSCORE)", RFC 8613, DOI 10.17487/RFC8613, July 2019, <https://www.rfc-editor.org/info/rfc8613>.

[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>.

[RFC9052] Schaad, J., "CBOR Object Signing and Encryption (COSE): Structures and Process", STD 96, RFC 9052, DOI 10.17487/RFC9052, August 2022, <https://www.rfc-editor.org/info/rfc9052>.

[RFC9053] Schaad, J., "CBOR Object Signing and Encryption (COSE): Initial Algorithms", RFC 9053, DOI 10.17487/RFC9053, August 2022, <https://www.rfc-editor.org/info/rfc9053>.

[RFC9147] Rescorla, E., Tschofenig, H., and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", RFC 9147, DOI 10.17487/RFC9147, April 2022, <https://www.rfc-editor.org/info/rfc9147>.

[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>.

[RFC9202] 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)", RFC 9202, DOI 10.17487/RFC9202, August 2022, <https://www.rfc-editor.org/info/rfc9202>.

[RFC9203] Palombini, F., Seitz, L., Selander, G., and M. Gunnarsson, "The Object Security for Constrained RESTful Environments (OSCORE) Profile of the Authentication and Authorization for Constrained Environments (ACE) Framework", RFC 9203, DOI 10.17487/RFC9203, August 2022, <https://www.rfc-editor.org/info/rfc9203>.

[RFC9237] Bormann, C., "An Authorization Information Format (AIF) for Authentication and Authorization for Constrained Environments (ACE)", RFC 9237, DOI 10.17487/RFC9237, August 2022, <https://www.rfc-editor.org/info/rfc9237>.

[RFC9277] Richardson, M. and C. Bormann, "On Stable Storage for Items in Concise Binary Object Representation (CBOR)", RFC 9277, DOI 10.17487/RFC9277, August 2022, <https://www.rfc-editor.org/info/rfc9277>.

17.2. Informative References

[I-D.ietf-ace-oscore-gm-admin]
Tiloca, M., Höglund, R., Van der Stok, P., and F. Palombini, "Admin Interface for the OSCORE Group Manager", Work in Progress, Internet-Draft, draft-ietf-ace-oscore-gm-admin-07, 24 October 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-ace-oscore-gm-admin-07>.

[I-D.ietf-core-coap-pubsub]
Koster, M., Keränen, A., and J. Jimenez, "Publish-Subscribe Broker for the Constrained Application Protocol (CoAP)", Work in Progress, Internet-Draft, draft-ietf-core-coap-pubsub-11, 7 November 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-core-coap-pubsub-11>.

[I-D.ietf-core-groupcomm-bis]
Dijk, E., Wang, C., and M. Tiloca, "Group Communication for the Constrained Application Protocol (CoAP)", Work in Progress, Internet-Draft, draft-ietf-core-groupcomm-bis-08, 11 January 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-core-groupcomm-bis-08>.

[I-D.ietf-cose-cbor-encoded-cert]
Mattsson, J. P., Selander, G., Raza, S., Höglund, J., and M. Furuhed, "CBOR Encoded X.509 Certificates (C509 Certificates)", Work in Progress, Internet-Draft, draft-ietf-cose-cbor-encoded-cert-05, 10 January 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-cose-cbor-encoded-cert-05>.

[I-D.tiloca-core-oscore-discovery]
Tiloca, M., Amsüss, C., and P. Van der Stok, "Discovery of OSCORE Groups with the CoRE Resource Directory", Work in Progress, Internet-Draft, draft-tiloca-core-oscore-discovery-12, 5 September 2022, <https://datatracker.ietf.org/doc/html/draft-tiloca-core-oscore-discovery-12>.

[RFC5869] Krawczyk, H. and P. Eronen, "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)", RFC 5869, DOI 10.17487/RFC5869, May 2010, <https://www.rfc-editor.org/info/rfc5869>.

[RFC6690] Shelby, Z., "Constrained RESTful Environments (CoRE) Link Format", RFC 6690, DOI 10.17487/RFC6690, August 2012, <https://www.rfc-editor.org/info/rfc6690>.

