

ACE Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 6, 2021

M. Tiloca
RISE AB
J. Park
Universitaet Duisburg-Essen
F. Palombini
Ericsson AB
November 02, 2020

Key Management for OSCORE Groups in ACE
draft-ietf-ace-key-groupcomm-oscore-09

Abstract

This specification 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 secured with Group Object Security for Constrained RESTful Environments (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

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 6, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1.	Introduction	3
1.1.	Terminology	4
2.	Protocol Overview	5
2.1.	Overview of the Joining Process	6
2.2.	Overview of the Group Rekeying Process	6
3.	Format of Scope	7
4.	Joining Node to Authorization Server	8
4.1.	Authorization Request	9
4.2.	Authorization Response	9
5.	Interface at the Group Manager	9
5.1.	ace-group/GROUPNAME/active	10
5.1.1.	GET Handler	10
5.2.	Admitted Methods	11
6.	Token POST and Group Joining	11
6.1.	Token Post	12
6.1.1.	'ecdh_info' Parameter	14
6.2.	Sending the Joining Request	16
6.2.1.	Value of the N_S Challenge	17
6.3.	Processing the Joining Request	17
6.4.	Joining Response	19
7.	Public Keys of Joining Nodes	23
8.	Retrieval of Updated Keying Material	25
8.1.	Retrieval of Group Keying Material	25
8.2.	Retrieval of Group Keying Material and Sender ID	26
9.	Requesting a Change of Keying Material	26
10.	Retrieval of Public Keys and Roles for Group Members	27
11.	Update of Public Key	28
12.	Retrieval of Group Policies	29
13.	Retrieval of Keying Material Version	29
14.	Retrieval of Group Status	29
15.	Retrieval of Group Names and URIs	30
16.	Request to Leave the Group	32
17.	Removal of a Group Member	32
18.	Group Rekeying Process	33
19.	Default Values for Group Configuration Parameters	35
20.	Security Considerations	38

20.1.	Management of OSCORE Groups	38
20.2.	Size of Nonces for Signature Challenge	39
20.3.	Reusage of Nonces for Signature Challenge	40
21.	IANA Considerations	40
21.1.	ACE Groupcomm Parameters Registry	40
21.2.	ACE Groupcomm Key Registry	41
21.3.	ACE Groupcomm Profile Registry	41
21.4.	Sequence Number Synchronization Method Registry	41
21.5.	OSCORE Security Context Parameters Registry	42
21.6.	TLS Exporter Label Registry	44
21.7.	AIF Registry	45
21.8.	Media Type Registrations	45
21.9.	CoAP Content-Format Registry	46
21.10.	Group OSCORE Roles Registry	46
21.11.	CoRE Resource Type Registry	47
21.12.	Expert Review Instructions	47
22.	References	48
22.1.	Normative References	48
22.2.	Informative References	51
22.3.	URIs	52
Appendix A.	Profile Requirements	52
Appendix B.	Document Updates	55
B.1.	Version -08 to -09	56
B.2.	Version -07 to -08	56
B.3.	Version -06 to -07	57
B.4.	Version -05 to -06	57
B.5.	Version -04 to -05	58
B.6.	Version -03 to -04	58
B.7.	Version -02 to -03	59
B.8.	Version -01 to -02	59
B.9.	Version -00 to -01	60
Acknowledgments	60
Authors' Addresses	61

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) [I-D.ietf-cose-rfc8152bis-struct][I-D.ietf-cose-rfc8152bis-algs] 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 over IP multicast [I-D.ietf-core-groupcomm-bis]. 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 specification is an application profile of [\[I-D.ietf-ace-key-groupcomm\]](#), which itself builds on the ACE framework for Authentication and Authorization [\[I-D.ietf-ace-oauth-authz\]](#). 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 over IP multicast.

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:

- o The terms and concepts described in the ACE framework for authentication and authorization [\[I-D.ietf-ace-oauth-authz\]](#) and in the Authorization Information Format (AIF) [\[I-D.ietf-ace-aif\]](#) to express authorization information. The terminology for entities in the considered architecture is defined in OAuth 2.0 [\[RFC6749\]](#). In particular, this includes Client (C), Resource Server (RS), and Authorization Server (AS).
- o The terms and concept 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.
- o The terms and concepts described in CBOR [\[I-D.ietf-cbor-7049bis\]](#) and COSE [\[I-D.ietf-cose-rfc8152bis-struct\]](#) [I-D.ietf-cose-rfc8152bis-algs].
- o 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".

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

Additionally, this document makes use of the following terminology.

- o Requester: member of an OSCORE group that sends request messages to other members of the group.
- o 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.
- o 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](#)].
- o Signature verifier: entity external to the OSCORE group and intended to verify the countersignature of messages exchanged in the group. An authorized signature verifier does not join the OSCORE group as an actual member, yet it can retrieve the public keys of the current group members from the Group Manager.

2. Protocol Overview

Group communication for CoAP over IP multicast has been enabled in [[I-D.ietf-core-groupcomm-bis](#)] and can be secured with Group Object Security for Constrained RESTful Environments (OSCORE) [[RFC8613](#)] as described 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 specification describes how to use [[I-D.ietf-ace-key-groupcomm](#)] and [[I-D.ietf-ace-oauth-authz](#)] to perform a number of authentication, authorization and key distribution actions, as defined in Section 2 of [[I-D.ietf-ace-key-groupcomm](#)], for an OSCORE group.

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

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

- o The Group Manager is the Key Distribution Center (KDC), acting as a Resource Server.
- o The Authorization Server associated to the Group Manager is the AS.

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 specification, the transport profile to use is pre-configured and well-known to nodes participating in constrained applications.

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

[2.1.](#) Overview of the Joining Process

A node performs the steps described in Section 4.3 of [\[I-D.ietf-ace-key-groupcomm\]](#) in order to join an OSCORE group. The format and processing of messages exchanged among the participants are further specified in [Section 4](#) and [Section 6](#) of this document.

[2.2.](#) Overview of the Group Rekeying Process

If the application requires backward and forward security, the Group Manager MUST generate new keying material and distribute it to the group (rekeying) upon membership changes.

That is, the group is rekeyed when a node joins the group as a new member, or after a current member leaves the group. By doing so, a joining node cannot access communications in the group prior its joining, while a leaving node cannot access communications in the group after its leaving.

The keying material distributed through a group rekeying MUST include:

- o 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.

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

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

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 Sender ID of each member in that group.

The Group Manager MUST support the Group Rekeying Process described in [Section 18](#). Future application profiles may define alternative message formats and distribution schemes to perform group rekeying.

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 to use.

To this end, this profile uses the Authorization Information Format (AIF) [\[I-D.ietf-ace-aif\]](#), and defines the following AIF specific data model AIF-OSCORE-GROUPCOMM.

With reference to the generic AIF model

$$\text{AIF-Generic}<\text{Toid}, \text{Tperm}> = [* [\text{Toid}, \text{Tperm}]]$$

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

Then, for each scope entry:

- o the object identifier ("Toid") is specialized as a CBOR text string, specifying the group name for the scope entry;
- o the permission set ("Tperm") is specialized as a CBOR unsigned integer with value R, specifying the role(s) that the client wishes to take in the group (REQ2). 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 table in Figure 1.
- * the set of N numbers is converted into the single value R, by taking each numeric identifier X₁, X₂, ..., X_N to the power of two, 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 countersignature of intercepted messages

Figure 1: Numeric identifier of roles in the OSCORE group

The CDDL [[RFC8610](#)] definition of the AIF-OSCORE-GROUPCOMM data model is as follows:

```
AIF-OSCORE-GROUPCOMM = AIF_Generic<path, permissions>
```

```
path = tstr ; Group name
permissions = uint . bits roles
roles = &(
    Requester: 1,
    Responder: 2,
    Monitor: 3,
    Verifier: 4
)
```

Future specifications that define new roles MUST register a corresponding numeric identifier in the "Group OSCORE Roles" Registry defined in [Section 21.10](#) of this specification.

4. Joining Node to Authorization Server

This section describes how the joining node interacts with the AS in order to be authorized to join an OSCORE group under a given Group

Manager. In particular, it considers a joining node that intends to contact that Group Manager for the first time.

The message exchange between the joining node and the AS consists of the messages Authorization Request and Authorization Response defined in Section 3 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 here.

4.1. Authorization Request

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

- o 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](#). In particular, for each OSCORE group to join:
 - + The group name is encoded as a CBOR text string.
 - + The set of requested roles is expressed as a single CBOR unsigned integer, computed as defined in [Section 3](#) (REQ2) from the numerical abbreviations defined in Figure 1 for each requested role (OPT7).

4.2. Authorization Response

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

- o 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 specification.
- o 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 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 4.1](#).

5. Interface at the Group Manager

The Group Manager provides the interface defined in Section 4.1 of [[I-D.ietf-ace-key-groupcomm](#)], with one additional sub-resource defined in [Section 5.1](#) of this specification.

Furthermore, [Section 5.2](#) 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 (REQ7aa).

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](#)]) (REQ1).

The Resource Type (rt=) Link Target Attribute value "core.osc.gm" is registered in [Section 21.11](#) (REQ7a), 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](#)].

[5.1.](#) ace-group/GROUPNAME/active

This resource implements a GET handler.

[5.1.1.](#) GET Handler

The handler expects a GET request.

The Group Manager verifies that the group name in the /ace-group/GROUPNAME/active path is a subset of the 'scope' stored in the Access Token associated to the requesting client.

The Group Manager also verifies that the roles granted to the requesting client in the group allow it to perform this operation on this resource (REQ7aa). If either verification fails, the Group Manager MUST respond with a 4.01 (Unauthorized) error message.

