ACE Working Group
Internet-Draft
Intended status: Standards Track
Expires: 15 November 2025

F. Palombini
Ericsson
C. Sengul
Brunel University
M. Tiloca
RISE AB

14 May 2025

CoAP Publish-Subscribe Profile for Authentication and Authorization for Constrained Environments (ACE)
draft-ietf-ace-coap-pubsub-profile-00

Abstract

This document defines an application profile of the Authentication and Authorization for Constrained Environments (ACE) framework, to enable secure group communication in the Publish-Subscribe (Pub-Sub) architecture for the Constrained Application Protocol (CoAP) [draft-ietf-core-coap-pubsub], where Publishers and Subscribers communicate through a Broker. This profile relies on 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. This document specifies the provisioning and enforcement of authorization information for Clients to act as Publishers and/or Subscribers, as well as the provisioning of keying material and security parameters that Clients use for protecting their communications end-to-end through the Broker.

Note to RFC Editor: Please replace "[draft-ietf-core-coap-pubsub]" with the RFC number of that document and delete this paragraph.

Discussion Venues

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

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

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

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Copyright Notice

Table of Contents

1. Introduction 3
  1.1. Terminology 4
2. Application Profile Overview 5
3. Getting Authorisation to Join a Pub-Sub security group (A) . 9
  3.1. Topic Discovery at the Broker (Optional) 11
  3.2. AS Discovery at the Broker (Optional) 11
  3.3. KDC Discovery at the Broker (Optional) 12
  3.4. Authorisation Request/Response for the KDC and the Broker 12
    3.4.1. Format of Scope 13
  3.5. Authorization response 16
  3.6. Token Transfer to the KDC 17
4. Client Group Communication Interface at the KDC 18
  4.1. Joining a Security Group 20
    4.1.1. Join Request 21
    4.1.2. Join Response 24
    4.1.3. Join Error Handling 28
  4.2. Other Group Operations through the KDC 29
    4.2.1. Obtaining Latest Information on the Group, Group Keying Material, and Sender ID 29

4.2.2. Requesting a New Sender ID 31
4.2.3. Updating Authentication Credentials 32
4.2.4. Leaving a Group 32
5. Group Rekeying Process 33
6. Pub-Sub Protected Communication 34
6.1. Using COSE to Protect the Published Topic Data 35
6.2. AEAD Nonce 38
6.3. Replay Checks 39
7. Security Considerations 40
8. IANA Considerations 41
8.1. ACE Groupcomm Key Types 42
8.2. ACE Groupcomm Profiles 42
8.3. CoRE Resource Type 42
8.4. AIF Media-Type Sub-Parameters 43
8.5. CoAP Content-Formats 43
8.6. TLS Exporter Labels 44
9. References 44
9.1. Normative References 44
9.2. Informative References 47
Appendix A. Requirements on Application Profiles 48
A.1. Mandatory-to-Address Requirements 48
A.2. Optional-to-Address Requirements 52
Appendix B. Document Updates 53
B.1. Version -10 to -11 53
B.2. Version -09 to -10 53
B.3. Version -08 to -09 54
B.4. Version -07 to -08 54
B.5. Version -06 to -07 55
Acknowledgments 55
Authors' Addresses 55

1. Introduction

In a publish-subscribe (Pub-Sub) scenario, devices acting as Publishers and/or Subscribers communicate via a Broker that enforces store-and-forward messaging between those. This effectively enables a form of group communication, where all the Publishers and Subscribers participating in the same Pub-Sub topic are considered members of the same application group associated with that topic.

With a focus on the Pub-Sub architecture defined in [I-D.ietf-core-coap-pubsub] for the Constrained Application Protocol (CoAP) [RFC7252], this document defines an application profile of the Authentication and Authorization for Constrained Environments (ACE) framework [RFC9200], which enables Pub-Sub communication where a group of Publishers and Subscribers securely communicate through a Broker using CoAP.

Building on the message formats and processing defined in [RFC9594], this document specifies the provisioning and enforcement of authorization information for Clients to act as Publishers and/or Subscribers at the Broker, as well as the provisioning of keying material and security parameters that Clients use for protecting end-to-end their communications via the Broker.

In order to protect the Pub-Sub operations at the Broker as well as the provisioning of keying material and security parameters, this profile relies on protocol-specific transport profiles of ACE (e.g., [RFC9202], [RFC9203], or [I-D.ietf-ace-edhoc-oscore-profile]) 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.

The content of published messages that are circulated by the Broker is protected end-to-end between the corresponding Publisher and the intended Subscribers. To this end, this profile relies on COSE [RFC9052][RFC9053] and on keying material provided to the Publishers and Subscribers participating in the same Pub-Sub topic. In particular, source authentication of published content is achieved by means of the corresponding Publisher signing such content with its own private key.

1.1. Terminology

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

Readers are expected to be familiar with:

* The terms and concepts described in the ACE framework for Authentication and Authorization [RFC9200]. The terminology for entities in the considered architecture is defined in OAuth 2.0 [RFC6749]. In particular, this includes Client, Resource Server (RS), and Authorization Server (AS).

* The Authorization Information Format (AIF) [RFC9237] used to express authorization information.

* The terms and concept related to the message formats and processing specified in [RFC9594], for provisioning and renewing keying material in group communication scenarios. These include the abbreviations REQx and OPTx denoting the numbered mandatory-to-address and optional-to-address requirements, respectively.

* The terms and concepts described in CDDL [RFC8610], CBOR [RFC8949], and COSE [RFC9052][RFC9053][RFC9338].

* The terms and concepts described in CoAP [RFC7252]. Note that the term "endpoint" is used here following its OAuth definition, aimed at denoting resources such as /token and /introspect at the AS, and /authz-info at the RS. This document does not use the CoAP definition of "endpoint", which is "An entity participating in the CoAP protocol".

* The terms and concepts of Pub-Sub group communication with CoAP, as described in [I-D.ietf-core-coap-pubsub].

A party interested in participating in group communication as well as already participating as a group member is interchangeably denoted as "Client", "Pub-Sub client", or "node".

* Group: a set of nodes that share common keying material and security parameters to protect their communications with one another. That is, the term refers to a "security group".

  This is not to be confused with an "application group", which has relevance at the application level and whose members are in this case the Clients acting as Publishers and/or Subscribers for a topic.

Examples throughout this document are expressed in CBOR diagnostic notation as defined in Section 8 of [RFC8949] and Appendix G of [RFC8610]. Diagnostic notation comments are often used to provide a textual representation of the parameters' keys and values.

2. Application Profile Overview

This document describes how to use [RFC9200] and [RFC9594] to perform authentication, authorization, and key distribution operations as overviewed in Section 2 of [RFC9594], where the considered group is the security group composed of the Pub-Sub clients that exchange end-to-end protected content through the Broker.

Pub-Sub clients communicate within their application groups, each of which is mapped to a topic. Depending on the application, a topic may consist of one or more sub-topics, which in turn may have their own sub-topics and so on, thus forming a hierarchy. A security group SHOULD be associated with a single application group. However, the same application group MAY be associated with multiple security groups. Further details and considerations on the mapping between the two types of groups are out of the scope of this document.

This profile considers the architecture shown in Figure 1. A Client can act as a Publisher, or a Subscriber, or both, e.g., by publishing to some topics and subscribing to others. However, for the simplicity of presentation, this profile describes Publisher and Subscriber Clients separately.

Both Publishers and Subscribers act as ACE Clients. The Broker acts as an ACE RS, and corresponds to the Dispatcher in [RFC9594]. The Key Distribution Center (KDC) also acts as an ACE RS, and builds on what is defined for the KDC in [RFC9594]. From a high-level point of view, the Clients interact with the KDC in order to join security groups, thereby obtaining the group keying material to protect end-to-end and verify the content published in the associated topics.

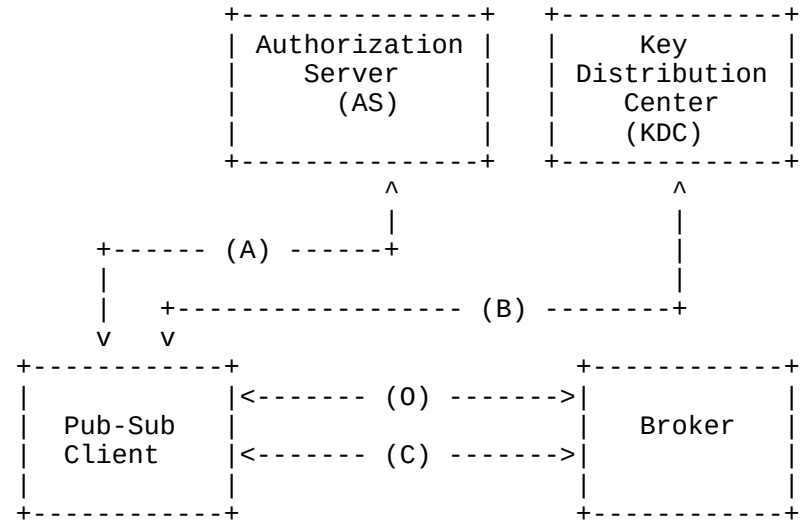

Figure 1: Architecture for Pub-Sub with Authorization Server and Key Distribution Center

Both Publishers and Subscribers MUST use the same protocol for interacting with the Broker and participating in Pub-Sub communications. When using the profile defined in this document, such a protocol MUST be CoAP [RFC7252], which is used as described in [I-D.ietf-core-coap-pubsub]. .

All Publishers and Subscribers MUST use CoAP when communicating with the KDC.

Furthermore, both Publishers and Subscribers MUST use the same transport profile of ACE (e.g., [RFC9202] for DTLS; or [RFC9203] or [I-D.ietf-ace-edhoc-oscore-profile] for OSCORE) in their interaction

with the Broker. In order to reduce the number of libraries that Clients have to support, it is RECOMMENDED that the same transport profile of ACE is used also for the interaction between the Clients and the KDC.

All communications between the involved entities MUST be secured.