[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>.

[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>.

[RFC7925] Tschofenig, H., Ed. and T. Fossati, "Transport Layer Security (TLS) / Datagram Transport Layer Security (DTLS) Profiles for the Internet of Things", RFC 7925, DOI 10.17487/RFC7925, July 2016, <https://www.rfc-editor.org/info/rfc7925>.

[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>.

Appendix A. Profile Requirements

This section lists how this application profile of ACE addresses the requirements defined in Appendix A of [I-D.ietf-ace-key-groupcomm].

A.1. Mandatory-to-Address Requirements

* REQ1 - Specify the format and encoding of 'scope'. This includes defining the set of possible roles and their identifiers, as well as the corresponding encoding to use in the scope entries according to the used scope format: see Section 3 and Section 5.1.

* REQ2 - If the AIF format of 'scope' is used, register its specific instance of "Toid" and "Tperm" as Media Type parameters and a corresponding Content-Format, as per the guidelines in [RFC9237]: see Section 16.8 and Section 16.9.

* REQ3 - if used, specify the acceptable values for 'sign_alg': values from the "Value" column of the "COSE Algorithms" registry [COSE.Algorithms].

* REQ4 - If used, specify the acceptable values for 'sign_parameters': format and values from the COSE algorithm capabilities as specified in the "COSE Algorithms" registry [COSE.Algorithms].

* REQ5 - If used, specify the acceptable values for 'sign_key_parameters': format and values from the COSE key type capabilities as specified in the "COSE Key Types" registry [COSE.Key.Types].

* REQ6 - Specify the acceptable formats for authentication credentials and, if used, the acceptable values for 'cred_fmt': acceptable formats explicitly provide the public key as well as the comprehensive set of information related to the public key algorithm (see Section 5.3 and Section 6.3). Consistent acceptable values for 'cred_fmt' are taken from the "Label" column of the "COSE Header Parameters" registry [COSE.Header.Parameters].

* REQ7 - If the value of the GROUPNAME URI path and the group name in the Access Token scope (gname in Section 3.1 of [I-D.ietf-ace-key-groupcomm]) are not required to coincide, specify the mechanism to map the GROUPNAME value in the URI to the group name: not applicable, since a perfect matching is required.

* REQ8 - Define whether the KDC has an authentication credential and if this has to be provided through the 'kdc_cred' parameter, see Section 4.1 of [I-D.ietf-ace-key-groupcomm]: yes, as required by the Group OSCORE protocol [I-D.ietf-core-oscore-groupcomm], see Section 6.3 of this document.

* REQ9 - Specify if any part of the KDC interface as defined in Section 4.1 of [I-D.ietf-ace-key-groupcomm] is not supported by the KDC: not applicable.

* REQ10 - Register a Resource Type for the root url-path, which is used to discover the correct url to access at the KDC (see Section 4.1 of [I-D.ietf-ace-key-groupcomm]): the Resource Type (rt=) Link Target Attribute value "core.osc.gm" is registered in Section 16.11.

* REQ11 - Define what specific actions (e.g., CoAP methods) are allowed on each resource provided by the KDC interface, depending on whether the Client is a current group member; the roles that a Client is authorized to take as per the obtained access token; and the roles that the Client has as current group member: see Section 8.4.

* REQ12 - Categorize possible newly defined operations for Clients into primary operations expected to be minimally supported and secondary operations, and provide accompanying considerations: see Section 8.5.

* REQ13 - Specify the encoding of group identifier (see Section 4.2.1 of [I-D.ietf-ace-key-groupcomm]): CBOR byte string (see Section 9.10).

* REQ14 - Specify the approaches used to compute and verify the PoP evidence to include in 'client_cred_verify', and which of those approaches is used in which case: see Section 6.1 and Section 6.2.

* REQ15 - Specify how the nonce N_S is generated, if the token is not provided to the KDC through the Token Transfer Request to the authz-info endpoint (e.g., if it is used directly to validate TLS instead): see Section 6.1.1.