Additionally, the handler verifies that the requesting client is a current member of the group. If verification fails, the Group Manager MUST respond with a 4.01 (Unauthorized) error message.

If verification succeeds, the handler returns a 2.05 (Content) message containing the CBOR simple value True if the group is currently active, or the CBOR simple value False 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.

The method to set the current group status, i.e. active or inactive, is out of the scope of this specification, and is defined for the administrator interface of the Group Manager specified in [[I-D.ietf-ace-oscore-gm-admin](#)].

5.2. 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	-
ace-group/GROUPNAME/	G Po	G Po	Po *	Po
ace-group/GROUPNAME/active	G	G	-	-
ace-group/GROUPNAME/pub-key	G F	G F	G 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	-	-	-

Type1 = Member as Requester and/or Responder	G = GET
Type2 = Member as Monitor	F = FETCH
Type3 = Non-member / Authorized to be Verifier	Po = POST
(*) = cannot join the group as Verifier	Pu = PUT
Type4 = Non-member / Not authorized to be Verifier	D = DELETE

Figure 2: Admitted CoAP Methods on the Group Manager Resources

6. Token POST and Group Joining

The rest of this section describes the interactions between the joining node and the Group Manager, i.e. the sending of the Access Token and the Request-Response exchange to join the OSCORE group. The message exchange between the joining node and the Group Manager consists of the messages defined in [Section 3.3](#) and 4.3 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 here.

A signature verifier provides the Group Manager with an Access Token, as described in [Section 6.1](#), just as any another joining node does. However, unlike candidate group members, it does not join any OSCORE group, i.e. it does not perform the joining process defined in [Section 6.2](#). After successfully posting an Access Token, a signature verifier is authorized to perform only the operations specified in [Section 10](#), to retrieve the public keys of group members, and only for the OSCORE groups specified in the validated Access Token. The Group Manager MUST respond with a 4.01 (Unauthorized) error message, in case a signature verifier attempts to access any other endpoint than /ace-group/GROUPNAME/pub-key at the Group Manager.

6.1. Token Post

The Token post exchange is defined in Section 3.3 of [\[I-D.ietf-ace-key-groupcomm\]](#).

Additionally to what defined in [\[I-D.ietf-ace-key-groupcomm\]](#), the following applies.

- o The CoAP POST request MAY additionally contain the following parameter, which, if included, MUST have the corresponding values:
 - * 'ecdh_info' defined in [Section 6.1.1](#), encoding the CBOR simple value Null to require information on the ECDH algorithm, the ECDH algorithm parameters, the ECDH key parameters and on the exact encoding of public keys used in the group, 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.

- o The 'kdcchallenge' parameter contains a dedicated nonce N_S generated by the Group Manager. For the N_S value, it is RECOMMENDED to use a 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 6.2](#)).

The 'kdcchallenge' parameter MAY be omitted from the 2.01 (Created) response, if the 'scope' of the Access Token specifies only the role "monitor" or only the role "verifier" or both of them, for each and every of the specified groups.

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

- * 'sign_alg' takes value from the "Value" column of the "COSE Algorithms" Registry [[COSE.Algorithms](#)].
- * 'sign_parameters' is a CBOR array including the following two elements:
 - + 'sign_alg_capab', encoded as a CBOR array. Its format and value are the same of the COSE capabilities 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_type_capab', encoded as a CBOR array. Its format and value are the same of the COSE capabilities 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](#)] (REQ4).
- * 'sign_key_parameters' is a CBOR array. Its format and value are the same of the COSE capabilities 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).
- * 'pub_key_enc' takes value 1 ("COSE_Key") from the 'Confirmation Key' column of the "CWT Confirmation Method" Registry [[CWT.Confirmation.Methods](#)], so indicating that public keys in the OSCORE group are encoded as COSE Keys [[I-D.ietf-cose-rfc8152bis-struct](#)]. Future specifications may define additional values for this parameter.
- o If 'ecdh_info' is included in the request, the Group Manager MAY include the 'ecdh_info' parameter defined in [Section 6.1.1](#), with the same encoding. Note that the field 'id' takes as value the group name, or array of group names, for which the corresponding 'ecdh_info_entry' 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 previously retrieved this information by other means, 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 (OPT2a).

Additionally, if allowed by the used transport profile of ACE, the joining node may instead provide the Access Token to the Group

Manager by other means, e.g. during a secure session establishment (see Section 3.3.2 of [[I-D.ietf-ace-dtls-authorize](#)]).

6.1.1. 'ecdh_info' Parameter

The 'ecdh_info' parameter is an OPTIONAL parameter of the Token Post response message defined in Section 5.1.2. of [[I-D.ietf-ace-oauth-authz](#)].

This parameter is used to require and retrieve from the Group Manager information and parameters about the ECDH algorithm and about the public keys to be used in the OSCORE group to compute a static-static Diffie-Hellman shared secret [[NIST-800-56A](#)], in case the group supports the pairwise mode of Group OSCORE [[I-D.ietf-core-oscore-groupcomm](#)].

When used in the request, the 'ecdh_info' parameter encodes the CBOR simple value Null, to require information and parameters on the ECDH algorithm and on the public keys to be used to compute Diffie-Hellman shared secrets in the OSCORE group.

The CDDL notation [[RFC8610](#)] of the 'ecdh_info' parameter formatted as in the request is given below.

```
ecdh_info_req = nil
```

The 'ecdh_info' parameter of the 2.01 (Created) response is a CBOR array of one or more elements. The number of elements is at most the number of OSCORE groups the client has been authorized to join.

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

- o 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.
- o 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](#)].
- o The third element 'ecdh_parameters' is a CBOR array indicating the parameters of the ECDH algorithm used in the OSCORE group identified by 'gname'. The CBOR array includes the following two

elements, and its exact content depends on the value of the 'ecdh_alg' element.

- * 'ecdh_alg_capab', encoded as a CBOR array. Its format and value are the same of the COSE capabilities for the algorithm indicated in 'ecdh_alg', as specified for that algorithm in the "Capabilities" column of the "COSE Algorithms" Registry [[COSE.Algorithms](#)].
- * 'ecdh_key_type_capab', encoded as a CBOR array. Its format and value are the same of the COSE capabilities 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](#)].
- o 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 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](#)].
- o The fifth element 'pub_key_enc' is CBOR integer indicating the encoding of public keys used in the OSCORE group identified by 'gname'. It takes value 1 ("COSE_Key") from the 'Confirmation Key' column of the "CWT Confirmation Method" Registry [[CWT.Confirmation.Methods](#)], so indicating that public keys in the OSCORE group are encoded as COSE Keys [[I-D.ietf-cose-rfc8152bis-struct](#)]. Future specifications may define additional values for this parameter.

The CDDL notation [[RFC8610](#)] of the 'ecdh_info' parameter formatted as in the response is given below.


```

ecdh_info_res = [ + ecdh_info_entry ]

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

gname = tstr

```

6.2. Sending the Joining Request

The joining node requests to join the OSCORE group by sending a Joining Request message to the related group-membership resource at the Group Manager, as per Section 4.3 of [\[I-D.ietf-ace-key-groupcomm\]](#).

Additionally to what defined in [\[I-D.ietf-ace-key-groupcomm\]](#), the following applies.

- o 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.
- o The 'get_pub_keys' parameter is present only if the joining node wants to retrieve the public keys of the group members from the Group Manager during the joining process (see [Section 7](#)). Otherwise, this parameter MUST NOT be present.

If this parameter is present and its value is non-null, each element of the first inner CBOR array 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](#).

- o 'cnonce' contains a dedicated nonce N_C generated by the joining node. For the N_C value, it is RECOMMENDED to use a 8-byte long random nonce.
- o The signature encoded in the 'client_cred_verify' parameter is computed by the joining node by using the same private key and countersignature algorithm it intends to use for signing messages in the OSCORE group. Moreover, N_S is as defined in [Section 6.2.1](#).

6.2.1. Value of the N_S Challenge

The value of the N_S challenge is determined as follows.

1. If the joining node has posted the Access Token to the /authz-info endpoint of the Group Manager as in [Section 6.1](#), 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 2.01 (Created) response to the Token POST, or the one possibly specified in a 4.00 (Bad Request) response to a following Joining Request (see [Section 6.3](#)).
2. If the Token posting has relied on the DTLS profile of ACE [[I-D.ietf-ace-dtls-authorize](#)] with the Access Token as content of the "psk_identity" field of the ClientKeyExchange message [[RFC6347](#)], 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 21.6](#) of this specification.

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

[Section 20.3](#) provides security considerations on the reuse of the N_S challenge.

6.3. Processing the Joining Request

The Group Manager processes the Joining Request as defined in [Section 4.1.2.1 of \[I-D.ietf-ace-key-groupcomm\]](#). Additionally, the following applies.