For each Client, the Client and the Broker MUST have a secure communication association, which they establish with the help of the AS and using a transport profile of ACE. This is shown by the interactions A and C in Figure 1. During this process, the Client obtains an access token from the AS and uploads it to the Broker, thus providing an evidence of the topics that it is authorised to participate in, and with which permissions.

For each Client, the Client and the KDC MUST have a secure communication association, which they also establish with the help of the AS and using a transport profile of ACE. This is shown by the interactions A and B in Figure 1. During this process, the Client obtains an access token from the AS and uploads it to the KDC, thus providing an evidence of the security groups that it can join, as associated with the topics of interest at the Broker. Based on the permissions specified in the access token, the Client can request the KDC to join a security group, after which the Client obtains from the KDC the keying material to use for communicating with the other group members. This builds on the process for joining security groups with ACE, as defined in [RFC9594] and further specified in this document.

In addition, this profile allows an anonymous Client to perform some of the discovery operations defined in Section 2.3 of [I-D.ietf-core-coap-pubsub] through the Broker, as shown by the interaction O in Figure 1. That is, an anonymous Client can discover:

* the Broker itself, by relying on the resource type "core.ps" (see Section 2.3.1 of [I-D.ietf-core-coap-pubsub]); and

* topics of interest at the Broker (i.e., the corresponding topic resources hosted at the Broker), by relying on the resource type "core.ps.conf" (see Section 2.3.3 of [I-D.ietf-core-coap-pubsub]).

However, an anonymous Client is not allowed to access topic resources at the Broker and obtain from those any additional information or metadata about the corresponding topic (e.g., the topic status, the URI of the topic-data resource where to publish or subscribe for that topic, or the URI to the KDC).

As highlighted in Figure 2, each Client maintains two different security associations pertaining to the Pub-Sub group communication. On the one hand, the Client has a pairwise security association with the Broker, which, as an ACE RS, verifies that the Client is authorised to perform data operations (i.e., publish, subscribe, read, delete) on a certain set of topics (Security Association 1). As discussed above, this security association is set up with the help of the AS and using a transport profile of ACE, when the Client obtains the access token to upload to the Broker.

On the other hand, separately for each topic, all the Publishers and Subscribers for that topic have a common, group security association, through which the published content sent through the Broker is protected end-to-end (Security Association 2). As discussed above, this security association is set up and maintained as the different Clients request the KDC to join the security group, upon which they obtain from the KDC the corresponding group keying material to use for protecting end-to-end and verifying the content of their Pub-Sub group communication.

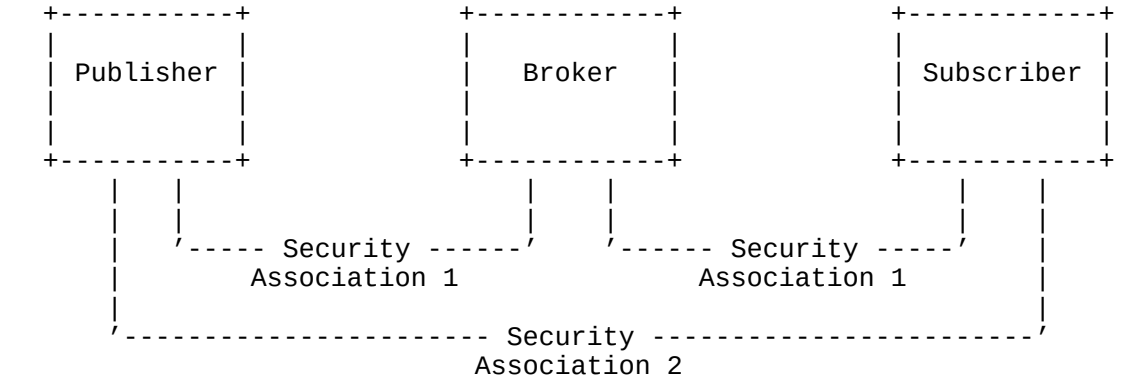

Figure 2: Security Associations between Publisher, Broker, and Subscriber

In summary, this profile specifies the following functionalities.

1. A Client obtains the authorization to participate in a Pub-Sub topic at the Broker with certain permissions. This pertains operations defined in [I-D.ietf-core-coap-pubsub] for taking part in Pub-Sub communication with CoAP.

2. A Client obtains the authorization to join a security group with certain permissions. This allows the Client to obtain from the KDC the group keying material for communicating with other group members, i.e., to protect end-to-end and verify the content published at the Broker on topics associated with the security group.

3. A Client obtains from the KDC the authentication credentials of other group members, and provides or updates the KDC with its own authentication credential.

4. A Client leaves the group or is removed from the group by the KDC.

5. The KDC renews and redistributes the group keying material (rekeying), e.g., due to a membership change in the group.

Appendix A lists the specifications on this application profile of ACE, based on the requirements defined in Appendix A of [RFC9594].

3. Getting Authorisation to Join a Pub-Sub security group (A)

Figure 3 provides a high level overview of the message flow for a Client getting authorisation to join a group. Square brackets denote optional steps. The message flow is expanded in the subsequent sections.

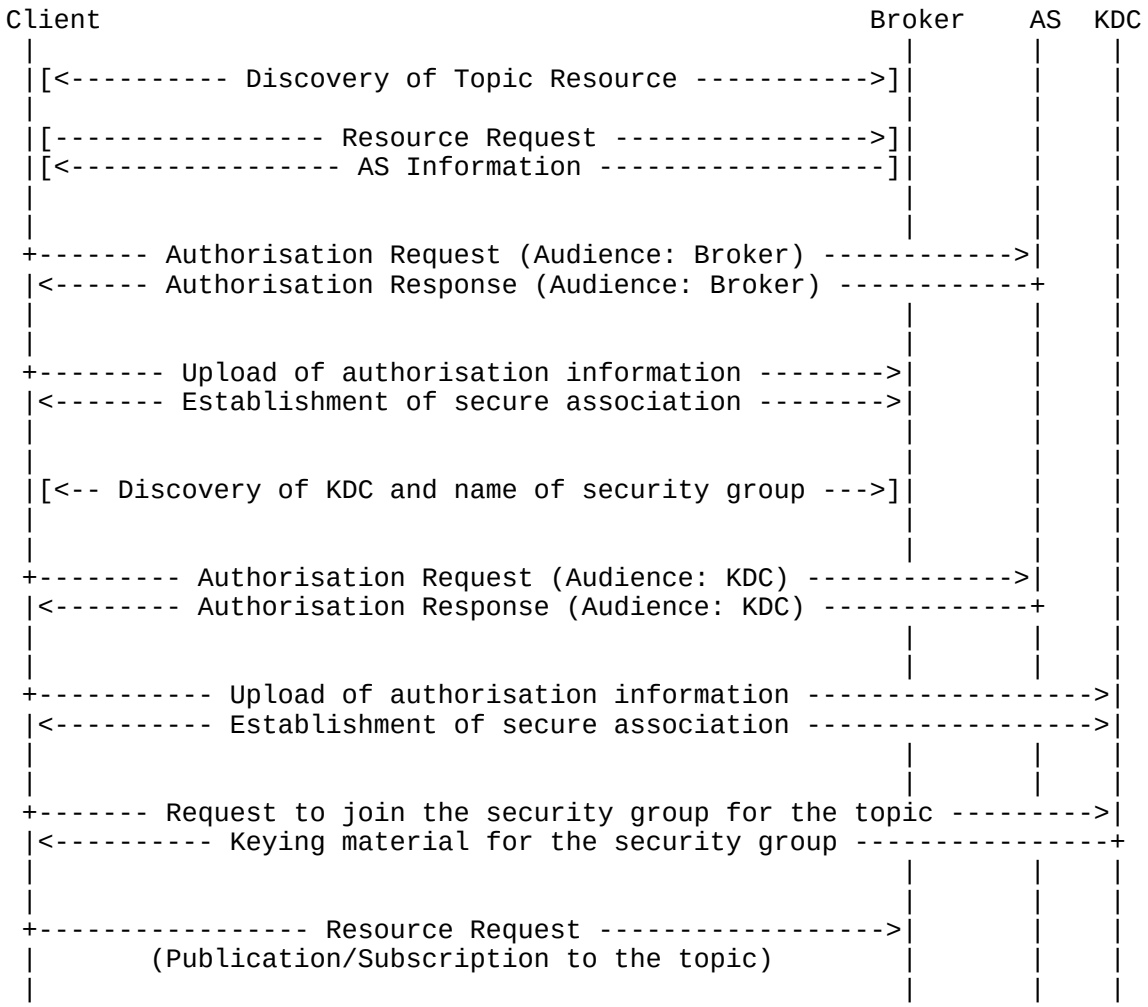

Figure 3: Authorisation Flow

After a Client uploads to the Broker the authorisation information for participating in a Pub-Sub topic with name TOPICNAME (see Section 3.4), the Client can perform the optional discovery of the KDC and security group name at the Broker, by accessing the topic resource corresponding to the topic in question (see Section 3.3).

In order to ensure that the Client can seamlessly access the right topic resource at the Broker, it is RECOMMENDED that a Broker implementing this application profile uses the path /ps/TOPICNAME to host the topic resource for the topic with name TOPICNAME.

Alternatively, the Client might not know the right topic resource to target, and thus would attempt to access different ones (e.g., based on the result of an early discovery of topic resources, see Section 3.1), until it finds the right one specifying TOPICNAME as value of the 'topic-name' parameter in the resource representation.

Since [RFC9200] recommends the use of CoAP and CBOR, this document describes the exchanges assuming that CoAP and CBOR are used.

However, using HTTP instead of CoAP is possible, by leveraging the corresponding parameters and methods. Analogously, JSON [RFC8259] can be used instead of CBOR, by relying on the conversion method specified in Sections 6.1 and 6.2 of [RFC8949]. In case JSON is used, the Content-Format of the message has to be specified accordingly. Exact definitions of these exchanges are out of scope for this document.

3.1. Topic Discovery at the Broker (Optional)

The discovery of a topic at the Broker can be performed by discovering the corresponding topic resource hosted at the Broker. For example, the Client can send a lookup request to /.well-known/core at the Broker, specifying as lookup criterion the resource type "core.ps.conf" (see Section 2.3.3 of [I-D.ietf-core-coap-pubsub]).