* REQ16 - Define the initial value of the 'num' parameter: the initial value MUST be set to 0 when creating the OSCORE group, e.g., as in [I-D.ietf-ace-oscore-gm-admin].

* REQ17 - Specify the format of the 'key' parameter: see Section 6.3.

* REQ18 - Specify acceptable values of the 'gkty' parameter: Group_OSCORE_Input_Material object (see Section 6.3).

* REQ19 - Specify and register the application profile identifier: coap_group_oscore_app (see Section 16.5).

* REQ20 - If used, specify the format and content of 'group_policies' and its entries: see Section 6.3.

* REQ21 - Specify the approaches used to compute and verify the PoP evidence to include in 'kdc_cred_verify', and which of those approaches is used in which case: see Section 6.3, Section 6.4 and Section 9.5.

* REQ22 - Specify the communication protocol that the members of the group must use: CoAP [RFC7252], also for group communication [I-D.ietf-core-groupcomm-bis].

* REQ23 - Specify the security protocols that the group members must use to protect their communication: Group OSCORE [I-D.ietf-core-oscore-groupcomm].

* REQ24 - Specify how the communication is secured between the Client and KDC: by means of any transport profile of ACE [RFC9200] between Client and Group Manager that complies with the requirements in Appendix C of [RFC9200].

* REQ25 - Specify the format of the identifiers of group members: the Sender ID used in the OSCORE group (see Section 6.3 and Section 9.3).

* REQ26 - Specify policies at the KDC to handle member ids that are not included in 'get_creds': see Section 9.3.

* REQ27 - Specify the format of newly-generated individual keying material for group members, or of the information to derive it, and corresponding CBOR label: see Section 9.2.

* REQ28 - Specify which CBOR tag is used for identifying the semantics of binary scopes, or register a new CBOR tag if a suitable one does not exist already: see Section 5.2.

* REQ29 - Categorize newly defined parameters according to the same criteria of Section 8 of [I-D.ietf-ace-key-groupcomm]: see Section 12.

* REQ30 - Define whether Clients must, should or may support the conditional parameters defined in Section 8 of [I-D.ietf-ace-key-groupcomm], and under which circumstances: see Section 12.

A.2. Optional-to-Address Requirements

* OPT1 (Optional) - If the textual format of 'scope' is used, specify CBOR values to use for abbreviating the role identifiers in the group: not applicable.

* OPT2 (Optional) - Specify additional parameters used in the exchange of Token Transfer Request and Response:

  - 'ecdh_info', to negotiate the ECDH algorithm, ECDH algorithm parameters, ECDH key parameters and exact format of authentication credentials used in the group, in case the joining node supports the pairwise mode of Group OSCORE (see Section 5.3).

  - 'kdc_dh_creds', to ask for and retrieve the Group Manager's Diffie-Hellman authentication credentials, in case the joining node supports the pairwise mode of Group OSCORE and the Access Token authorizes to join parwise-only groups (see Section 5.3).

* OPT3 (Optional) - Specify the negotiation of parameter values for signature algorithm and signature keys, if 'sign_info' is not used: possible early discovery by using the approach based on the CoRE Resource Directory described in [I-D.tiloca-core-oscore-discovery].

* OPT4 (Optional) - Specify possible or required payload formats for specific error cases: send a 4.00 (Bad Request) error response to a Join Request (see Section 6.2).

* OPT5 (Optional) - Specify additional identifiers of error types, as values of the 'error' field in an error response from the KDC: see Section 16.12.

* OPT6 (Optional) - Specify the encoding of 'creds_repo' if the default is not used: no.

* OPT7 (Optional) - Specify the functionalities implemented at the 'control_uri' resource hosted at the Client, including message exchange encoding and other details (see Section 4.3.1 of [I-D.ietf-ace-key-groupcomm]): see Section 10 for the eviction of a group member; see Section 11 for the group rekeying process.

* OPT8 (Optional) - Specify the behavior of the handler in case of failure to retrieve an authentication credential for the specific node: send a 4.00 (Bad Request) error response to a Join Request (see Section 6.2).