- o The Group Manager MUST return a 5.03 (Service Unavailable) response in case the OSCORE group that the joining node has been trying to join is currently inactive (see [Section 5.1](#)).
- o In case the joining node is not going to join the group exclusively as monitor and the Joining Request does not include the 'client_cred' parameter, the joining process fails if the Group Manager either: i) does not store a public key with an accepted format for the joining node; or ii) stores multiple public keys with an accepted format for the joining node.
- o To compute the signature contained in 'client_cred_verify', the Group Manager considers:

- * as signed value, the value of the 'scope' parameter from the Joining 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. In particular, N_S is determined as described in [Section 6.2.1](#), while N_C is the nonce provided in the 'cnonce' parameter of the Joining Request;
 - * the countersignature algorithm used in the OSCORE group, and possible corresponding parameters;
 - * the public key of the joining node, either retrieved from the 'client_cred' parameter, or already stored as acquired from previous interactions with the joining node.
- o A 4.00 (Bad Request) response from the Group Manager to the joining node MUST have content format application/ace+cbor. The response payload is a CBOR map which MUST contain the 'sign_info' parameter, including a single element 'sign_info_entry' pertaining to the OSCORE group that the joining node has tried to join with the Joining Request. If the group supports the pairwise mode of Group OSCORE, the CBOR map MUST contain also the 'ecdh_info' parameter, including a single element 'ecdh_info_entry' pertaining to the OSCORE group that the joining node has tried to join with the Joining Request.
 - o The Group Manager MUST return a 4.00 (Bad Request) response in case the 'scope' parameter is not present in the Joining Request, or if it is present and specifies any set of roles not included in the following list: "requester", "responder", "monitor", ("requester", "responder"). Future specifications that define a new role MUST define possible sets of roles including the new one and existing ones, that are acceptable to specify in the 'scope' parameter of a Joining Request.
 - o The Group Manager MUST return a 4.00 (Bad Request) response in case the Joining Request includes the 'client_cred' parameter but does not include both the 'cnonce' and 'client_cred_verify' parameters.
 - o The Group Manager MUST return a 4.00 (Bad Request) response in case it cannot retrieve a public key with an accepted format for the joining node, either from the 'client_cred' parameter or as already stored.
 - o When receiving a 4.00 Bad Request response, the joining node SHOULD send a new Joining Request to the Group Manager, where:

- * The 'cnonce' parameter MUST include a new dedicated nonce N_C generated by the joining node.
- * The 'client_cred' parameter MUST include a public key compatible with the encoding, countersignature algorithm and possible associated parameters indicated by the Group Manager.
- * The 'client_cred_verify' parameter MUST include a signature computed as described in [Section 6.2](#), by using the public key indicated in the current 'client_cred' parameter, with the countersignature algorithm and possible associated parameters indicated by the Group Manager. If the error response from the Group Manager included the 'kdcchallenge' parameter, the joining node MUST use its content as new N_S challenge to compute the signature.

6.4. Joining Response

If the processing of the Joining Request described in [Section 6.3](#) 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 4.1.2.1 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.

- o The Group Manager selects an available OSCORE Sender ID in the OSCORE group, and exclusively assigns it to the joining node. Consistently with Section 3 of [\[I-D.ietf-core-oscore-groupcomm\]](#), the Group Manager MUST assign a Sender ID that has never been assigned before in the OSCORE group. The Group Manager MUST NOT assign a Sender ID to the joining node if this joins the group exclusively with the role of monitor, according to what specified in the Access Token (see [Section 4.2](#)).
- o The Group Manager stores the association between i) the public key of the joining node; and ii) the Group Identifier (Gid), i.e. the OSCORE ID Context, associated to 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 Joining Response is formatted as defined in Section 4.1.2.1 of [\[I-D.ietf-ace-key-groupcomm\]](#), with the following additions:

- o The 'gkty' parameter identifies a key of type "Group_OSCORE_Input_Material object", defined in [Section 21.2](#) of this specification.
- o 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 specification and extends the OSCORE_Input_Material object encoded in CBOR as defined in Section 3.2.1 of [\[I-D.ietf-ace-oscore-profile\]](#). In particular, it contains the additional parameters 'group_senderId' 'cs_alg', 'cs_params', 'cs_key_params', 'cs_key_enc', 'ecdh_alg', 'ecdh_params' and 'ecdh_key_params' defined in [Section 21.5](#) of this specification.

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

- * The 'ms' parameter MUST be present and includes the OSCORE Master Secret value used in the OSCORE group.
- * The 'hkdf' parameter, if present, has as value the KDF algorithm used in the OSCORE group.
- * The 'alg' parameter, if present, has as value the AEAD algorithm used in the OSCORE group.
- * The 'salt' parameter, if present, has as value the OSCORE Master Salt used in the OSCORE group.
- * The 'contextId' parameter MUST be present and has as value the Group Identifier (Gid), i.e. the OSCORE ID Context of the OSCORE group.
- * The 'group_senderId' parameter, if present, has as value the OSCORE Sender ID assigned to the joining node by the Group Manager, as described above. This parameter is not present if the node joins the group exclusively with the role of monitor, according to what specified in the Access Token (see [Section 4.2](#)). In any other case, this parameter MUST be present.
- * The 'cs_alg' parameter MUST be present and specifies the algorithm used to countersign messages in the OSCORE group. This parameter takes values from the "Value" column of the "COSE Algorithms" Registry [\[COSE.Algorithms\]](#).

- * The 'cs_params' parameter MAY be present and specifies the parameters for the counter signature algorithm. This parameter is a CBOR array, which includes the following two elements:
 - + 'sign_alg_capab', with the same encoding as defined in [Section 6.1](#). The value is the same as in the Token Post response where the 'sign_parameters' value was non-null.
 - + 'sign_key_type_capab', with the same encoding as defined in [Section 6.1](#). The value is the same as in the Token Post response where the 'sign_parameters' value was non-null.
- * The 'cs_key_params' parameter MAY be present and specifies the parameters for the key used with the counter signature algorithm. This parameter is a CBOR array, with the same non-null encoding and value as 'sign_key_parameters' of the [Section 6.1](#).
- * The 'cs_key_enc' parameter MAY be present and specifies the encoding of the public keys of the group members. This parameter is a CBOR integer, whose value is 1 ("COSE_Key") taken from the 'Confirmation Key' column of the "CWT Confirmation Method" Registry [[CWT.Confirmation.Methods](#)], so indicating that public keys in the OSCORE group are encoded as COSE Keys [[I-D.ietf-cose-rfc8152bis-struct](#)]. Future specifications may define additional values for this parameter. If this parameter is not present, 1 ("COSE_Key") MUST be assumed as default value.
- * The 'ecdh_alg' parameter, if present, specifies the ECDH algorithm used in the OSCORE group, if this supports the pairwise mode of Group OSCORE. This parameter takes values from the "Value" column of the "COSE Algorithms" Registry [[COSE.Algorithms](#)]. This parameter MUST be present if the OSCORE group supports the pairwise mode of Group OSCORE, and MUST NOT be present otherwise.
- * The 'ecdh_params' parameter, if present, specifies the parameters for the ECDH algorithm. It MUST be present if the 'ecdh_alg' parameter is present, and MUST NOT be present otherwise. This parameter is a CBOR array, which includes the following two elements:
 - + 'ecdh_alg_capab', with the same encoding as defined in [Section 6.1.1](#). The value is the same as in the Token Post response where the 'ecdh_parameters' value is non-null.

- + 'ecdh_key_type_capab', with the same encoding as defined in [Section 6.1.1](#). The value is the same as in the Token Post response where the 'ecdh_parameters' value is non-null.
- * The 'ecdh_key_params' parameter, if present, specifies the parameters for the key used with the ECDH algorithm. It MUST be present if the 'ecdh_alg' parameter is present, and MUST NOT be present otherwise. This parameter is a CBOR array, with the same non-null encoding and value of 'ecdh_key_parameters' defined in [Section 6.1.1](#).
- o The 'exp' parameter MUST be present.
- o The 'ace-groupcomm-profile' parameter MUST be present and has value coap_group_oscore_app (TBD3), which is defined in [Section 21.3](#) of this specification.
- o The 'pub_keys' parameter, if present, includes the public keys requested by the joining node by means of the 'get_pub_keys' parameter in the Joining Request. If public keys are encoded as COSE_Keys, each of them has as 'kid' the Sender ID that the corresponding owner has in the OSCORE group, thus used as group member identifier encoded as a CBOR byte string (REQ9).

If the joining node has asked for the public keys of all the group members, i.e. 'get_pub_keys' had value Null in the Joining Request, then the Group Manager provides only the public keys of the group members that are relevant to the joining node. That is, in such a case, 'pub_keys' includes only: i) the public keys of the responders currently in the OSCORE group, in case the joining node is configured (also) as requester; and ii) the public keys of the requesters currently in the OSCORE group, in case the joining node is configured (also) as responder or monitor.

- o The 'group_policies' parameter SHOULD be present, and SHOULD include the following elements:
 - * "Sequence Number Synchronization Method" defined in Section 4.1.2.1 of [[I-D.ietf-ace-key-groupcomm](#)], with default value 1 ("Best effort");
 - * "Key Update Check Interval" defined in Section 4.1.2.1 of [[I-D.ietf-ace-key-groupcomm](#)], with default value 3600;
 - * "Expiration Delta" defined in Section 4.1.2.1 of [[I-D.ietf-ace-key-groupcomm](#)], with default value 0.

Finally, the joining node uses the information received in the Joining Response to set up the Group OSCORE Security Context, as described in Section 2 of [[I-D.ietf-core-oscore-groupcomm](#)]. In addition, the joining node maintains an association between each public key retrieved from the 'pub_keys' 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:

- o The joining node MUST NOT process an incoming request message, if protected by a group member whose public key is not associated to the role "Requester".
- o The joining node MUST NOT process an incoming response message, if protected by a group member whose public key is not associated to the role "Responder".
- o The joining node MUST NOT use the pairwise mode of Group OSCORE to process messages in the group, if the Joining 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 18](#)). As a consequence, the joining node is not able to access secure communication in the OSCORE group occurred prior its joining.

7. Public Keys of Joining Nodes

Source authentication of a message sent within the group and protected with Group OSCORE is ensured by means of a digital counter 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 public key from a trusted key 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 public keys of the group members, and provides those public keys to group members if requested to. Upon joining an OSCORE group, a joining node is thus expected to provide its own public key to the Group Manager.

In particular, one of the following four cases can occur when a new node joins an OSCORE group.