Although the links to the topic resources are also specified in the representation of the collection resource at the Broker (see Section 2.4 of [I-D.ietf-core-coap-pubsub]), the Client is not supposed to access such a resource, as intended for administrative operations that are out of the scope of this document.

3.2. AS Discovery at the Broker (Optional)

Complementary to what is defined in Section 5.1 of [RFC9200] for AS discovery, the Broker MAY send the address of the AS to the Client in the 'AS' parameter of the AS Request Creation Hints, as a response to an Unauthorised Resource Request (see Section 5.2 of [RFC9200]). An example using the CBOR diagnostic notation and CoAP is given below.

```
4.01 Unauthorized
Content-Format: application/ace+cbor
Payload:
{
 / AS / 1 : "coaps://as.example.com/token"
}
```

Figure 4: AS Request Creation Hints Example

3.3. KDC Discovery at the Broker (Optional)

Once a Client has obtained an access token from the AS and accordingly established a secure association with the Broker, the Client has the permission to access the topic resources at the Broker that pertain to the topics on which the Client is authorised to operate.

In particular the Client is authorised to retrieve the representation of a topic resource, from which the Client can retrieve information related to the topic in question, as specified in Section 2.5 of [I-D.ietf-core-coap-pubsub].

This profile extends the set of CoAP Pub-Sub Parameters that is possible to specify within the representation of a topic resource, as originally defined in Section 3 of [I-D.ietf-core-coap-pubsub]. In particular, this profile defines the following two parameters that the Broker can specify in a response from a topic resource (see Section 2.5 of [I-D.ietf-core-coap-pubsub]). Note that, when these parameters are transported in their respective fields of the message payload, the Content-Format application/core-pubsub+cbor defined in [I-D.ietf-core-coap-pubsub] MUST be used.

* 'kdc_uri', with value the URI of the group membership resource at the KDC, where Clients can send a request to join the security group associated with the topic in question. The URI is encoded as a CBOR text string. Clients will have to obtain an access token from the AS to upload to the KDC, before starting the process to join the security group at the KDC.

* 'sec_gp', specifying the name of the security group associated with the topic in question, as a stable and invariant identifier. The name of the security group is encoded as a CBOR text string.

Furthermore, the Resource Type (rt=) Link Target Attribute value "core.ps.gm" is registered in Section 8.3 (REQ10), and can be used to describe group-membership resources at the KDC, e.g., by using a link-format document [RFC6690]. As an alternative to the discovery approach defined above and provided by the Broker, applications can use this common resource type to discover links to group-membership resources at the KDC for joining security groups associated with Pub-Sub topics.

3.4. Authorisation Request/Response for the KDC and the Broker

A Client sends two Authorisation Requests to the AS, targeting two different audiences, i.e., the Broker and the KDC.

As to the former, the AS handles Authorisation Requests related to a topic for which the Client is allowed to perform topic data operations at the Broker, as corresponding to an application group.

As to the latter, the AS handles Authorization Requests for security groups that the Client is allowed to join, in order to obtain the group keying material for protecting end-to-end and verifying the content of exchanged messages on the associated Pub-Sub topics.

This section builds on Section 3 of [RFC9594] and defines only additions or modifications to that specification.

Both Authorisation Requests include the following fields (see Section 3.1 of [RFC9594]):

* 'scope': Optional. If present, it specifies the following information, depending on the specifically targeted audience.

  If the audience is the Broker, the scope specifies the name of the topics that the Client wishes to access, together with the corresponding requested permissions.

  If the audience is the KDC, the scope specifies the name of the security groups that the Client wishes to join, together with the corresponding requested permissions.

  This parameter is encoded as a CBOR byte string, whose value is the binary encoding of a CBOR array. The format of the encoded scope MUST follow the data model AIF-PUBSUB-GROUPCOMM defined in Section 3.4.1.

* 'audience': Required identifier corresponding to either the Broker or the KDC.

Other additional parameters can be included if necessary, as defined in [RFC9200].

When using this profile, it is expected that a one-to-one mapping is enforced between the application group and the security group (see Section 2). If this is not the case, the correct access policies for both sets of scopes have to be available to the AS.

3.4.1. Format of Scope

Building on Section 3.1 of [RFC9594], this section defines the exact format and encoding of scope used in this profile.

To this end, this profile uses the Authorization Information Format (AIF) [RFC9237] (REQ1). With reference to the generic AIF model

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

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

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

In particular, the following holds for each scope entry.

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

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

More specifically, the following applies when, as defined in this document, a scope entry includes a set of permissions for user-related operations performed by a pubsub Client.

* The object identifier ("Toid") is a CBOR text string, specifying the name of one application group (topic) or of the corresponding security group to which the scope entry pertains.

* The permission set ("Tperm") is a CBOR unsigned integer, whose value R specifies the operations that the Client wishes to or has been authorised to perform on the resources at the Broker associated with the application group (topic) indicated by "Toid", or on the resources at the KDC associated with the security group indicated by "Toid" (REQ1). The value R is computed as follows.

  - Each operation (i.e., permission detail) in the permission set is converted into the corresponding numeric identifier X taken from the following set.

    o Admin (0): This operation is reserved for scope entries that express permissions for Administrators of Pub-Sub groups, which are not specified in this document.

- AppGroup (1): This operation is signaled as wished/ authorised when "Toid" specifies the name of an application group (topic), while it is signaled as not wished/authorised when Toid specifies the name of a security group.

- Publish (2): This operation concerns the publication of data to the topic in question, performed by means of a PUT request sent by a Publisher Client to the corresponding topic-data resource at the Broker.

- Read (3): This operation concerns both: i) the subscription at the topic-data resource for the topic in question at the Broker, performed by means of a GET request with the CoAP Observe Option set to 0 and sent by a Subscriber Client; and ii) the simple reading of the latest data published to the topic in question, performed by means of a simple GET request sent to the same topic-data resource.

- Delete (4): This operation concerns the deletion of the topic-data resource for the topic in question at the Broker, performed by means of a DELETE request sent to that resource.

  If a Client wishes to have authorised only the Delete operation on an application group, then the Client does not need to join the corresponding security group, hence it does not need to request an access token for interacting with the KDC.

  If a Client wishes to have authorised the Delete operation on a security group, then the AS and the KDC ignore the wish for that operation when processing the scope entry in question.

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

Since this application profile considers user-related operations, the "Admin" operation is signaled as not wished/authorised. That is, the scope entries MUST have the least significant bit of "Tperm" set to 0.

If the "Toid" of a scope entry in an access token specifies the name of an application group (i.e., the "AppGroup" operation is signaled as authorised), the Client has also the permission to retrieve the configuration of the application group (topic) whose name is indicated by "Toid", by sending a GET or FETCH request to the corresponding topic resource at the Broker.

The specific interactions between the Client and the Broker are defined in [I-D.ietf-core-coap-pubsub].

The following CDDL [RFC8610] notation defines a scope entry that uses the AIF-PUBSUB-GROUPCOMM data model and expresses a set of permissions.

```
AIF-PUBSUB-GROUPCOMM = AIF-Generic<pubsub-group, pubsub-perm>
 pubsub-group = tstr ; name of Pub-Sub topic or of
                     ; the associated security group

 pubsub-perm = uint .bits pubsub-perm-details

 pubsub-perm-details = &(
  Admin: 0,
  AppGroup: 1,
  Publish: 2,
  Read: 3,
  Delete: 4
 )

 scope_entry = [pubsub-group, pubsub-perm]
```

Figure 5: Pub-Sub scope using the AIF format

3.5. Authorization response

The AS responds with an Authorization Response as defined in Section 5.8.2 of [RFC9200] and Section 3.2 of [RFC9594], with the following additions:

* In the Authorization Response, the AS MUST include the 'expires_in' parameter. Other means for the AS to specify the lifetime of access tokens are out of the scope of this document.

* In the Authorization Response, the AS MUST include the 'scope' parameter, when the value included in the access token differs from the one specified by the Client in the Authorization Request. In such a case, the second element of each scope entry specifies the set of operations that the Client is authorised for that scope entry, encoded as specified in Section 3.4.

Furthermore, the AS MAY use the extended format of scope defined in Section 7 of [RFC9594] for the ’scope’ claim of the access token. In such a case, the AS MUST use the CBOR tag with tag number TAG_NUMBER, associated with the CoAP Content-Format CF_ID for the media type application/aif+cbor registered in Section 8.5 of this document (REQ28).

Note to RFC Editor: In the previous paragraph, please replace "TAG_NUMBER" with the CBOR tag number computed as TN(ct) in Section 4.3 of [RFC9277], where ct is the ID assigned to the CoAP Content-Format registered in Section 8.5 of this document. Then, please replace "CF_ID" with the ID assigned to that CoAP Content-Format. Finally, please delete this paragraph.

This indicates that the binary encoded scope follows the scope semantics defined for this application profile in Section 3.4.1 of this document.

3.6. Token Transfer to the KDC

The Client transfers its access token to the KDC using one of the methods defined in the Section 3.3 of [RFC9594]. This typically includes sending a POST request to the /authz-info endpoint. However, if the DTLS transport profile of ACE [RFC9202] is used and the Client uses a symmetric proof-of-possession key in the DTLS handshake, the Client MAY provide the access token to the KDC in the "psk_identity" field of the DTLS ClientKeyExchange message when using DTLS 1.2 [RFC6347], or in the "identity" field of a PskIdentity within the PreSharedKeyExtension of the ClientHello message when using DTLS 1.3 [RFC9147]. In addition to that, the following applies.

In the Token Transfer Response to the Publishers, i.e., the Clients whose scope of the access token includes the "Publish" permission for at least one scope entry, the KDC MUST include the parameter ’kdcchallenge’ in the CBOR map. ’kdcchallenge’ is a challenge N_S generated by the KDC, and is RECOMMENDED to be an 8-byte long random nonce. Later when joining the group, the Publisher can use the ’kdcchallenge’ as part of proving possession of its private key (see [RFC9594]). If a Publisher provides the access token to the KDC through an /authz-info endpoint, the Client MUST support the parameter ’kdcchallenge’.