* OPT9 (Optional) - Define a default group rekeying scheme, to refer to in case the 'rekeying_scheme' parameter is not included in the Join Response (see Section 4.3.1.1 of [I-D.ietf-ace-key-groupcomm]): the "Point-to-Point" rekeying scheme registered in Section 11.12 of [I-D.ietf-ace-key-groupcomm], whose detailed use for this profile is defined in Section 11 of this document.

* OPT10 (Optional) - Specify the functionalities implemented at the 'control_group_uri' resource hosted at the Client, including message exchange encoding and other details (see Section 4.3.1 of [I-D.ietf-ace-key-groupcomm]): see Section 10 for the eviction of multiple group members.

* OPT11 (Optional) - Specify policies that instruct Clients to retain unsuccessfully decrypted messages and for how long, so that they can be decrypted after getting updated keying material: no.

* OPT12 (Optional) - Specify for the KDC to perform group rekeying (together or instead of renewing individual keying material) when receiving a Key Renewal Request: the Group Manager SHOULD NOT perform a group rekeying, unless already scheduled to occur shortly (see Section 9.2).

* OPT13 (Optional) - Specify how the identifier of a group members's authentication credential is included in requests sent to other group members: no.

* OPT14 (Optional) - Specify additional information to include in rekeying messages for the "Point-to-Point" group rekeying scheme (see Section 6.1 of [I-D.ietf-ace-key-groupcomm]): see Section 11.1.

* OPT15 (Optional) - Specify if Clients must or should support any of the parameters defined as optional in Section 8 of [I-D.ietf-ace-key-groupcomm]: no.

Appendix B. Extensibility for Future COSE Algorithms

As defined in Section 8.1 of [RFC9053], future algorithms can be registered in the "COSE Algorithms" registry [COSE.Algorithms] as specifying none or multiple COSE capabilities.

To enable the seamless use of such future registered algorithms, this section defines a general, agile format for:

* Each 'ecdh_info_entry' of the 'ecdh_info' parameter in the Token Transfer Response (see Section 5.3 and Section 5.3.1);

- The ’sign_params’ and ’ecdh_params’ parameters within the ’key’ parameter (see Section 6.3), as part of the response payloads used in Section 6.3, Section 9.1.1, Section 9.1.2 and Section 11.

Appendix B of [I-D.ietf-ace-key-groupcomm] describes the analogous general format for ’sign_info_entry’ of the ’sign_info’ parameter in the Token Transfer Response (see Section 5.3 of this document).

If any of the currently registered COSE algorithms is considered, using this general format yields the same structure defined in this document for the items above, thus ensuring retro-compatibility.

B.1. Format of ’ecdh_info_entry’

The format of each ’ecdh_info_entry’ (see Section 5.3 and Section 5.3.1) is generalized as follows. Given N the number of elements of the ’ecdh_parameters’ array, i.e., the number of COSE capabilities of the ECDH algorithm, then:

- ’ecdh_key_parameters’ is replaced by N elements ’ecdh_capab_i’, each of which is a CBOR array.

- The i-th array following ’ecdh_parameters’, i.e., ’ecdh_capab_i’ (i = 0, ..., N-1), is the array of COSE capabilities for the algorithm capability specified in ’ecdh_parameters’[i].

The CDDL notation [RFC8610] of the ’ecdh_info_entry’ parameter is given below.

```
ecdh_info_entry =
[
  id : gname / [ + gname ],
  ecdh_alg : int / tstr,
  ecdh_parameters : [ alg_capab_1 : any,
                      alg_capab_2 : any,
                      ...,
                      alg_capab_N : any],
  ecdh_capab_1 : [ any ],
  ecdh_capab_2 : [ any ],
  ...,
  ecdh_capab_N : [ any ],
  cred_fmt : int / null
]

gname = tstr
```

Figure 13: ’ecdh_info_entry’ with general format

B.2. Format of 'key'

The format of 'key' (see Section 6.3) is generalized as follows.

* The 'sign_params' array includes N+1 elements, whose exact structure and value depend on the value of the signature algorithm specified in 'sign_alg'.