- o The joining node is going to join the group exclusively as monitor. That is, it is not going to send messages to the group, and hence to produce signatures with its own private key. In this case, the joining node is not required to provide its own public key to the Group Manager, which thus does not have to perform any check related to the public key encoding, or to a countersignature algorithm and possible associated parameters for that joining node. In case that joining node still provides a public key in the 'client_cred' parameter of the Joining Request (see [Section 6.2](#)), the Group Manager silently ignores that parameter, as well as the related parameters 'cnonce' and 'client_cred_verify'.
- o The Group Manager already acquired the public key of the joining node during a past joining process. In this case, the joining node MAY choose not to provide again its own public key to the Group Manager, in order to limit the size of the Joining Request. The joining node MUST provide its own public key again if it has provided the Group Manager with multiple public keys during past joining processes, intended for different OSCORE groups. If the joining node provides its own public key, the Group Manager performs consistency checks as per [Section 6.3](#) and, in case of success, considers it as the public key associated to the joining node in the OSCORE group.
- o 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. The proof-of-possession key is compatible with the encoding as well as with the counter signature algorithm and possible associated parameters used in the OSCORE group. Then, the Group Manager considers the proof-of-possession key as the public key associated to the joining node in the OSCORE group. If the joining node is aware that the proof-of-possession key is also valid for the OSCORE group, it MAY not provide it again as its own public key to the Group Manager. The joining node MUST provide its own public key again if it has provided the Group Manager with multiple public keys during past joining processes, intended for different OSCORE groups. If the joining node provides its own public key in the 'client_cred' parameter of the Joining Request (see [Section 6.2](#)), the Group Manager performs consistency checks as per [Section 6.3](#) and, in case of success, considers it as the public key associated to the joining node in the OSCORE group.

2. The proof-of-possession key is not compatible with the encoding or with the counter signature algorithm and possible associated parameters used in the OSCORE group. In this case, the joining node MUST provide a different compatible public key to the Group Manager in the 'client_cred' parameter of the Joining Request (see [Section 6.2](#)). Then, the Group Manager performs consistency checks on this latest provided public key as per [Section 6.3](#) and, in case of success, considers it as the public key associated to the joining node in the OSCORE group.
- o 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 public key in the 'client_cred' parameter of the Joining Request targeting the group-membership endpoint (see [Section 6.2](#)).

8. Retrieval of 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 Joining 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.

8.1. Retrieval of 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.

In particular, 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 4.1.2.2 of [\[I-D.ietf-ace-key-groupcomm\]](#). The Key Distribution Response is formatted as defined in Section 4.1.2.2 of [\[I-D.ietf-ace-key-groupcomm\]](#). In addition:

- o The 'key' parameter is formatted as defined in [Section 6.4](#) of this specification, with the difference that it does not include the 'group_SenderId' parameter.
- o The 'exp' parameter MUST be present.

- o 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 [\[I-D.ietf-core-oscore-groupcomm\]](#).

8.2. Retrieval of Group Keying Material and Sender ID

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

In particular, 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 4.1.6.2 of [\[I-D.ietf-ace-key-groupcomm\]](#). The Key Distribution Response is formatted as defined in Section 4.1.6.2 of [\[I-D.ietf-ace-key-groupcomm\]](#). In addition:

- o The 'key' parameter is formatted as defined in [Section 6.4](#) of this specification. In particular, if the requesting group member has exclusively the role of monitor, no 'group_SenderId' is specified within the 'key' parameter.

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

- o 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. Requesting a Change of Keying Material

As discussed in Section 2.4.2 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 4.5](#) of

[[I-D.ietf-ace-key-groupcomm](#)]. In particular, 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 4.1.6.1 of [[I-D.ietf-ace-key-groupcomm](#)], and performs one of the following actions.

1. If the requesting group member has exclusively the role of monitor, the Group Manager replies with a 4.00 (Bad Request) error response.
2. Otherwise, the Group Manager takes one of the following actions.
 - a. The Group Manager rekeys the OSCORE group. That is, the Group Manager generates new group keying material for that group (see [Section 18](#)), and replies to the group member with a group rekeying message as defined in [Section 18](#), providing the new group keying material. Then, the Group Manager rekeys the rest of the OSCORE group, as discussed in [Section 18](#).

The Group Manager SHOULD perform a group rekeying only if already scheduled to occur shortly, e.g. according to an application-dependent rekeying period, 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 (OPT8).

- b. The Group Manager generates a new Sender ID for that group member and replies with a Key Renewal Response, formatted as defined in Section 4.1.6.1 of [[I-D.ietf-ace-key-groupcomm](#)]. In particular, the CBOR Map in the response payload includes a single parameter 'group_SenderId' defined in [Section 21.1](#) of this document, specifying the new Sender ID of the group member encoded as a CBOR byte string.

Consistently with Section 2.4.3.1 of [[I-D.ietf-core-oscore-groupcomm](#)], the Group Manager MUST assign a new Sender ID that has never been assigned before in the OSCORE group.

[10.](#) Retrieval of Public Keys and Roles for Group Members

A group member or a signature verifier may need to retrieve the public keys of (other) group members. To this end, the group member or signature verifier sends a Public Key Request message to the Group Manager, as per Section 4.6 of [[I-D.ietf-ace-key-groupcomm](#)]. In

particular, it sends the request to the endpoint `/ace-group/GROUPNAME/pub-key` at the Group Manager.

If the Public Key Request uses the method `FETCH`, the Public Key Request is formatted as defined in Section 4.1.3.1 of [\[I-D.ietf-ace-key-groupcomm\]](#). In particular:

- o Each element (if any) of the first inner CBOR array is formatted as in the first inner CBOR array of the `'get_pub_keys'` parameter of the Joining Request when the parameter value is non-null (see [Section 6.2](#)).
- o Each element (if any) of the second inner CBOR array is a CBOR byte string (REQ9), which encodes the Sender ID of the group member for which the associated public key is requested.

Upon receiving the Public Key Request, the Group Manager processes it as per [Section 4.1.3.1](#) or 4.1.3.2 of [\[I-D.ietf-ace-key-groupcomm\]](#), depending on the request method being `FETCH` or `GET`, respectively. Additionally, if the Public Key Request uses the method `FETCH`, the Group Manager silently ignores node identifiers included in the `'get_pub_keys'` parameter of the request that are not associated to any current group member.

The success Public Key Response is formatted as defined in [Section 4.1.3.1](#) or 4.1.3.2 of [\[I-D.ietf-ace-key-groupcomm\]](#), depending on the request method being `FETCH` or `GET`, respectively.

11. Update of Public Key

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

To this end, the group member sends a Public Key Update Request message to the Group Manager, as per Section 4.7 of [\[I-D.ietf-ace-key-groupcomm\]](#). In particular, it sends a CoAP POST request to the endpoint `/ace-group/GROUPNAME/nodes/NODENAME/pub-key` at the Group Manager.

Upon receiving the Group Leaving Request, the Group Manager processes it as per Section 4.1.7.1 of [\[I-D.ietf-ace-key-groupcomm\]](#), with the following additions.

- o If the requesting group member has exclusively the role of monitor, the Group Manager replies with a 4.00 (Bad request) error response.
- o The N_S signature challenge is computed as per point (1) in [Section 6.2.1](#) (REQ17).
- o If the request is successfully processed, the Group Manager stores the association between i) the new public key of the group member; and ii) the Group Identifier (Gid), i.e. the OSCORE ID Context, associated to 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.

12. Retrieval of 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 4.8 of [[I-D.ietf-ace-key-groupcomm](#)]. In particular, 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 4.1.4.1 of [[I-D.ietf-ace-key-groupcomm](#)]. The success Policies Response is formatted as defined in Section 4.1.4.1 of [[I-D.ietf-ace-key-groupcomm](#)].

13. Retrieval of 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 4.9 of [[I-D.ietf-ace-key-groupcomm](#)]. In particular, 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 4.1.5.1 of [[I-D.ietf-ace-key-groupcomm](#)]. The success Version Response is formatted as defined in Section 4.1.5.1 of [[I-D.ietf-ace-key-groupcomm](#)].

14. Retrieval of Group Status

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

In particular, the group member sends a CoAP GET request to the endpoint `/ace-group/GROUPNAME/active` at the Group Manager defined in [Section 5.1](#) of this specification, where GROUPNAME is the name of the OSCORE group. The success Group Version Response is formatted as defined in [Section 5.1](#) of this specification.

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 3 gives an overview of the exchange described above.

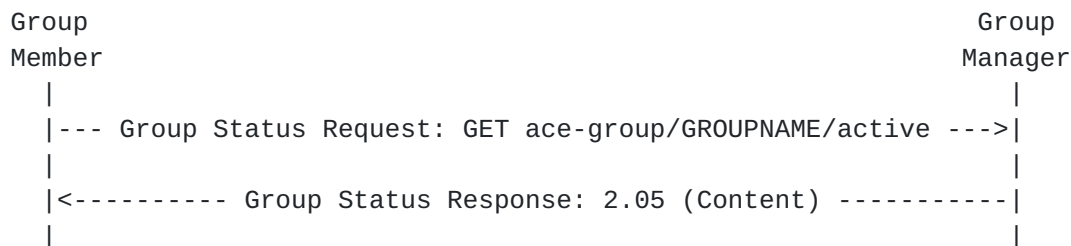


Figure 3: Message Flow of Group Status Request-Response

15. Retrieval of Group Names and URIs

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.

- o 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.
- o As current group member in several groups, the node has missed a previous group rekeying in one of them (see [Section 18](#)). 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 public keys from the Group Manager (see [Section 8.1](#), [Section 8.2](#) and [Section 10](#)). 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 [\[I-D.ietf-core-oscore-groupcomm\]](#)).