If ’sign_info’ is included in the Token Transfer Request, the KDC SHOULD include the ’sign_info’ parameter in the Token Transfer Response. Note that the joining node may have obtained such information by alternative means, e.g., the ’sign_info’ may have been pre-configured (OPT3).

The following applies for each element 'sign_info_entry'.

* 'sign_alg' MUST take its value from the "Value" column of one of the recommended algorithms in the "COSE Algorithms" registry [IANA.cose_algorithms] (REQ3).

* 'sign_parameters' is a CBOR array. Its format and value are the same of the COSE capabilities array for the algorithm indicated in 'sign_alg' under the "Capabilities" column of the "COSE Algorithms" registry [IANA.cose_algorithms] (REQ4).

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

* 'cred_fmt' takes value from the "Label" column of the "COSE Header Parameters" registry [IANA.cose_header-parameters] (REQ6). Acceptable values denote a format of authentication credential that MUST explicitly provide the public key as well as the comprehensive set of information related to the public key algorithm, including, e.g., the used elliptic curve (when applicable).

  Acceptable formats of authentication credentials include CBOR Web Tokens (CWTs) and CWT Claims Sets (CCSs) [RFC8392], X.509 certificates [RFC7925], and C509 certificates [I-D.ietf-cose-cbor-encoded-cert]. Future formats would be acceptable to use as long as they comply with the criteria defined above.

4. Client Group Communication Interface at the KDC

In order to enable secure group communication for the Pub-Sub Clients, the KDC provides the resources listed in Table 1. Each resource is marked as REQUIRED or OPTIONAL to be hosted at the KDC.

KDC resource	Description	Operations
/ace-group	REQUIRED. Contains a set of group names, each corresponding to one of the specified	FETCH (All Clients)

	group identifiers.	
/ace-group/GROUPNAME	REQUIRED. Contains symmetric group keying material associated with GROUPNAME.	GET, POST (All Clients)
/ace-group/GROUPNAME/creds	REQUIRED. Contains the authentication credentials of all the Publishers of the group with name GROUPNAME.	GET, FETCH (All Clients)
/ace-group/GROUPNAME/num	REQUIRED. Contains the current version number for the symmetric group keying material of the group with name GROUPNAME.	GET (All Clients)
/ace-group/GROUPNAME/nodes/ NODENAME	REQUIRED. Contains the group keying material for that group member NODENAME in GROUPNAME.	GET, DELETE (All Clients). POST (Publishers).
/ace-group/GROUPNAME/nodes/NODENAME/cred	REQUIRED. Authentication credential for NODENAME in the group	POST (Publishers)

	GROUPNAME.	
/ace-group/GROUPNAME/kdc-cred	REQUIRED if a group re-keying mechanism is used. Contains the authentication credential of the KDC for the group with name GROUPNAME.	GET (All Clients)
/ace-group/GROUPNAME/policies	OPTIONAL. Contains the group policies of the group with name GROUPNAME.	GET (All Clients)

Table 1: Resources at the KDC

The use of these resources follows what is defined in [RFC9594], and only additions or modifications to that specification are defined in this document.

Consistent with what is defined in Section 4.1.2 of [RFC9594], some error responses from the KDC can convey error-specific information according to the problem-details format specified in [RFC9290].

4.1. Joining a Security Group

This section describes the interactions between a Client and the KDC to join a security group. Source authentication of a message sent within the group is ensured by means of a digital signature embedded in the message. Subscribers must be able to retrieve Publishers' authentication credentials from a trusted repository, to verify source authentication of received messages. Hence, on joining a security group, a Publisher is expected to provide its own authentication credential to the KDC.

On a successful join, the Clients receive from the KDC the symmetric COSE Key used as shared group key to protect the payload of a published topic data.

The message exchange between the joining node and the KDC follows what is defined in Section 4.3.1.1 of [RFC9594], and only additions or modifications to that specification are defined in this document.

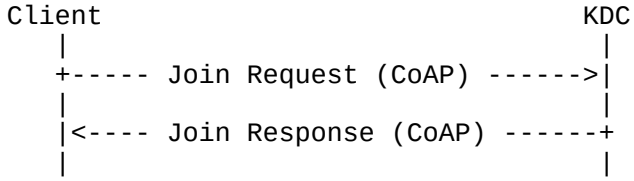

Figure 6: Join Flow

4.1.1. Join Request

After establishing a secure communication association with the KDC, the Client sends a Join Request to the KDC as described in Section 4.3 of [RFC9594]. More specifically, the Client sends a POST request to the /ace-group/GROUPNAME endpoint, with Content-Format "application/ace-groupcomm+cbor". The payload contains the following information formatted as a CBOR map, which MUST be encoded as defined in Section 4.3.1 of [RFC9594]:

* 'scope': It MUST be present and specify the group that the Client is attempting to join, i.e., the group name, and the permissions that the Client wishes to have in the group. This value corresponds to one scope entry, as defined in Section 3.4.1.

* 'get_creds': It MAY be present if the Client wishes to join as a Subscriber and wants to retrieve the public keys of all the Publishers upon joining. Otherwise, this parameter MUST NOT be present. If the parameter is present, the parameter MUST encode the CBOR simple value null (0xf6). Note that the parameter 'role_filter' is not necessary, as the KDC returns the authentication credentials of Publishers by default.

* 'client_cred': The use of this parameter is detailed in Section 4.1.1.1.

* 'cnonce': It specifies a dedicated nonce N_C generated by the Client. It is RECOMMENDED to use an 8-byte long random nonce. Join Requests MUST include a new 'cnonce' at each join attempt.

* 'client_cred_verify': The use of this parameter is detailed in Section 4.1.1.2.

As a Publisher Client has its own authentication credential to use in a group, it MUST support the client_cred', 'cnonce', and 'client_cred_verify' parameters.

4.1.1.1. Client Credential in 'client_cred'

One of the following cases can occur when a new Client attempts to join a security group.

* The joining node is not a Publisher, i.e., it is not going to send data to the application group. In this case, the joining node is not required to provide its own authentication credential to the KDC. In case the joining node still provides an authentication credential in the 'client_cred' parameter of the Join Request (see Section 4.1.1), the KDC silently ignores that parameter, as well as the related parameter 'client_cred_verify'.

* The joining node wishes to join as a Publisher, and the KDC has not previously acquired an authentication credential of the joining node. Then, the joining node MUST provide a compatible authentication credential in the 'client_cred' parameter of the Join Request (see Section 4.1.1).

* The joining node wishes to join as a Publisher, and the KDC already acquired the authentication credential of the joining node either during a past group joining process, or when establishing a secure communication association using asymmetric proof-of-possession keys.

  If the joining node's proof-of-possession key is compatible with the signature algorithm used in the security group and with possible associated parameters, then the corresponding authentication credential can be used in the group. In this case, the joining node MAY choose not to provide again its authentication credential to the KDC in order to limit the size of the Join Request.

The joining node MUST provide the KDC with its own authentication credential again, if it has previously provided the KDC with multiple authentication credentials intended for different security groups.

If the joining node provides its authentication credential, the KDC performs the consistency checks above and, in case of success, considers it as the authentication credential associated with the joining node in the group.

4.1.1.2. Proof-of-Possession through 'client_cred_verify'

The 'client_cred_verify' parameter contains the proof-of-possession evidence, and is computed as defined below (REQ14).

The Publisher signs the scope, concatenated with N_S and further concatenated with N_C, by using the private key corresponding to the public key that is specified in the 'client_cred' parameter.

The N_S may be either of the following:

* The challenge received from the KDC in the 'kdcchallenge' parameter of the 2.01 (Created) response to the Token Transfer Request (see Section 3.6).

* If the provisioning of the access token to the KDC has relied on the DTLS profile of ACE [RFC9202], and the access token was specified in the "psk_identity" field of the ClientKeyExchange message when using DTLS 1.2 [RFC6347], then N_S is an exporter value computed as defined in Section 4 of [RFC5705] (REQ15).

  Specifically, N_S is exported from the DTLS session between the joining node and the KDC, using an empty context value (i.e., a context value of zero-length), 32 as length value in bytes, and the exporter label "EXPORTER-ACE-Sign-Challenge-pubsub-app" defined in Section 8.6 of this document.

* If the provisioning of the access token to the KDC has relied on the DTLS profile of ACE [RFC9202], and the access token was specified in the "identity" field of a PskIdentity within the PreSharedKeyExtension of the ClientHello message when using DTLS 1.3 [RFC9147], then N_S is an exporter value computed as defined in Section 7.5 of [RFC8446] (REQ15).

  Specifically, N_S is exported from the DTLS session between the joining node and the KDC, using an empty 'context_value' (i.e., a 'context_value' of zero length), 32 as 'key_length' in bytes, and the exporter label "EXPORTER-ACE-Sign-Challenge-pubsub-app" defined in Section 8.6 of this document.

* If the Join Request is a retry in response to an error response from the KDC, which included a new 'kdcchallenge' parameter, then N_S MUST be the new value from this parameter.

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

4.1.2. Join Response

Upon receiving the Join Request, the KDC processes it as defined in Section 4.3.1 of [RFC9594], and returns a success or error response.

If the 'client_cred' parameter is present, the KDC verifies the signature in the 'client_cred_verify' parameter. As PoP input, the KDC uses the value of the 'scope' parameter from the Join Request as a CBOR byte string, concatenated with N_S encoded as a CBOR byte string, concatenated with N_C encoded as a CBOR byte string.

As public key of the joining node, the KDC uses either the one included in the authentication credential retrieved from the 'client_cred' parameter of the Join Request, or the one from the already stored authentication credential from previous interactions with the joining node. The KDC verifies the signature used as PoP evidence by means of the public key of the joining node, according to the signature algorithm used in the group and possible corresponding parameters.

In case of any Join Request error, the KDC and the joining node follow the procedure defined in Section 4.1.3.

In case of success, the KDC responds with a Join Response, whose payload is formatted as a CBOR map and MUST contain the following fields as per Section 4.3.1 of [RFC9594]:

* 'gkty': this field specifies the key type "Group_PubSub_Keying_Material" (REQ18) registered in Section 8.1 for the 'key' parameter.