  - The first element, i.e., 'sign_params'[0], is the array of the N COSE capabilities for the signature algorithm, as specified for that algorithm in the "Capabilities" column of the "COSE Algorithms" registry [COSE.Algorithms] (see Section 8.1 of [RFC9053]).

  - Each following element 'sign_params'[i], i.e., with index i > 0, is the array of COSE capabilities for the algorithm capability specified in 'sign_params'[0][i-1].

  For example, if 'sign_params'[0][0] specifies the key type as capability of the algorithm, then 'sign_params'[1] is the array of COSE capabilities for the COSE key type associated with the signature algorithm, as specified for that key type in the "Capabilities" column of the "COSE Key Types" registry [COSE.Key.Types] (see Section 8.2 of [RFC9053]).

* The 'ecdh_params' array includes M+1 elements, whose exact structure and value depend on the value of the ECDH algorithm specified in 'ecdh_alg'.

  - The first element, i.e., 'ecdh_params'[0], is the array of the M COSE capabilities for the ECDH algorithm, as specified for that algorithm in the "Capabilities" column of the "COSE Algorithms" registry [COSE.Algorithms] (see Section 8.1 of [RFC9053]).

  - Each following element 'ecdh_params'[i], i.e., with index i > 0, is the array of COSE capabilities for the algorithm capability specified in 'ecdh_params'[0][i-1].

  For example, if 'ecdh_params'[0][0] specifies the key type as capability of the algorithm, then 'ecdh_params'[1] is the array of COSE capabilities for the COSE key type associated with the ECDH algorithm, as specified for that key type in the "Capabilities" column of the "COSE Key Types" registry [COSE.Key.Types] (see Section 8.2 of [RFC9053]).

Appendix C. Document Updates

RFC EDITOR: PLEASE REMOVE THIS SECTION.

C.1. Version -15 to -16

* Early mentioning of invalid combinations of roles.

* Revised presentation of handling of stale Sender IDs.

* Fixed CDDL notation.

* Fixed diagnostic notation in examples.

* Possible reason to deviate from default parameter values.

* Clarifications and editorial fixes.

C.2. Version -14 to -15

* Alignment with renaming in draft-ietf-ace-key-groupcomm.

* Updated signaling of semantics for binary encoded scopes.

* Considered the upload of Access Tokens in the DTLS 1.3 Handshake.

* Fixes in IANA registrations.

* Editorial fixes.

C.3. Version -13 to -14

* Major reordering of the document sections.

* The HKDF Algorithm is specified by the HMAC Algorithm.

* Group communication does not necessarily use IP multicast.

* Generalized AIF data model, also for draft-ace-oscore-gm-admin.

* Clarifications and editorial improvements.

C.4. Version -12 to -13

* Renamed parameters about authentication credentials.

* It is optional for the Group Manager to reassign Gids by tracking "Birth Gids".

- Distinction between authentication credentials and public keys.

- Updated IANA considerations related to AIF.

- Updated textual description of registered ACE Scope Semantics value.

C.5. Version -11 to -12

- Clarified semantics of 'ecdh_info' and 'kdc_dh_creds'.

- Definition of /ace-group/GROUPNAME/kdc-pub-key moved to draft-ietf-ace-key-groupcomm.

- ace-group/ accessible also to non-members that are not Verifiers.

- Clarified what resources are accessible to Verifiers.

- Revised error handling for the newly defined resources.

- Revised use of CoAP error codes.

- Use of "Token Tranfer Request" and "Token Transfer Response".

- New parameter 'rekeying_scheme'.

- Categorization of new parameters and inherited conditional parameters.

- Clarifications on what to do in case of enhanced error responses.

- Changed UCCS to CCS.

- Authentication credentials of just joined Clients can be in rekeying messages.

- Revised names of new IANA registries.

- Clarified meaning of registered CoRE resource type.

- Alignment to new requirements from draft-ietf-ace-key-groupcomm.

- Fixes and editorial improvements.

C.6. Version -10 to -11

- Removed redundancy of key type capabilities, from 'sign_info', 'ecdh_info' and 'key'.