- o As signature verifier, the node also refers to a group name for retrieving the required public keys from the Group Manager (see [Section 10](#)). 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 counter signatures, and especially the retrieval of required, new public keys from the Group Manager.

In either case, the node only knows the current Gid of the group, as learnt 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 either join the group as a candidate group member, get the latest keying material as a current group member, or retrieve public keys 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 of [\[I-D.ietf-ace-key-groupcomm\]](#).

In particular, the node sends a CoAP FETCH request to the endpoint /ace-group at the Group Manager formatted as defined in Section 4.1.1.1 of [\[I-D.ietf-ace-key-groupcomm\]](#). Each element of the CBOR array 'gid' is a CBOR byte string (REQ7b), 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 4.1.1.1 of [\[I-D.ietf-ace-key-groupcomm\]](#). The success Group Name and URI Retrieval Response is formatted as defined in Section 4.1.1.1 of [\[I-D.ietf-ace-key-groupcomm\]](#). In particular, each element of the CBOR array 'gid' is a CBOR byte string (REQ7b), 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 MUST NOT maintain an association between the former pair and any other Gid for that group than the current, most recent one.

Figure 4 gives an overview of the exchanges described above.

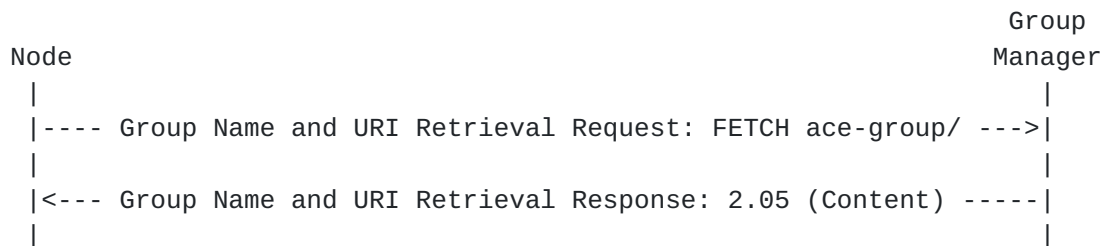


Figure 4: Message Flow of Group Name and URI Retrieval Request-Response

16. Request to 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 4.10 of [I-D.ietf-ace-key-groupcomm]. In particular, 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 4.1.6.3 of [I-D.ietf-ace-key-groupcomm].

17. Removal of a Group Member

Other than after a spontaneous request to the Group Manager as described in [Section 16](#), a node may be forcibly removed from the OSCORE group, e.g. due to expired or revoked authorization.

If, upon joining the group (see [Section 6.2](#)), the leaving node specified a URI in the 'control_path' parameter defined in Section 4.1.2.1 of [I-D.ietf-ace-key-groupcomm], the Group Manager MUST inform the leaving node of its eviction, by sending a DELETE request targeting the URI specified in the 'control_path' parameter (OPT9).

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

- o The Group Manager frees the OSCORE Sender ID value of the leaving node. However, this value MUST NOT become available for possible upcoming joining nodes in the same group.

- o The Group Manager cancels the association between, on one hand, the public key of the leaving node and, on the other hand, the Group Identifier (Gid) associated to the OSCORE group together with the freed OSCORE Sender ID value. The Group Manager deletes the public key of the leaving node, if that public key has no remaining association with any pair (Gid, Sender ID).

If the application requires forward security, the Group Manager MUST generate updated security parameters and group keying material, and provide it to the remaining group members (see [Section 18](#)). As a consequence, the leaving node is not able to acquire the new security parameters and group keying material distributed after its leaving.

Same considerations in Section 5 of [[I-D.ietf-ace-key-groupcomm](#)] apply here as well, considering the Group Manager acting as KDC.

18. 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.

Consistently with Section 2.3 of [[I-D.ietf-core-oscore-groupcomm](#)], the Group Manager MUST assign a Gid that has never been assigned before to the OSCORE group.

Furthermore, the Group Manager MUST preserve the same unchanged Sender IDs for all group members. This avoids affecting the retrieval of public keys from the Group Manager as well as the verification of group messages.

The Group Manager MUST support at least the following group rekeying scheme. Future application profiles may define alternative message formats and distribution schemes.

As group rekeying message, the Group Manager uses the same format of the Joining Response message in [Section 6.4](#). In particular:

- o Only the parameters 'gkty', 'key', 'num', 'ace-groupcomm-profile' and 'exp' are present.
- o The 'ms' parameter of the 'key' parameter specifies the new OSCORE Master Secret value.

- o The 'contextId' parameter of the 'key' parameter specifies the new Group ID used as OSCORE ID Context value.

The Group Manager separately sends a group rekeying message to each group member to be rekeyed.

Each rekeying message MUST be secured with the pairwise secure communication channel between the Group Manager and the group member used during the joining process. In particular, each rekeying message can target the 'control_path' URI path defined in Section 4.1.2.1 of [[I-D.ietf-ace-key-groupcomm](#)] (OPT9), if provided by the intended recipient upon joining the group (see [Section 6.2](#)).

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.

This approach requires group members to act (also) as servers, in order to correctly handle unsolicited group rekeying messages from the Group Manager. In particular, 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 [[I-D.ietf-ace-oscore-profile](#)].

Group members and the Group Manager SHOULD additionally support alternative rekeying approaches that do not require group members to act (also) as servers. A number of such approaches are defined in Section 4.4 of [[I-D.ietf-ace-key-groupcomm](#)]. In particular, a group member may subscribe for updates to the group-membership resource of the group, at the endpoint /ace-group/GROUPNAME/ of the Group Manager. This can rely on CoAP Observe [[RFC7641](#)] or on a full-fledged Pub-Sub model [[I-D.ietf-core-coap-pubsub](#)] with the Group Manager acting as Broker.

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 to 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 13](#)) and then possibly requesting the current keying material (see [Section 8.1](#)).

Furthermore, some of these group members can be in multiple groups, all of which associated to 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 to that Group Manager.

19. 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 [\[I-D.ietf-ace-oscore-gm-admin\]](#).

The Group Manager SHOULD use the same default values defined in [Section 3.2 of \[RFC8613\]](#) for both the HKDF algorithm and the AEAD algorithm used in the group.

The Group Manager SHOULD use the following default values for the algorithm, algorithm parameters and key parameters used to countersign messages in the group, consistently with the "COSE Algorithms" Registry [\[COSE.Algorithms\]](#), the "COSE Key Types" Registry [\[COSE.Key.Types\]](#) and the "COSE Elliptic Curves" Registry [\[COSE.Elliptic.Curves\]](#).

- o For the algorithm 'cs_alg' used to countersign messages in the group, the signature algorithm EdDSA [\[RFC8032\]](#).
- o For the parameters 'cs_params' of the counter signature algorithm:
 - * The array `[[OKP], [OKP, Ed25519]]`, indicating the elliptic curve Ed25519 [\[RFC8032\]](#), in case EdDSA is assumed or specified for 'cs_alg'.
 - * The array `[[EC2], [EC2, P-256]]`, indicating the elliptic curve P-256, in case ES256 [\[RFC6979\]](#) is specified for 'cs_alg'.

- * The array `[[EC2], [EC2, P-384]]`, indicating the elliptic curve P-384, in case ES384 [\[RFC6979\]](#) is specified for `'cs_alg'`.
 - * The array `[[EC2], [EC2, P-521]]`, indicating the elliptic curve P-521, in case ES512 [\[RFC6979\]](#) is specified for `'cs_alg'`.
 - * The array `[[], [RSA]]`, in case PS256, PS384 or PS512 [\[RFC8017\]](#) is specified for `'cs_alg'`.
- o For the parameters `'cs_key_params'` of the key used with the counter signature algorithm:
- * The array `[OKP, Ed25519]` as pair (key type, elliptic curve), in case EdDSA is assumed or specified for `'cs_alg'` and Ed25519 is assumed or specified within the second array of `'cs_params'`.
 - * The array `[OKP, Ed448]` as pair (key type, elliptic curve), in case EdDSA is assumed or specified for `'cs_alg'` and the elliptic curve Ed448 [\[RFC8032\]](#) is specified within the second array of `'cs_params'`.
 - * The array `[EC2, P-256]` as pair (key type, elliptic curve), in case ES256 [\[RFC6979\]](#) is specified for `'cs_alg'` and the elliptic curve P-256 is assumed or specified within the second array of `'cs_params'`.
 - * The array `[EC2, P-384]` as pair (key type, elliptic curve), in case ES384 [\[RFC6979\]](#) is specified for `'cs_alg'` and the elliptic curve P-384 is specified within the second array of `'cs_params'`.
 - * The array `[EC2, P-521]` as pair (key type, elliptic curve), in case ES512 [\[RFC6979\]](#) is specified for `'cs_alg'` and the elliptic curve P-521 is specified within the second array of `'cs_params'`.
 - * The array `[RSA]` indicating RSA as key type, in case PS256, PS384 or PS512 [\[RFC8017\]](#) is specified for `'cs_alg'`.
- o For the `'cs_key_enc'` encoding of the public keys of the group members, COSE_Key from the "CWT Confirmation Methods" Registry [\[CWT.Confirmation.Methods\]](#).

If the group supports the pairwise mode of Group OSCORE, the Group Manager SHOULD use the following default values for the algorithm, algorithm parameters and key parameters used to compute static-static Diffie-Hellman shared secrets, 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](#)].