* 'key': this field specifies the keying material to use for secure communication in the group (REQ17). This field has as value a CBOR map that includes the following parameters.

  - 'group_key': this parameter is identified by the CBOR unsigned integer 0 used as map key. Its value is a COSE_Key object as defined in [RFC9052] and conveying the group key to use in the security group.

    The COSE_Key object MUST contain the following parameters:

    o 'kty', with value 4 (Symmetric).

    o 'alg', with value the identifier of the AEAD algorithm used in the security group. The value is taken from the "Value" column of the "COSE Algorithms" registry [IANA.cose_algorithms].

  - o ’Base IV’, with value the Base Initialization Vector (Base IV) to use in the security group with this group key.

  - o ’k’, with value the symmetric encryption key to use as group key.

  - o ’kid’, with value the identifier of the COSE_Key object, hence of the group key.

    This value is used as Group Identifier (Gid) of the security group, as long as the present key is used as group key in the security group.

- ’group_SenderId’: this parameter is identified by the CBOR unsigned integer 1 used as map key. Its value is the Client’s Sender ID encoded as a CBOR byte string. This parameter MUST be included if the Client is joining the security group as a Publisher, and MUST NOT be included otherwise. A Publisher Client MUST support the ’group_SenderId’ parameter (REQ29).

  The Sender ID MUST be unique within the security group. The KDC MUST only assign an available Sender ID that has not been used in the security group since the last time when the current Gid value was assigned to the group (i.e., since the latest group rekeying, see Section 5). The KDC MUST NOT assign a Sender ID to the joining node if the node is not joining the group as a Publisher.

  The Sender ID can be short in length. Its maximum length in bytes is the length in bytes of the AEAD nonce for the AEAD algorithm, minus 6. This means that, when using AES-CCM-16-64-128 as AEAD algorithm in the security group, the maximum length of Sender IDs is 7 bytes.

- ’cred_fmt’: this parameter is identified by the CBOR unsigned integer 2 used as map key. Its value specifies the format of authentication credentials used in the group, and is taken from the "Label" column of the "COSE Header Parameters" registry [IANA.cose_header-parameters].

  At the time of writing this specification, acceptable formats of authentication credentials are CBOR Web Tokens (CWTs) and CWT Claims Sets (CCSs) [RFC8392], X.509 certificates [RFC7925], and C509 certificates [I-D.ietf-cose-cbor-encoded-cert]. Further formats may be available in the future, and would be acceptable to use as long as they comply with the criteria defined above (REQ6).

- 'sign_alg': this parameter is identified by the CBOR unsigned integer 3 used as map key. Its value specifies the Signature Algorithm used to sign messages in the group, and is taken from the "Value" column of the "COSE Algorithms" registry [IANA.cose_algorithms].

- 'sign_params': this parameter is identified by the CBOR unsigned integer 4 used as map key. Its value specifies the parameters of the Signature Algorithm, and is encoded as a CBOR array including the following two elements:

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

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

* 'num', specifying the version number of the keying material specified in the 'key' field. The initial value of the version number MUST be set to 0 upon creating the group (REQ16).

* 'exi', which MUST be present.

* 'ace-groupcomm-profile', which MUST be present and has value "coap_group_pubsub_app" (PROFILE_TBD), which is registered in Section 8.2 (REQ19).

* 'creds', which MUST be present if the 'get_creds' parameter was present in the Join Request, and MUST NOT be present otherwise. It specifies the authentication credentials of all the Publishers in the security group.

* 'peer_roles', which MAY be omitted even if 'creds' is present, since: i) each authentication credential conveyed in the 'creds' parameter is associated with a Client authorised to be Publisher in the group; and ii) it is irrelevant whether such a Client is also authorised to be Subscriber in the group. If 'creds' is not present, 'peer_roles' MUST NOT be present.

* 'peer_identifiers', which MUST be present if 'creds' is also present, and MUST NOT be present otherwise. The identifiers are the Publisher Sender IDs, whose corresponding authentication credentials are specified in the 'creds' parameter (REQ25).

* 'kdc_cred', which MUST be present if group re-keying is used. It is encoded as a CBOR byte string, with value the original binary representation of the KDC's authentication credential (REQ8).

* 'kdc_nonce', which MUST be present if 'kdc_cred' is present. It is encoded as a CBOR byte string, with value a dedicated nonce N_KDC generated by the KDC. For N_KDC, it is RECOMMENDED to use an 8-byte long random nonce.

* 'kdc_cred_verify', which MUST be present if 'kdc_cred' is present. It is encoded as a CBOR byte string. The KDC MUST compute the specified PoP evidence as a signature by using the signature algorithm used in the group, as well as its own private key associated with the authentication credential specified in the 'kdc_cred' parameter (REQ21).

* 'group_rekeying', which MAY be omitted if the KDC uses the "Point-to-Point" group rekeying scheme registered in Section 11.13 of [RFC9594] as the default rekeying scheme in the group (OPT9). In any other case, the 'group_rekeying' parameter MUST be included.

After sending a successful Join Response, the KDC adds the Client to the list of current members of the security group, if that Client is not already a group member. Also, the Client is assigned a name NODENAME and a sub-resource /ace-group/GROUPNAME/nodes/NODENAME at the KDC. Furthermore, the KDC associates NODENAME with the Client's access token and with the secure communication association that the KDC has with the Client. If the Client is a Publisher, its authentication credential is also associated with the tuple containing NODENAME, GROUPNAME, the current Gid, the newly assigned Publisher's Sender ID, and the Client's access token. The KDC MUST keep this association updated over time.

Note that, as long as the secure communication association between the Client and the KDC persists, the same Client re-joining the group is recognized by the KDC by virtue of such a secure communication association. As a consequence, the re-joining Client keeps the same NODENAME and the associated subresource /ace-group/GROUPNAME/nodes/NODENAME. Also, if the Client is a Publisher, it receives a new Sender ID according to the same criteria defined above.

If the application requires backward security, the KDC 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 5). In such a case, the joining node is not able to access secure communication in the Pub-Sub group prior its joining.

Upon receiving the Join Response, the joining node retrieves the KDC's authentication credential from the 'kdc_cred' parameter (if present). The joining node MUST verify the signature used as proof-of-possession (PoP) evidence, which is specified by the 'kdc_cred_verify' parameter of the Join Response (REQ21).

4.1.3. Join Error Handling

The KDC MUST reply with a 4.00 (Bad Request) error response (OPT4) to the Join Request in the following cases:

* The 'client_cred' parameter is present in the Join Request and its value is not an eligible authentication credential (e.g., it is not of the format accepted in the group) (OPT8).

* The 'client_cred' parameter is present, but the 'cnonce' and 'client_cred_verify' parameters are not present.

* The 'client_cred' parameter is not present while the joining node is not requesting to join the group exclusively as a Subscriber, and any of the following conditions holds:

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

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

* The 'scope' parameter is not present in the Join Request, or it is present and specifies any set of permissions not included in the list defined in Section 3.4.1.

A 4.00 (Bad Request) error response from the KDC to the joining node MAY have content format application/ace-groupcomm+cbor and contain a CBOR map as payload.

The CBOR map MAY include the 'kdcchallenge' parameter. If present, this parameter is a CBOR byte string, with value a newly generated 'kdcchallenge' value that the Client can use when preparing a new Join Request. In such a case, the KDC MUST store the newly generated value as the 'kdcchallenge' value associated with the joining node, which replaces the currently stored value (if any).

Upon receiving the Join Response, if 'kdc_cred' is present but the Client cannot verify the PoP evidence, the Client MUST stop processing the Join Response and MAY send a new Join Request to the KDC.

The KDC MUST return a 5.03 (Service Unavailable) response to a Client that sends a Join Request to join the security group as Publisher, in case there are currently no Sender IDs available to assign. The response MUST have Content-Format set to application/concise-problem-details+cbor and is formatted as defined in Section 4.1.2 of [RFC9594]. Within the Custom Problem Detail entry 'ace-groupcomm-error', the value of the 'error-id' field MUST be set to 4 ("No available individual keying material").

4.2. Other Group Operations through the KDC

4.2.1. Obtaining Latest Information on the Group, Group Keying Material, and Sender ID

A Client can access the following resources at the KDC, in order to retrieve latest information about the group or the group keying material.

* '/ace-group': All Clients can send a FETCH request to retrieve a set of group names associated with their group identifiers specified in the request payload. Each element of the CBOR array 'gid' is a CBOR byte string (REQ13), which encodes the Gid of the group (see Section 4.1.2) for which the group name and the URI to the group-membership resource are provided in the returned response.

* '/ace-group/GROUPNAME': All group member Clients can send a GET request to retrieve the symmetric group keying material of the group with the name GROUPNAME. The value of the GROUPNAME URI path and the group name in the access token scope ('gname') MUST coincide.

  The KDC processes the Key Distribution Request according to Section 4.3.2 of [RFC9594]. The Key Distribution Response is formatted as defined in Section 4.3.2 of [RFC9594], with the following additions.

- The 'key' field is formatted as defined in Section 4.1.2 of this document, with the difference that it does not include the 'group_SenderId' parameter.

- The 'exi' field MUST be present.

- The 'ace_groupcomm_profile' field MUST be present and has value "coap_group_pubsub_app".

Upon receiving the Key Distribution Response, the requesting group member retrieves the updated security parameters and group keying material. If they differ from the currently stored ones, then the group member uses the received one as group keying material to protect/unprotect published topic data thereafter.

* '/ace-group/GROUPNAME/creds': The KDC acts as a repository of authentication credentials for the Publishers that are member of the security group with name GROUPNAME. The members of the group that are Subscribers can send GET/FETCH requests to this resource in order to retrieve the authentication credentials of all or a subset of the group members that are Publishers. The KDC silently ignores the Sender IDs included in the 'get_creds' parameter of the request that are not associated with any current Publisher group member (REQ26).

  The response from the KDC MAY omit the parameter 'peer_roles', since: i) each authentication credential conveyed in the 'creds' parameter is associated with a Client authorised to be Publisher in the group; and ii) it is irrelevant whether such a Client is also authorised to be Subscriber in the group. If 'creds' is not present, 'peer_roles' MUST NOT be present.