* New resource to retrieve the Group Manager’s authentication credential.

* New resource to retrieve material for Signature Verifiers.

* New parameter ’sign_enc_alg’ related to the group mode.

* ’cred_fmt’ takes value from the COSE Header Parameters registry.

* Improved alignment of the Join Response payload with the Group OSCORE Security Context parameters.

* Recycling Group IDs by tracking "Birth GIDs".

* Error handling in case of non available Sender IDs upon joining.

* Error handling in case EdDSA public keys with invalid Y coordinate when the pairwise mode of Group OSCORE is supported.

* Generalized proof-of-possession (PoP) for the joining node’s private key; defined Diffie-Hellman based PoP for OSCORE groups using only the pairwise mode.

* Proof-of-possession of the Group Manager’s private key in the Join Response.

* Always use ’peer_identifiers’ to convey Sender IDs as node identifiers.

* Stale Sender IDs provided when rekeying the group.

* New resource for late retrieval of stale Sender IDs.

* Added examples of message exchanges.

* Revised default values of group configuration parameters.

* Fixes to IANA registrations.

* General format of parameters related to COSE capabilities, supporting future registered COSE algorithms (new Appendix).

C.7. Version -09 to -10

* Updated non-recycling policy of Sender IDs.

* Removed policies about Sender Sequence Number synchronization.

* 'control_path' renamed to 'control_uri'.

* Format of 'get_pub_keys' aligned with draft-ietf-ace-key-groupcomm.

* Additional way to inform of group eviction.

* Registered semantics identifier for extended scope format.

* Extended error handling, with error type specified in some error responses.

* Renumbered requirements.

C.8. Version -08 to -09

* The url-path "ace-group" is used.

* Added overview of admitted methods on the Group Manager resources.

* Added exchange of parameters relevant for the pairwise mode of Group OSCORE.

* The signed value for 'client_cred_verify' includes also the scope.

* Renamed the key material object as Group_OSCORE_Input_Material object.

* Replaced 'clientId' with 'group_SenderId'.

* Added message exchange for Group Names request-response.

* No reassignment of Sender ID and Gid in the same OSCORE group.

* Updates on group rekeying contextual with request of new Sender ID.

* Signature verifiers can also retrieve Group Names and URIs.

* Removed group policy about supporting Group OSCORE in pairwise mode.

* Registration of the resource type rt="core.osc.gm".

* Update list of requirements.

* Clarifications and editorial revision.

C.9. Version -07 to -08

* AIF specific data model to express scope entries.

* A set of roles is checked as valid when processing the Join Request.

* Updated format of 'get_pub_keys' in the Join Request.

* Payload format and default values of group policies in the Join Response.

* Updated payload format of the FETCH request to retrieve authentication credentials.

* Default values for group configuration parameters.

* IANA registrations to support the AIF specific data model.

C.10. Version -06 to -07

* Alignments with draft-ietf-core-oscore-groupcomm.

* New format of 'sign_info', using the COSE capabilities.

* New format of Join Response parameters, using the COSE capabilities.

* Considerations on group rekeying.

* Editorial revision.

C.11. Version -05 to -06

* Added role of external signature verifier.

* Parameter 'rsnonce' renamed to 'kdcchallenge'.

* Parameter 'kdcchallenge' may be omitted in some cases.

* Clarified difference between group name and OSCORE Gid.

* Removed the role combination ["requester", "monitor"].

* Admit implicit scope and audience in the Authorization Request.

* New format for the 'sign_info' parameter.

- Scope not mandatory to include in the Join Request.

- Group policy about supporting Group OSCORE in pairwise mode.

- Possible individual rekeying of a single requesting node combined with a group rekeying.

- Security considerations on reusage of signature challenges.

- Addressing optional requirement OPT12 from draft-ietf-ace-key-groupcomm

- Editorial improvements.

C.12. Version -04 to -05

- Nonce N_S also in error responses to the Join Requests.

- Supporting single Access Token for multiple groups/topics.

- Supporting legal requesters/responders using the 'peer_roles' parameter.