- o For the 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 [[I-D.ietf-cose-rfc8152bis-algs](#)].
- o For the parameters 'ecdh_params' of the ECDH algorithm:
 - * The array [[OKP], [OKP, X25519]], indicating the elliptic curve X25519 [[RFC8032](#)], in case EdDSA is assumed or specified for 'cs_alg'.
 - * The array [[EC2], [EC2, P-256]], indicating the elliptic curve P-256, in case ES256 [[RFC6979](#)] is specified for 'cs_alg'.
 - * The array [[EC2], [EC2, P-384]], indicating the elliptic curve P-384, in case ES384 [[RFC6979](#)] is specified for 'cs_alg'.
 - * The array [[EC2], [EC2, P-521]], indicating the elliptic curve P-521, in case ES512 [[RFC6979](#)] is specified for 'cs_alg'.
- o For the parameters 'ecdh_key_params' of the key used with the ECDH algorithm:
 - * The array [OKP, X25519] as pair (key type, elliptic curve), in case EdDSA is assumed or specified for 'cs_alg' and X25519 is assumed or specified within the second array of 'ecdh_params'.
 - * The array [OKP, X448] as pair (key type, elliptic curve), in case EdDSA is assumed or specified for 'cs_alg' and the elliptic curve X448 [[RFC8032](#)] is specified within the second array of 'ecdh_params'.
 - * The array [EC2, P-256] as pair (key type, elliptic curve), in case ES256 [[RFC6979](#)] is specified for 'cs_alg' and the elliptic curve P-256 is assumed or specified within the second array of 'ecdh_params'.
 - * The array [EC2, P-384] as pair (key type, elliptic curve), in case ES384 [[RFC6979](#)] is specified for 'cs_alg' and the elliptic curve P-384 is specified within the second array of 'ecdh_params'.
 - * The array [EC2, P-521] as pair (key type, elliptic curve), in case ES512 [[RFC6979](#)] is specified for 'cs_alg' and the elliptic curve P-521 is specified within the second array of 'ecdh_params'.

20. Security Considerations

Security considerations for this profile are inherited from [\[I-D.ietf-ace-key-groupcomm\]](#), the ACE framework for Authentication and Authorization [\[I-D.ietf-ace-oauth-authz\]](#), and the specific transport profile of ACE signalled by the AS, such as [\[I-D.ietf-ace-dtls-authorize\]](#) and [\[I-D.ietf-ace-oscore-profile\]](#).

The following security considerations also apply for this profile.

20.1. 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.

- o Management of group keying material (see Section 3.1 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). According to the specific application requirements, this can include rekeying the group upon changes in its membership. In particular, renewing the group keying material is required upon a new node's joining or a current node's leaving, in case backward security and forward security have to be preserved, respectively.
- o Provisioning and retrieval of public keys (see Section 2 of [\[I-D.ietf-core-oscore-groupcomm\]](#)). The Group Manager acts as key repository of public keys of group members, and provides them upon request.
- o Synchronization of sequence numbers (see Section 6.1 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 Joining 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, the joining node can use its own public key as asymmetric proof-of-possession key to establish a secure channel with the Group Manager, e.g. as in Section 3.2 of [\[I-D.ietf-ace-dtls-authorize\]](#). However, this requires such proof-of-possession key to be compatible with the encoding as well as with the countersignature 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.

20.2. Size of Nonces for Signature Challenge

With reference to the Joining Request message in [Section 6.2](#), the proof-of-possession signature included in 'client_cred_verify' is computed over the challenge $N_C \parallel N_S$, where \parallel denotes concatenation.

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

As defined in [Section 6.2.1](#), 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.

- o If the Access Token is not explicitly posted to the /authz-info endpoint of the Group Manager, then N_S is computed as a 32-byte long challenge share. For an example, see point (2) of [Section 6.2.1](#).
- o If the Access Token has been explicitly posted to the /authz-info endpoint of the Group Manager, 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 6.2.1](#). This is provided either in the 2.01 response to the Token Post (see [Section 6.1](#)), or in a 4.00 response to a following Joining Request (see [Section 6.3](#)). In either case, it is RECOMMENDED to use a 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.

- o 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

2³² new posted Access Tokens, while the average collision for N_C will happen after 2³² new Joining Requests. This amounts to considerably more token provisionings than the expected new joinings of 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.

- o Section 7 of [[I-D.ietf-ace-oscore-profile](#)] 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 specification 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.

[20.3.](#) Reusage of Nonces for Signature Challenge

As long as the Group Manager preserves the same N_S value currently associated to 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 signature challenge for multiple Joining Requests to that Group Manager.

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

[21.](#) IANA Considerations

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

This document has the following actions for IANA.

[21.1.](#) ACE Groupcomm Parameters Registry

IANA is asked to register the following entry to the "ACE Groupcomm Parameters" Registry defined in Section 8.5 of [[I-D.ietf-ace-key-groupcomm](#)].

- o Name: group_senderId
- o CBOR Key: TBD1
- o CBOR Type: Byte string
- o Reference: [[This specification]] ([Section 9](#))

[21.2.](#) ACE Groupcomm Key Registry

IANA is asked to register the following entry to the "ACE Groupcomm Key" Registry defined in Section 8.6 of [[I-D.ietf-ace-key-groupcomm](#)].

- o Name: Group_OSCORE_Input_Material object
- o Key Type Value: TBD2
- o Profile: "coap_group_oscore_app", defined in [Section 21.3](#) of this specification.
- o Description: A Group_OSCORE_Input_Material object encoded as described in [Section 6.4](#) of this specification.
- o Reference: [[This specification]] ([Section 6.4](#))

[21.3.](#) ACE Groupcomm Profile Registry

IANA is asked to register the following entry to the "ACE Groupcomm Profile" Registry defined in Section 8.7 of [[I-D.ietf-ace-key-groupcomm](#)].

- o Name: coap_group_oscore_app
- o Description: Application profile to provision keying material for participating in group communication protected with Group OSCORE as per [[I-D.ietf-core-oscore-groupcomm](#)].
- o CBOR Value: TBD3
- o Reference: [[This specification]] ([Section 6.4](#))

[21.4.](#) Sequence Number Synchronization Method Registry

IANA is asked to register the following entries in the "Sequence Number Synchronization Method" Registry defined in Section 8.9 of [[I-D.ietf-ace-key-groupcomm](#)].

- o Name: Best effort

- o Value: 1
- o Description: No action is taken.
- o Reference: [[I-D.ietf-core-oscore-groupcomm](#)] (Appendix E.1)
- o Name: Baseline
- o Value: 2
- o Description: The first received request sets the baseline reference point, and is discarded with no delivery to the application.
- o Reference: [[I-D.ietf-core-oscore-groupcomm](#)] (Appendix E.2)
- o Name: Echo challenge-response
- o Value: 3
- o Description: Challenge response using the Echo Option for CoAP from [[I-D.ietf-core-echo-request-tag](#)].
- o Reference: [[I-D.ietf-core-oscore-groupcomm](#)] (Appendix E.3)

21.5. OSCORE Security Context Parameters Registry

IANA is asked to register the following entries in the "OSCORE Security Context Parameters" Registry defined in Section 9.4 of [[I-D.ietf-ace-oscore-profile](#)].

- o Name: group_SenderId
- o CBOR Label: TBD4
- o CBOR Type: bstr
- o Registry: -
- o Description: OSCORE Sender ID assigned to a member of an OSCORE group
- o Reference: [[This specification]] ([Section 6.4](#))
- o Name: cs_alg

- o CBOR Label: TBD5
- o CBOR Type: tstr / int
- o Registry: COSE Algorithms
- o Description: OSCORE Counter Signature Algorithm Value
- o Reference: [[This specification]] ([Section 6.4](#))

- o Name: cs_params
- o CBOR Label: TBD6
- o CBOR Type: array
- o Registry: COSE Algorithms, COSE Key Types, COSE Elliptic Curves
- o Description: OSCORE Counter Signature Algorithm Additional Parameters
- o Reference: [[This specification]] ([Section 6.4](#))

- o Name: cs_key_params
- o CBOR Label: TBD7
- o CBOR Type: array
- o Registry: COSE Algorithms, COSE Key Types, COSE Elliptic Curves
- o Description: OSCORE Counter Signature Key Additional Parameters
- o Reference: [[This specification]] ([Section 6.4](#))

- o Name: cs_key_enc
- o CBOR Label: TBD8
- o CBOR Type: integer
- o Registry: CWT Confirmation Methods
- o Description: Encoding of Public Keys to be used with the OSCORE Counter Signature Algorithm

- o Reference: [[This specification]] ([Section 6.4](#))
- o Name: ecdh_alg
- o CBOR Label: TBD9
- o CBOR Type: tstr / int
- o Registry: COSE Algorithms
- o Description: OSCORE ECDH Algorithm Value
- o Reference: [[This specification]] ([Section 6.4](#))
- o Name: ecdh_params
- o CBOR Label: TBD10
- o CBOR Type: array
- o Registry: COSE Algorithms, COSE Key Types, COSE Elliptic Curves
- o Description: OSCORE ECDH Algorithm Additional Parameters
- o Reference: [[This specification]] ([Section 6.4](#))
- o Name: ecdh_key_params
- o CBOR Label: TBD11
- o CBOR Type: array
- o Registry: COSE Algorithms, COSE Key Types, COSE Elliptic Curves
- o Description: OSCORE ECDH Key Additional Parameters
- o Reference: [[This specification]] ([Section 6.4](#))

21.6. TLS Exporter Label Registry

IANA is asked to register the following entry to the "TLS Exporter Label" Registry defined in [Section 6 of \[RFC5705\]](#) and updated in [Section 12 of \[RFC8447\]](#).

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

- o DTLS-OK: Y
- o Recommended: N
- o Reference: [[This specification]] ([Section 6.2.1](#))

[21.7.](#) AIF Registry

IANA is asked to register the following entry to the "Toid" sub-registry of the "AIF" Registry defined in Section 5.2 of [\[I-D.ietf-ace-aif\]](#).