* '/ace-group/GROUPNAME/num': All group member Clients can send a GET request to this resource in order to retrieve the current version number for the symmetric group keying material of the group with name GROUPNAME.

* '/ace-group/GROUPNAME/kdc-cred': All group member Clients can send a GET request to this resource in order to retrieve the current authentication credential of the KDC.

* '/ace-group/GROUPNAME/nodes/NODENAME': A group member can send a Key Distribution to the KDC by sending a GET request to this resource to retrieve the latest group keying material as well as its Sender ID that it has in the group (if Publisher).

The KDC processes the Key Distribution Request according to Section 4.8.1 of [RFC9594]. The Key Distribution Response is formatted as defined in Section 4.8.1 of [RFC9594], with the following additions.

- The 'key' field is formatted as defined in Section 4.1.2 of this document. If the requesting group member is not a Publisher Client, then the 'key' field does not include the 'group_SenderId' parameter.

- The 'exi' field MUST be present.

Upon receiving the Key Distribution Response, the group member retrieves the updated security parameters, group keying material, and Sender ID (if the 'key' field includes the 'group_SenderId' parameter). If they differ from the currently stored ones, then the group member uses the received one as group keying material to protect/unprotect published topic data thereafter.

4.2.2. Requesting a New Sender ID

A Publisher group member with node name NODENAME may at some point exhaust its Sender Sequence Numbers used for protecting its published topic data (see Section 6.1).

When this happens, the Publisher MUST send a Key Renewal Request message to the KDC, as per Section 4.8.2.1 of [RFC9594]. That is, it sends a CoAP POST request to the endpoint /ace-group/GROUPNAME/nodes/NODENAME at the KDC.

Upon receiving the Key Renewal Request, the KDC processes it as defined in Section 4.8.2 of [RFC9594], with the addition that the KDC takes one of the following actions.

* The KDC rekeys the group. That is, the KDC generates new group keying material for that group (see Section 5), and replies to the Publisher with a group rekeying message as defined in Section 5, providing the new group keying material. Then, the KDC rekeys the rest of the group, as discussed in Section 5.

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

* The KDC determines and assigns a new Sender ID for the Publisher, and replies with a Key Renewal Response formatted as defined in Section 4.8.2 of [RFC9594]. The CBOR Map in the response payload includes only the parameter 'group_SenderId' registered in Section 16.3 of [I-D.ietf-ace-key-groupcomm-oscore], and specifies the new Sender ID of the Publisher encoded as a CBOR byte string.

  The KDC MUST assign a new Sender ID that has not been used in the group since the latest time when the current Gid value was assigned to the group (i.e., since the latest group rekeying, see Section 5).

  The KDC MUST return a 5.03 (Service Unavailable) response in case there are currently no Sender IDs available to assign in the group. The response MUST have Content-Format set to application/concise-problem-details+cbor and is formatted as defined in Section 4.1.2 of [RFC9594]. Within the Custom Problem Detail entry 'ace-groupcomm-error', the value of the 'error-id' field MUST be set to 4 ("No available individual keying material").

4.2.3. Updating Authentication Credentials

A Publisher group member with node name NODENAME can contact the KDC to upload a new authentication credential to use in the security group with name GROUPNAME, and replace the currently stored one.

To this end, the Publisher sends a CoAP POST request to its associated sub-resource /ace-group/GROUPNAME/nodes/NODENAME/cred at the KDC (see Section 4.9.1 of [RFC9594]).

Following a successful processing of the request, the KDC replaces the stored authentication credential of this Client for the group GROUPNAME with the one specified in the request.

4.2.4. Leaving a Group

A group member with node name NODENAME can actively request to leave the security group with name GROUPNAME.

To this end, the Client sends a CoAP DELETE request to the associated sub-resource /ace-group/GROUPNAME/nodes/NODENAME at the KDC (see Section 4.8.3 of [RFC9594]).

The KDC can also remove a group member due to any of the reasons described in Section 5 of [RFC9594].

5. Group Rekeying Process

Rekeying a group consists in the KDC generating and distributing a new symmetric key, which is used as group key from then on to protect the publication of topic data with COSE (see Section 6.1).

The KDC MUST trigger a group rekeying as described in Section 6 of [RFC9594], upon a change in the group membership or due to the current group keying material approaching its expiration time. In addition, the KDC MAY perform regularly scheduled group rekeying executions.

Upon generating the new group key and before starting its distribution:

* The KDC MUST increment the version number of the group keying material.

* The KDC MUST generate a new Group Identifier (Gid) for the group. This is used as identifier of the new group key, when providing it to the current group members through the group rekeying, and to Clients (re-)joining the security group thereafter (see Section 4.1.2).

  That is, the value of the new Gid is specified by the 'kid' parameter of the COSE_Key Object that is used to encode the new group key.

When rekeying the group, the KDC MUST preserve the current value of the Sender ID of each member in that group.

The default rekeying scheme is "Point-to-Point" (see Section 6.1 of [RFC9594]), where the KDC individually targets each node to rekey, using the pairwise secure communication association with that node.

In particular, a group rekeying message MUST have Content-Format set to application/ace-groupcomm+cbor and have the same format used for the Join Response message defined in Section 4.1.2, with the following differences:

* Within the 'key' field, only the parameter 'group_key' is present.

* The fields 'kdc_cred' ,'kdc_nonce', 'kdc_cred_verify', and 'group_rekeying' are not present.

* The fields 'creds' and 'peer_identifiers' SHOULD be present, if the group rekeying is performed due to one or multiple Clients joining the group as Publishers. Following the same semantics

used in the Join Response message, the two parameters specify the authentication credentials and Sender IDs of such Clients. Like in the Join Response message, the 'peer_roles' parameter MAY be omitted.

6. Pub-Sub Protected Communication

In the diagram shown in Figure 7, (D) corresponds to the publication on a topic at the Broker, by using a CoAP PUT request. The Publisher protects the published topic data end-to-end for the Subscribers by using COSE ([RFC9052][RFC9053][RFC9338]), as detailed in Section 6.1.

In the same diagram, (E) corresponds to the subscription of a Subscriber to the same topic, by means of a CoAP GET request with the Observe option set to 0 (register) [RFC7641], as per [I-D.ietf-core-coap-pubsub]. Finally, (F) corresponds to the Observe notification response from the Broker to the Subscriber, where the published topic data is conveyed as originally protected end-to-end with COSE by the Publisher.

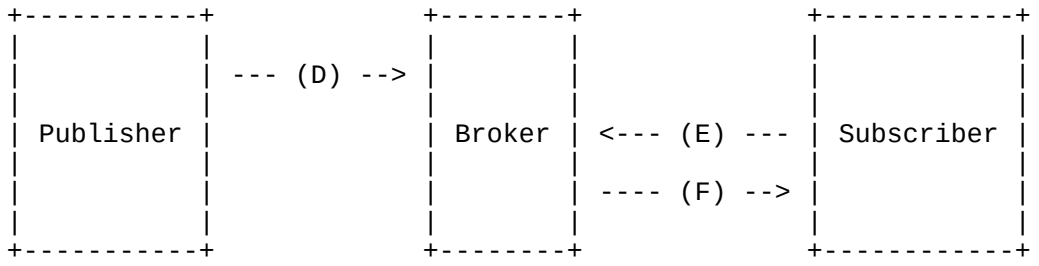

Figure 7: Secure Pub-Sub Communication between Publisher and Subscriber

Figure 8 provides a more detailed example of such a secure Pub-Sub communication. All the messages exchanged between a Client and the Broker are protected with the secure communication association between that Client and the Broker. In addition, COSE is used to protect end-to-end the published topic data, which is conveyed in a PUT request from the Publisher to the topic-data resource at the Broker and in a 2.05 (Content) response from that resource to a Subscriber.

The example also shows a delete operation, where the Publisher deletes the topic-data resource by sending a CoAP DELETE request to the URI of that resource. In case of success, the Broker replies with a 2.02 (Deleted) response. Consequently, the Broker also unsubscribes all the Clients subscribed to that topic-data resource, by removing them from the list of observers and sending them a final 4.04 (Not Found) response as per Section 3.2 of [RFC7641].

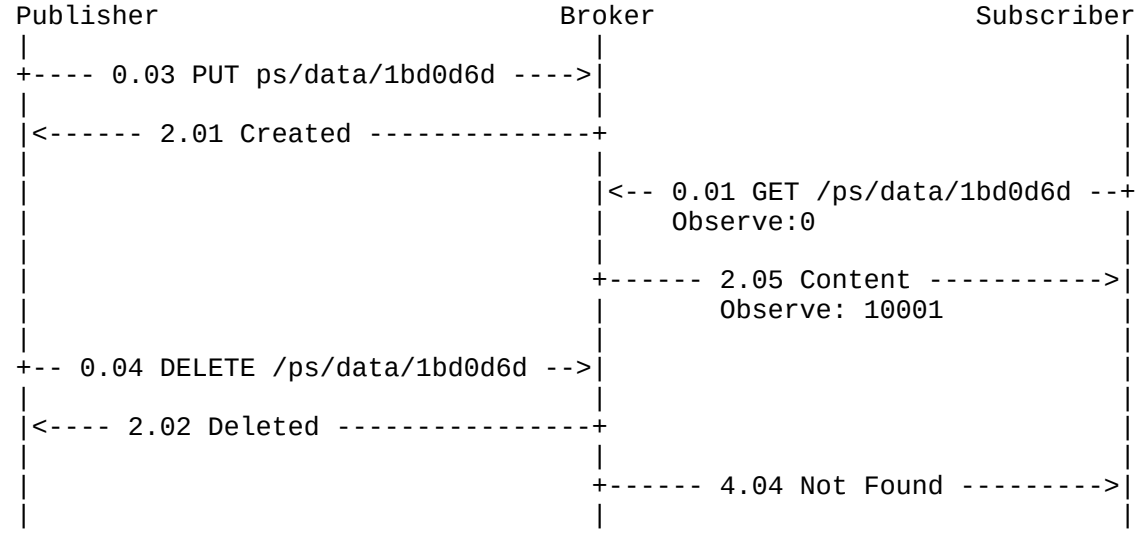

Figure 8: Example of Secure Pub-Sub Communication using CoAP

6.1. Using COSE to Protect the Published Topic Data

The Publisher uses the symmetric COSE Key received from the KDC to protect the payload of the Publish operation (see Section 3.2.1 of [I-D.ietf-core-coap-pubsub]). Specifically, the Publisher creates a COSE_Encrypt0 object [RFC9052][RFC9053] by means of the COSE Key currently used as group key. The encryption algorithm and Base IV to use are specified by the 'alg' and 'Base IV' parameters of the COSE Key, together with its key identifier in the 'kid' parameter.