- Registered and used dedicated label for TLS Exporter.

- Added method for uploading a new authentication credential to the Group Manager.

- Added resource and method for retrieving the current group status.

- Fixed inconsistency in retrieving group keying material only.

- Clarified retrieval of keying material for monitor-only members.

- Clarification on incrementing version number when rekeying the group.

- Clarification on what is re-distributed with the group rekeying.

- Security considerations on the size of the nonces used for the signature challenge.

- Added CBOR values to abbreviate role identifiers in the group.

C.13. Version -03 to -04

- New abstract.

- Moved general content to draft-ietf-ace-key-groupcomm

- Terminology: node name; node resource.

- Creation and pointing at node resource.

- Updated Group Manager API (REST methods and offered services).

- Size of challenges 'cnonce' and 'rsnonce'.

- Value of 'rsnonce' for reused or non-traditionally-posted tokens.

- Removed reference to RFC 7390.

- New requirements from draft-ietf-ace-key-groupcomm

- Editorial improvements.

C.14. Version -02 to -03

- New sections, aligned with the interface of ace-key-groupcomm .

- Exchange of information on the signature algorithm and related parameters, during the Token POST (Section 4.1).

- Nonce 'rsnonce' from the Group Manager to the Client (Section 4.1).

- Client PoP signature in the Key Distribution Request upon joining (Section 4.2).

- Local actions on the Group Manager, upon a new node's joining (Section 4.2).

- Local actions on the Group Manager, upon a node's leaving (Section 12).

- IANA registration in ACE Groupcomm Parameters registry.

- More fulfilled profile requirements (Appendix A).

C.15. Version -01 to -02

- Editorial fixes.

- Changed: "listener" to "responder"; "pure listener" to "monitor".

* Changed profile name to "coap_group_oscore_app", to reflect it is an application profile.

* Added the 'type' parameter for all requests to a Join Resource.

* Added parameters to indicate the encoding of authentication credentials.

* Challenge-response for proof-of-possession of signature keys (Section 4).

* Renamed 'key_info' parameter to 'sign_info'; updated its format; extended to include also parameters of the signature key (Section 4.1).

* Code 4.00 (Bad request), in responses to joining nodes providing an invalid authentication credential (Section 4.3).

* Clarifications on provisioning and checking of authentication credentials (Sections 4 and 6).

* Extended discussion on group rekeying and possible different approaches (Section 7).

* Extended security considerations: proof-of-possession of signature keys; collision of OSCORE Group Identifiers (Section 8).

* Registered three entries in the IANA registry "Sequence Number Synchronization Method" (Section 9).

* Registered one public key encoding in the "ACE Public Key Encoding" IANA registry (Section 9).

C.16. Version -00 to -01

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

* Added negotiation of signature algorithm/parameters between Client and Group Manager (Section 4).

* Updated format of the Key Distribution Response as a whole (Section 4.3).

* Added parameter 'cs_params' in the 'key' parameter of the Key Distribution Response (Section 4.3).

- New IANA registrations in the "ACE Authorization Server Request Creation Hints" registry, "ACE Groupcomm Key" registry, "OSCORE Security Context Parameters" registry and "ACE Groupcomm Profile" registry (Section 9).

Acknowledgments

The authors sincerely thank Christian Amsüss, Santiago Aragón, Stefan Beck, Carsten Bormann, Martin Gunnarsson, Rikard Höglund, Watson Ladd, Daniel Migault, Jim Schaad, Ludwig Seitz, Göran Selander and Peter van der Stok for their comments and feedback.

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

Authors' Addresses

Marco Tiloca
RISE AB
Isafjordsgatan 22
SE-164 29 Stockholm Kista
Sweden
Email: marco.tiloca@ri.se

Jiye Park
Universitaet Duisburg-Essen
Schuetzenbahn 70
45127 Essen
Germany
Email: ji-ye.park@uni-due.de

Francesca Palombini
Ericsson AB
Torshamnsgatan 23
SE-16440 Stockholm Kista
Sweden
Email: francesca.palombini@ericsson.com