- o Name: oscore-group-name
- o Description/Specification: group name of the OSCORE group, as specified in [[This specification]].

IANA is asked to register the following entry to the "Tperm" sub-Registry of the "AIF" Registry defined in Section 5.2 of [\[I-D.ietf-ace-aif\]](#).

- o Name: oscore-group-roles
- o Description/Specification: role(s) of the member of the OSCORE group, as specified in [[This specification]].

[21.8.](#) Media Type Registrations

This specification registers the 'application/aif-groupcomm-oscore+cbor' media type for the AIF specific data model AIF-OSCORE-GROUPCOMM defined in [Section 3](#) of [[This specification]]. This registration follows the procedures specified in [\[RFC6838\]](#).

These media type has parameters for specifying the object identifier ("Toid") and set of permissions ("Tperm") defined for the AIF-generic model in [\[I-D.ietf-ace-aif\]](#); default values are the values "oscore-group-name" for "Toid" and "oscore-group-roles" for "Tperm".

Type name: application

Subtype name: aif-groupcomm-oscore+cbor

Required parameters: "Toid", "Tperm"

Optional parameters: none

Encoding considerations: Must be encoded as a CBOR array, each element of which is an array [Toid, Tperm] as defined in [Section 3](#) of [\[\[This specification\]\]](#).

Security considerations: See [Section 20](#) of [\[\[This specification\]\]](#).

Interoperability considerations: n/a

Published specification: [\[\[This specification\]\]](#)

Applications that use this media type: The type is used by applications that want to express authorization information about joining OSCORE groups, as specified in [\[\[This specification\]\]](#).

Additional information: n/a

Person & email address to contact for further information:
iesg@ietf.org [\[1\]](#)

Intended usage: COMMON

Restrictions on usage: None

Author: Marco Tiloca marco.tiloca@ri.se [\[2\]](#)

Change controller: IESG

[21.9](#). CoAP Content-Format Registry

IANA is asked to register the following entry to the "CoAP Content-Formats" registry, within the "CoRE Parameters" registry:

Media Type: application/aif-groupcomm-oscore+cbor;Toid="oscore-group-name",Tperm"oscore-group-roles"

Encoding: -

ID: TBD12

Reference: [\[\[This specification\]\]](#)

[21.10](#). Group OSCORE Roles Registry

This specification establishes the IANA "Group OSCORE Roles" Registry. The Registry has been created to use the "Expert Review Required" registration procedure [\[RFC8126\]](#). Expert review guidelines are provided in [Section 21.12](#).

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 [\[\[This specification\]\]](#).

The columns of this registry are:

- o 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.
- o 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" [\[RFC8126\]](#).
- o Description: This field contains a brief description of the role.
- o 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 [\[\[This specification\]\]](#).

[21.11.](#) CoRE Resource Type Registry

IANA is asked to register a new Resource Type (rt=) Link Target Attribute in the "Resource Type (rt=) Link Target Attribute Values" subregistry under the "Constrained Restful Environments (CoRE) Parameters" [\[CORE.Parameters\]](#) registry.

- o Value: "core.osc.gm"
- o Description: Group-membership resource of an OSCORE Group Manager.
- o Reference: [\[\[This specification\]\]](#)

[21.12.](#) 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:

- o 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. Expert 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 objective of clarity and completeness should not be registered.

- o 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.
- o Experts should take into account the expected usage of roles when approving point assignment. 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.
- o Specifications are recommended. When specifications are not provided, the description provided needs to have sufficient information to verify the points above.

22. References

22.1. Normative References

[CORE.Parameters]

IANA, "Constrained RESTful Environments (CoRE) Parameters", <<https://www.iana.org/assignments/core-parameters/core-parameters.xhtml>>.

[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.Key.Types]

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

[CWT.Confirmation.Methods]

IANA, "CWT Confirmation Methods",
<<https://www.iana.org/assignments/cwt/cwt.xhtml#confirmation-methods>>.

[I-D.ietf-ace-aif]

Bormann, C., "An Authorization Information Format (AIF) for ACE", [draft-ietf-ace-aif-00](#) (work in progress), July 2020.

[I-D.ietf-ace-key-groupcomm]

Palombini, F. and M. Tiloca, "Key Provisioning for Group Communication using ACE", [draft-ietf-ace-key-groupcomm-10](#) (work in progress), November 2020.

[I-D.ietf-ace-oauth-authz]

Seitz, L., Selander, G., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "Authentication and Authorization for Constrained Environments (ACE) using the OAuth 2.0 Framework (ACE-OAuth)", [draft-ietf-ace-oauth-authz-35](#) (work in progress), June 2020.

[I-D.ietf-ace-oscore-profile]

Palombini, F., Seitz, L., Selander, G., and M. Gunnarsson, "OSCORE Profile of the Authentication and Authorization for Constrained Environments Framework", [draft-ietf-ace-oscore-profile-13](#) (work in progress), October 2020.

[I-D.ietf-cbor-7049bis]

Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", [draft-ietf-cbor-7049bis-16](#) (work in progress), September 2020.

[I-D.ietf-core-oscore-groupcomm]

Tiloca, M., Selander, G., Palombini, F., and J. Park, "Group OSCORE - Secure Group Communication for CoAP", [draft-ietf-core-oscore-groupcomm-10](#) (work in progress), November 2020.

- [I-D.ietf-cose-rfc8152bis-algs]
Schaad, J., "CBOR Object Signing and Encryption (COSE): Initial Algorithms", [draft-ietf-cose-rfc8152bis-algs-12](#) (work in progress), September 2020.
- [I-D.ietf-cose-rfc8152bis-struct]
Schaad, J., "CBOR Object Signing and Encryption (COSE): Structures and Process", [draft-ietf-cose-rfc8152bis-struct-14](#) (work in progress), September 2020.
- [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>>.
- [RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", [BCP 13](#), [RFC 6838](#), DOI 10.17487/RFC6838, January 2013, <<https://www.rfc-editor.org/info/rfc6838>>.
- [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>>.
- [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>>.

22.2. Informative References

- [I-D.ietf-ace-dtls-authorize]
Gerdes, S., Bergmann, O., Bormann, C., Selander, G., and L. Seitz, "Datagram Transport Layer Security (DTLS) Profile for Authentication and Authorization for Constrained Environments (ACE)", [draft-ietf-ace-dtls-authorize-14](#) (work in progress), October 2020.
- [I-D.ietf-ace-oscore-gm-admin]
Tiloca, M., Hoeglund, R., Stok, P., Palombini, F., and K. Hartke, "Admin Interface for the OSCORE Group Manager", [draft-ietf-ace-oscore-gm-admin-01](#) (work in progress), November 2020.
- [I-D.ietf-core-coap-pubsub]
Koster, M., Keranen, A., and J. Jimenez, "Publish-Subscribe Broker for the Constrained Application Protocol (CoAP)", [draft-ietf-core-coap-pubsub-09](#) (work in progress), September 2019.
- [I-D.ietf-core-echo-request-tag]
Amsuess, C., Mattsson, J., and G. Selander, "CoAP: Echo, Request-Tag, and Token Processing", [draft-ietf-core-echo-request-tag-10](#) (work in progress), July 2020.

[I-D.ietf-core-groupcomm-bis]

Dijk, E., Wang, C., and M. Tiloca, "Group Communication for the Constrained Application Protocol (CoAP)", [draft-ietf-core-groupcomm-bis-02](#) (work in progress), November 2020.

[I-D.tiloca-core-oscore-discovery]

Tiloca, M., Amsuess, C., and P. Stok, "Discovery of OSCORE Groups with the CoRE Resource Directory", [draft-tiloca-core-oscore-discovery-07](#) (work in progress), November 2020.

[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>>.

[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>>.

[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>>.

[22.3. URIs](#)

[1] <mailto:iesg@ietf.org>

[2] <mailto:marco.tiloca@ri.se>

[Appendix A. Profile Requirements](#)

This appendix lists the specifications on this application profile of ACE, based on the requirements defined in [Appendix A](#) of [\[I-D.ietf-ace-key-groupcomm\]](#).

- o REQ1 - If the value of the GROUPNAME URI path and the group name in the Access Token scope (gname in Section 3.1 of [\[I-D.ietf-ace-key-groupcomm\]](#)) do not match, specify the mechanism to map the GROUPNAME value in the URI to the group name: not applicable, since a match is required.
- o REQ2 - Specify the encoding and value of roles, for scope entries of 'scope': see [Section 3](#) and [Section 4.1](#).
- o REQ3 - if used, specify the acceptable values for 'sign_alg': values from the "Value" column of the "COSE Algorithms" Registry [\[COSE.Algorithms\]](#).
- o 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 [\[COSE.Algorithms\]](#) and from the COSE key type capabilities as specified in the "COSE Key Types" Registry [\[COSE.Key.Types\]](#).
- o 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 [\[COSE.Key.Types\]](#).
- o REQ6 - If used, specify the acceptable values for 'pub_key_enc': 1 ("COSE_Key") from the 'Confirmation Key' column of the "CWT Confirmation Method" Registry [\[CWT.Confirmation.Methods\]](#). Future specifications may define additional values for this parameter.
- o REQ7a - 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 21.11](#).
- o REQ7aa - Define what operations (i.e. CoAP methods) are allowed on each resource, for each role defined in REQ2: see [Section 5.2](#).
- o REQ7b - Specify the exact encoding of group identifier (see Section 4.1.1.1 of [\[I-D.ietf-ace-key-groupcomm\]](#)): CBOR byte string (see [Section 15](#)).
- o REQ7 - Format of the 'key' value: see [Section 6.4](#).
- o REQ8 - Acceptable values of 'gkty': Group_OSCORE_Input_Material object (see [Section 6.4](#)).