Also, the Publisher uses its private key corresponding to the public key sent to the KDC, in order to countersign the COSE_Encrypt0 object as specified in [RFC9338]. The countersignature is specified in the 'Countersignature version 2' parameter, within the 'unprotected' field of the COSE_Encrypt0 object.

Finally, the Publisher sends the COSE_Encrypt0 object conveying the countersignature to the Broker, as payload of the PUT request sent to the topic-data of the topic targeted by the Publish operation.

Upon receiving a response from the topic-data resource at the Broker, the Subscriber uses the 'kid' parameter from the 'Countersignature version 2' parameter within the 'unprotected' field of the COSE_Encrypt0 object, in order to retrieve the Publisher's public key from the Broker or from its local storage. Then, the Subscriber uses that public key to verify the countersignature.

In case of successful verification, the Subscriber uses the 'kid' parameter from the 'unprotected' field of the COSE_Encrypt0 object, in order to retrieve the COSE Key used as current group key from its local storage. Then, the Subscriber uses that group key to verify and decrypt the COSE_Encrypt0 object. In case of successful verification, the Subscriber delivers the received topic data to the application.

The COSE_Encrypt0 object is constructed as follows.

The 'protected' field MUST include:

* The 'alg' parameter, with value the identifier of the AEAD algorithm specified in the 'alg' parameter of the COSE Key used as current group key.

The 'unprotected' field MUST include:

* The 'kid' parameter, with the same value specified in the 'kid' parameter of the COSE Key used as current group key. This value represents the current Group ID (Gid) of the security group associated with the application group (topic).

* The 'Partial IV' parameter, with value set to the current Sender Sequence Number of the Publisher. All leading bytes of value zero SHALL be removed when encoding the Partial IV, except in the case of Partial IV value 0, which is encoded to the byte string 0x00.

  The Publisher MUST initialize the Sender Sequence Number to 0 upon joining the security group, and MUST reset it to 0 upon receiving a new group key as result of a group rekeying (see Section 5). The Publisher MUST increment its Sender Sequence Number value by 1, after having completed an encryption operation by means of the current group key.

When the Publisher exhausts its Sender Sequence Numbers, the Publisher MUST NOT protect further topic data by using the current group key while still retaining its current Sender ID, and MUST send a Key Renewal Request message to the KDC (see Section 4.2.2). This will result in the KDC rekeying the group and distributing a new group key, or in the KDC providing the Publisher with a new Sender ID. The Publisher MUST reset its Sender Sequence Number to 0 upon receiving a new Sender ID from the KDC.

* The 'Countersignature version 2' parameter, specifying the countersignature of the COSE_Encrypt0 object. In particular:

  - The 'protected' field includes the 'alg' parameter, with value the identifier of the Signature Algorithm used in the security group.

  - The 'unprotected' field includes the 'kid' parameter, with value the Publisher's Sender ID that the Publisher obtained from the KDC when joining the security group, as value of the 'group_SenderId' parameter of the Join Response (see Section 4.1.2).

  - The 'signature' field, with value the countersignature.

  The countersignature is computed as defined in [RFC9338], by using the private key of the Publisher as signing key, and by means of the Signature Algorithm used in the group. The fields of the Countersign_structure are populated as follows:

  - 'context' takes "CounterSignature".

  - 'body_protected' takes the serialized parameters from the 'protected' field of the COSE_Encrypt0 object, i.e., the 'alg' parameter.

  - 'sign_protected' takes the serialized parameters from the 'protected' field of the 'Countersignature version 2' parameter, i.e., the 'alg' parameter.

  - 'external_aad is not supplied.

  - 'payload' is the ciphertext of the COSE_Encrypt0 object (see below).

The 'ciphertext' field specifies the ciphertext computed over the topic data to publish. The ciphertext is computed as defined in [RFC9052][RFC9053], by using the current group key as encryption key, the AEAD Nonce computed as defined in Section 6.2, the topic data to publish as plaintext, and the Enc_structure populated as follows:

* 'context' takes "Encrypt0".

* 'protected' takes the serialization of the protected parameter 'alg' from the 'protected' field of the COSE_Encrypt0 object.

* 'external_aad is not supplied.

6.2. AEAD Nonce

This section defines how to generate the AEAD nonce used for encrypting and decrypting the COSE_Encrypt0 object protecting the published topic data. This construction is analogous to that used to generate the AEAD nonce in the OSCORE security protocol (see Section 5.2 of [RFC8613]).

The AEAD nonce for producing or consuming the COSE_Encrypt0 object is constructed as defined below and also shown in Figure 9.

1. Left-pad the Partial IV (PIV) with zeroes to exactly 5 bytes.

2. Left-pad the Sender ID of the Publisher that generated the Partial IV (ID_PIV) with zeroes to exactly the nonce length of the AEAD algorithm minus 6 bytes.

3. Concatenate the size of the ID_PIV (a single byte S) with the padded ID_PIV and the padded PIV.

4. XOR the result from the previous step with the Base IV.

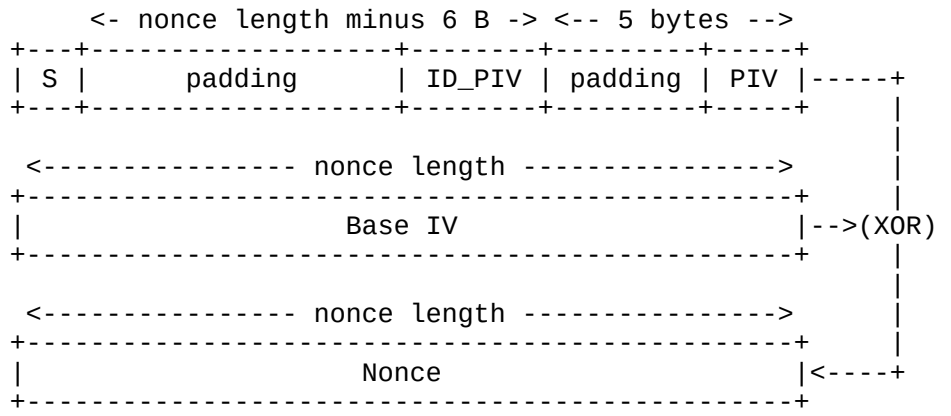

Figure 9: AEAD Nonce Construction

The construction above only supports AEAD algorithms that use nonces with length equal or greater than 7 bytes. At the same time, it makes it easy to verify that the nonces will be unique when used with the same group key, even though this is shared and used by all the Publishers in the security group. In fact:

* Since Publisher's Sender IDs are unique within the security group and they are not reassigned until a group rekeying occurs (see Section 4.1.2 and Section 5), two Publisher Clients cannot share the same tuple (S, padded ID_PIV) by construction.

* Since a Publisher increments by 1 its Sender Sequence Number after each use that it makes of the current group key, the Publisher never reuses the same tuple (S, padded ID_PIV, padded PIV) together with the same group key.

* Therefore neither the same Publisher reuses the same AEAD nonce with the same group key, nor any two Publishers use the same AEAD nonce with the same group key.

6.3. Replay Checks

In order to protect from replay of published topic data, every Subscriber maintains a Replay Window for each different Publisher in the same group. It is RECOMMENDED that the Replay Window has a default size of 32.

Upon receiving a topic data published by a given Publisher P, the Subscriber retrieves the Sender ID of P conveyed as 'kid' in the 'Countersignature version 2' parameter of the COSE_Encrypt0 object (see Section 6.1), and determines the Replay Window W_P associated with P.

The Subscriber MUST verify that, according to W_P, the Sender Sequence Number SN_P specified by the 'Partial IV' parameter of the COSE_Encrypt0 object has not been received before from P.

If the verification above fails, the Subscriber MUST stop processing the COSE_Encrypt0 object conveying the topic data. If the value of SN_P is strictly smaller than the currently smallest value in W_P, then the Subscriber MUST stop processing the COSE_Encrypt0 object.

If the verification above succeeds, the Subscriber proceeds with processing the COSE_Encrypt0 object, by verifying the countersignature from P using P's public key as well as by decrypting the COSE_Encrypt0 object using the group key. If both operations succeed, the Subscriber updates W_P as follows:

* If SN_P is strictly greater than the currently largest value in W_P, then W_P is updated in order to set SN_P as its largest value.

* SN_P is marked to denote that it has been received.

The operation of validating the 'Partial IV' and updating the Replay Window MUST be atomic.

Upon installing a new group key (e.g., due to a group rekeying performed by the KDC, see Section 5) or upon receiving published topic data from a given Publisher for the first time, the Subscriber initializes the Replay Window corresponding to that Publisher, i.e., the smallest value of the Replay Window is set to 0.

7. Security Considerations

Security considerations for this profile are inherited from [RFC9594], the ACE framework for Authentication and Authorization [RFC9200], and the specific transport profile of ACE used, such as [RFC9202] and [RFC9203].

The following security considerations also apply for this profile.

Consistent with the intended group-confidentiality model, each Client in a security group is able to decrypt the data published in the topic(s) associated with that group, by using the symmetric group key that is shared with all the other group members.

At the same time, source authentication of the published topic data is achieved by means of a digital signature, which the Publisher of the data in question computes with its private key and embeds in the published data. This ensures integrity of the published topic data as well as its origin, thus preventing a group member from impersonating another one.