- o REQ9 - Specify the format of the identifiers of group members: CBOR byte string (see [Section 6.4](#) and [Section 10](#)).
- o REQ10 - Specify the communication protocol that the members of the group must use: CoAP [[RFC7252](#)], possibly over IP multicast [[I-D.ietf-core-groupcomm-bis](#)].
- o REQ11 - Specify the security protocols that the group members must use to protect their communication: Group OSCORE [[I-D.ietf-core-oscore-groupcomm](#)].
- o REQ12 - Specify and register the application profile identifier: `coap_group_oscore_app` (see [Section 21.3](#)).
- o REQ13 - Specify policies at the KDC to handle member ids that are not included in 'get_pub_keys': see [Section 10](#).
- o REQ14 - If used, specify the format and content of 'group_policies' and its entries: see [Section 6.4](#); see the three values defined and registered as content of the entry "Sequence Number Synchronization Method" (see [Section 21.4](#)).
- o REQ15 - 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](#).
- o REQ16 - Specify how the communication is secured between the Client and KDC: by means of any transport profile of ACE [[I-D.ietf-ace-oauth-authz](#)] between Client and Group Manager that complies with the requirements in [Appendix C](#) of [[I-D.ietf-ace-oauth-authz](#)].
- o REQ17 - Specify how the nonce `N_S` is generated, if the token is not being posted (e.g. if it is used directly to validate TLS instead): see [Section 6.2.1](#).
- o REQ18 - Specify if 'mgt_key_material' is used, and if yes specify its format and content: not used in this version of the profile.
- o REQ19 - 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](#)].
- o OPT1 (Optional) - Specify the encoding of public keys, of 'client_cred', and of 'pub_keys' if COSE_Keys are not used: no.
- o OPT2a (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 [\[I-D.tiloca-core-oscore-discovery\]](#).

- o OPT2b (Optional) - Specify additional parameters used in the Token Post exchange: 'ecdh_info', to negotiate the ECDH algorithm, ECDH algorithm parameters, ECDH key parameters and exact encoding of public keys used in the group, in case the joining node supports the pairwise mode of Group OSCORE.
- o OPT3 (Optional) - Specify the encoding of 'pub_keys_repos' if the default is not used: no.
- o OPT4 (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.
- o OPT5 (Optional) - Specify the behavior of the handler in case of failure to retrieve a public key for the specific node: send a 4.00 Bad Request response to a Joining Request (see [Section 6.3](#)).
- o OPT6 (Optional) - Specify possible or required payload formats for specific error cases: send a 4.00 Bad Request response to a Joining Request (see [Section 6.3](#)).
- o OPT7 (Optional) - Specify CBOR values to use for abbreviating identifiers of roles in the group or topic: see [Section 4.1](#).
- o OPT8 (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](#)).
- o OPT9 (Optional) - Specify the functionalities implemented at the 'control_path' resource hosted at the Client, including message exchange encoding and other details (see Section 4.1.2.1 of [\[I-D.ietf-ace-key-groupcomm\]](#)): see [Section 17](#) for the eviction of a group member; see [Section 18](#) for the group rekeying process.
- o OPT10 (Optional) - Specify how the identifier of the sender's public key is included in the group request: no.

[Appendix B](#). Document Updates

RFC EDITOR: PLEASE REMOVE THIS SECTION.

B.1. Version -08 to -09

- o The url-path "ace-group" is used.
- o Added overview of admitted methods on the Group Manager resources.
- o Added exchange of parameters relevant for the pairwise mode of Group OSCORE.
- o The signed value for 'client_cred_verify' includes also the scope.
- o Renamed the key material object as Group_OSCORE_Input_Material object.
- o Replaced 'clientId' with 'group_SenderId'.
- o Added message exchange for Group Names request-response.
- o No reassignment of Sender ID and Gid in the same OSCORE group.
- o Updates on group rekeying contextual with request of new Sender ID.
- o Signature verifiers can also retrieve Group Names and URIs.
- o Removed group policy about supporting Group OSCORE in pairwise mode.
- o Registration of the resource type rt="core.osc.gm".
- o Update list of requirements.
- o Clarifications and editorial revision.

B.2. Version -07 to -08

- o AIF specific data model to express scope entries.
- o A set of roles is checked as valid when processing the Joining Request.
- o Updated format of 'get_pub_keys' in the Joining Request.
- o Payload format and default values of group policies in the Joining Response.
- o Updated payload format of the FETCH request to retrieve public keys.

- o Default values for group configuration parameters.
- o IANA registrations to support the AIF specific data model.

B.3. Version -06 to -07

- o Alignments with [draft-ietf-core-oscure-groupcomm](#).
- o New format of 'sign_info', using the COSE capabilities.
- o New format of Joining Response parameters, using the COSE capabilities.
- o Considerations on group rekeying.
- o Editorial revision.

B.4. Version -05 to -06

- o Added role of external signature verifier.
- o Parameter 'rsnonce' renamed to 'kdcchallenge'.
- o Parameter 'kdcchallenge' may be omitted in some cases.
- o Clarified difference between group name and OSCORE Gid.
- o Removed the role combination ["requester", "monitor"].
- o Admit implicit scope and audience in the Authorization Request.
- o New format for the 'sign_info' parameter.
- o Scope not mandatory to include in the Joining Request.
- o Group policy about supporting Group OSCORE in pairwise mode.
- o Possible individual rekeying of a single requesting node combined with a group rekeying.
- o Security considerations on reuse of signature challenges.
- o Addressing optional requirement OPT9 from [draft-ietf-ace-key-groupcomm](#)
- o Editorial improvements.

B.5. Version -04 to -05

- o Nonce N_S also in error responses to the Joining Requests.
- o Supporting single Access Token for multiple groups/topics.
- o Supporting legal requesters/responders using the 'peer_roles' parameter.
- o Registered and used dedicated label for TLS Exporter.
- o Added method for uploading a new public key to the Group Manager.
- o Added resource and method for retrieving the current group status.
- o Fixed inconsistency in retrieving group keying material only.
- o Clarified retrieval of keying material for monitor-only members.
- o Clarification on incrementing version number when rekeying the group.
- o Clarification on what is re-distributed with the group rekeying.
- o Security considerations on the size of the nonces used for the signature challenge.
- o Added CBOR values to abbreviate role identifiers in the group.

B.6. Version -03 to -04

- o New abstract.
- o Moved general content to [draft-ietf-ace-key-groupcomm](#)
- o Terminology: node name; node resource.
- o Creation and pointing at node resource.
- o Updated Group Manager API (REST methods and offered services).
- o Size of challenges 'cnonce' and 'rsnonce'.
- o Value of 'rsnonce' for reused or non-traditionally-posted tokens.
- o Removed reference to [RFC 7390](#).
- o New requirements from [draft-ietf-ace-key-groupcomm](#)

- o Editorial improvements.

B.7. Version -02 to -03

- o New sections, aligned with the interface of `ace-key-groupcomm` .
- o Exchange of information on the countersignature algorithm and related parameters, during the Token POST ([Section 4.1](#)).
- o Nonce 'rsnonce' from the Group Manager to the Client ([Section 4.1](#)).
- o Client PoP signature in the Key Distribution Request upon joining ([Section 4.2](#)).
- o Local actions on the Group Manager, upon a new node's joining ([Section 4.2](#)).
- o Local actions on the Group Manager, upon a node's leaving ([Section 12](#)).
- o IANA registration in ACE Groupcomm Parameters Registry.
- o More fulfilled profile requirements (Appendix A).

B.8. Version -01 to -02

- o Editorial fixes.
- o Changed: "listener" to "responder"; "pure listener" to "monitor".
- o Changed profile name to "coap_group_oscore_app", to reflect it is an application profile.
- o Added the 'type' parameter for all requests to a Join Resource.
- o Added parameters to indicate the encoding of public keys.
- o Challenge-response for proof-of-possession of signature keys ([Section 4](#)).
- o Renamed 'key_info' parameter to 'sign_info'; updated its format; extended to include also parameters of the countersignature key ([Section 4.1](#)).
- o Code 4.00 (Bad request), in responses to joining nodes providing an invalid public key ([Section 4.3](#)).

- o Clarifications on provisioning and checking of public keys (Sections [4](#) and [6](#)).
- o Extended discussion on group rekeying and possible different approaches ([Section 7](#)).
- o Extended security considerations: proof-of-possession of signature keys; collision of OSCORE Group Identifiers ([Section 8](#)).
- o Registered three entries in the IANA Registry "Sequence Number Synchronization Method Registry" ([Section 9](#)).
- o Registered one public key encoding in the "ACE Public Key Encoding" IANA Registry ([Section 9](#)).

B.9. Version -00 to -01

- o Changed name of 'req_aud' to 'audience' in the Authorization Request ([Section 3.1](#)).
- o Added negotiation of countersignature algorithm/parameters between Client and Group Manager ([Section 4](#)).
- o Updated format of the Key Distribution Response as a whole ([Section 4.3](#)).
- o Added parameter 'cs_params' in the 'key' parameter of the Key Distribution Response ([Section 4.3](#)).
- o 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 Santiago Aragon, Stefan Beck, Carsten Bormann, Martin Gunnarsson, Rikard Hoeglund, Daniel Migault, Jim Schaad, Ludwig Seitz, Goeran 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
Kista SE-164 29 Stockholm
Sweden

Email: marco.tiloca@ri.se

Jiye Park
Universitaet Duisburg-Essen
Schuetzenbahn 70
Essen 45127
Germany

Email: ji-ye.park@uni-due.de

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

Email: francesca.palombini@ericsson.com