To this end, both Publishers and Subscribers rely on asymmetric cryptography, while Subscribers must be able to access the public keys of all the Publishers to a specific topic in order to verify the signature over the published topic data. This might make the message exchange quite heavy for small constrained devices.

The Broker is only trusted with verifying that a Publisher is authorised to publish on a certain topic, and with distributing that data only to the Subscribers authorised to obtain it. However, the Broker is not trusted with the published data in itself, which the Broker cannot read or modify as it does not have access to the group key required for decrypting the data.

With respect to the reusage of nonces for Proof-of-Possession input, the same considerations apply as in [I-D.ietf-ace-key-groupcomm-oscore].

Access tokens might have to be revoked before their expiration time. [I-D.ietf-ace-revoked-token-notification] provides a list of possible circumstances where this can happen, and specifies a method that an Authorization Server can use in order to notify the KDC, the Broker, and the Clients about pertaining access tokens that have been revoked but are not expired yet.

Clients can be excluded from future communications related to a topic, by appropriately re-keying the group associated with the topic in question.

8. IANA Considerations

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

This document has the following actions for IANA.

8.1. ACE Groupcomm Key Types

IANA is asked to register the following entry in the "ACE Groupcomm Key Types" registry defined in Section 11.8 of [RFC9594].

* Name: Group_PubSub_Keying_Material
* Key Type Value: GROUPCOMM_KEY_TBD
* Profile: coap_group_pubsub_app (Section 8.2 of [RFC-XXXX]).
* Description: Encoded as described in Section 4.1.2 of [RFC-XXXX].
* References: [RFC9052], [RFC9053], [RFC-XXXX]

8.2. ACE Groupcomm Profiles

IANA is asked to register the following entries in the "ACE Groupcomm Profiles" registry defined in Section 11.9 of [RFC9594].

* Name: coap_group_pubsub_app
* Description: Application profile to provision keying material for participating in group communication based on the Pub-Sub architecture [I-D.ietf-core-coap-pubsub] for CoAP [RFC7252] and protected with COSE [RFC9052][RFC9053][RFC9338].
* CBOR Value: TBD
* Reference: [RFC-XXXX]

8.3. CoRE Resource Type

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

* Value: "core.ps.gm"
* Description: Group-membership resource for Pub-Sub communication.
* Reference: [RFC-XXXX]

Clients can use this resource type to discover a group membership resource at the KDC.

8.4. AIF Media-Type Sub-Parameters

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

* Parameter: Toid

* Name: pubsub-topic

* Description/Specification: Name of one application group (topic) or of a corresponding security group.

* Reference: [RFC-XXXX]

* Parameter: Tperm

* Name: pubsub-perm

* Description/Specification: Permissions pertaining to application groups (topics) or to corresponding security groups.

* Reference: [RFC-XXXX]

8.5. CoAP Content-Formats

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

* Content Type: application/aif+cbor;Toid="pubsub-topic",Tperm="pubsub-perm"

* Content Coding: -

* ID: 294 (suggested)

* Reference: [RFC-XXXX]

* Content Type: application/aif+json;Toid="pubsub-topic",Tperm="pubsub-perm"

* Content Coding: -

* ID: 295 (suggested)

* Reference: [RFC-XXXX]

8.6. TLS Exporter Labels

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

* Value: EXPORTER-ACE-Sign-Challenge-pubsub-app

* DTLS-OK: Y

* Recommended: N

* Reference: Section 4.1.1.2 of [RFC-XXXX]

9. References

9.1. Normative References

[I-D.ietf-core-coap-pubsub]
Jimenez, J., Koster, M., and A. Ker辰nen, "A publish-subscribe architecture for the Constrained Application Protocol (CoAP)", Work in Progress, Internet-Draft, draft-ietf-core-coap-pubsub-18, 28 February 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-core-coap-pubsub-18>.

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

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

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

[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/rfc/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/rfc/rfc5705>.

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

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

[RFC6749] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework", RFC 6749, DOI 10.17487/RFC6749, October 2012, <https://www.rfc-editor.org/rfc/rfc6749>.

[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/rfc/rfc7252>.

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

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

[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/rfc/rfc8174>.

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

[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/rfc/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/rfc/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/rfc/rfc8610>.

[RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <https://www.rfc-editor.org/rfc/rfc8949>.

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

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

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

[RFC9200] Seitz, L., Selander, G., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "Authentication and Authorization for Constrained Environments Using the OAuth 2.0 Framework (ACE-OAuth)", RFC 9200, DOI 10.17487/RFC9200, August 2022, <https://www.rfc-editor.org/rfc/rfc9200>.

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

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

[RFC9290] Fossati, T. and C. Bormann, "Concise Problem Details for Constrained Application Protocol (CoAP) APIs", RFC 9290, DOI 10.17487/RFC9290, October 2022, <https://www.rfc-editor.org/rfc/rfc9290>.

[RFC9338] Schaad, J., "CBOR Object Signing and Encryption (COSE): Countersignatures", STD 96, RFC 9338, DOI 10.17487/RFC9338, December 2022, <https://www.rfc-editor.org/rfc/rfc9338>.

[RFC9594] Palombini, F. and M. Tiloca, "Key Provisioning for Group Communication Using Authentication and Authorization for Constrained Environments (ACE)", RFC 9594, DOI 10.17487/RFC9594, September 2024, <https://www.rfc-editor.org/rfc/rfc9594>.

9.2. Informative References

[I-D.ietf-ace-edhoc-oscore-profile] Selander, G., Mattsson, J. P., Tiloca, M., and R. H旦glund, "Ephemeral Diffie-Hellman Over COSE (EDHOC) and Object Security for Constrained Environments (OSCORE) Profile for Authentication and Authorization for Constrained Environments (ACE)", Work in Progress, Internet-Draft, draft-ietf-ace-edhoc-oscore-profile-07, 3 March 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-ace-edhoc-oscore-profile-07>.

[I-D.ietf-ace-key-groupcomm-oscore] Tiloca, M. and F. Palombini, "Key Management for Group Object Security for Constrained RESTful Environments (Group OSCORE) Using Authentication and Authorization for Constrained Environments (ACE)", Work in Progress, Internet-Draft, draft-ietf-ace-key-groupcomm-oscore-17, 11 April 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-ace-key-groupcomm-oscore-17>.

[I-D.ietf-ace-revoked-token-notification] Tiloca, M., Palombini, F., Echeverria, S., and G. Lewis, "Notification of Revoked Access Tokens in the Authentication and Authorization for Constrained Environments (ACE) Framework", Work in Progress, Internet-Draft, draft-ietf-ace-revoked-token-notification-09, 22 September 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-ace-revoked-token-notification-09>.

[I-D.ietf-cose-cbor-encoded-cert] Mattsson, J. P., Selander, G., Raza, S., H旦glund, J., and M. Furuhed, "CBOR Encoded X.509 Certificates (C509 Certificates)", Work in Progress, Internet-Draft, draft-ietf-cose-cbor-encoded-cert-13, 3 March 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-cose-cbor-encoded-cert-13>.

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/rfc/rfc8259>.

[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/rfc/rfc8613>.

[RFC9202] Gerdes, S., Bergmann, O., Bormann, C., Selander, G., and L. Seitz, "Datagram Transport Layer Security (DTLS) Profile for Authentication and Authorization for Constrained Environments (ACE)", RFC 9202, DOI 10.17487/RFC9202, August 2022, <https://www.rfc-editor.org/rfc/rfc9202>.

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

Appendix A. Requirements on Application Profiles

This section lists how this application profile of ACE addresses the requirements defined in Appendix A of [RFC9594].

A.1. Mandatory-to-Address Requirements

* REQ1: Specify the format and encoding of scope. This includes defining the set of possible roles and their identifiers, as well as the corresponding encoding to use in the scope entries according to the used scope format: see Section 3.4.1.

* REQ2: If scope uses AIF, register its specific instance of "Toid" and "Tperm" as media type parameters and a corresponding Content-Format, as per the guidelines in [RFC9237]: see Section 8.4 and Section 8.5.

* REQ3: If used, specify the acceptable values for the ’sign_alg’ parameter: values from the "Value" column of the "COSE Algorithms" registry [IANA.cose_algorithms].

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

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

* REQ6: Specify the acceptable formats for authentication credentials and, if applicable, the acceptable values for the 'cred_fmt' parameter: acceptable formats explicitly provide the public key as well as the comprehensive set of information related to the public key algorithm (see Section 3.6 and Section 4.1.2). Consistent acceptable values for 'cred_fmt' are taken from the "Label" column of the "COSE Header Parameters" registry [IANA.cose_header-parameters].

* REQ7: If the value of the GROUPNAME URI path and the group name in the access token scope ('gname') are not required to coincide, specify the mechanism to map the GROUPNAME value in the URI to the group name: not applicable; a perfect matching is required.

* REQ8: Define whether the KDC has an authentication credential as required for the correct group operation and if this has to be provided through the 'kdc_cred' parameter: optional, see Section 4.1.2.

* REQ9: Specify if any part of the KDC interface as defined in [RFC9594] is not supported by the KDC: some are left optional, see Section 4.

* REQ10: Register a Resource Type for the group-membership resources, which is used to discover the correct URL for sending a Join Request to the KDC: the Resource Type (rt=) Link Target Attribute value "core.ps.gm" is registered in Section 8.3.

* REQ11: Define what specific actions (e.g., CoAP methods) are allowed on each resource accessible through the KDC interface, depending on: whether the Client is a current group member; the roles that a Client is authorised to take as per the obtained access token; and the roles that the Client has as a current group member: see Section 4 of this document.

* REQ12: Categorize possible newly defined operations for Clients into primary operations expected to be minimally supported and secondary operations, and provide accompanying considerations: none added.

* REQ13: Specify the encoding of group identifiers: CBOR byte string, with value used also to identify the current group key used in the security group (see Section 4.1.2).

* REQ14: Specify the approaches used to compute and verify the PoP evidence to include in the 'client_cred_verify' parameter and which of those approaches is used in which case: see Section 4.1.1.2.

* REQ15: Specify how the nonce N_S is generated, if the access token is not provided to the KDC through the Token Transfer Request sent to the /authz-info endpoint (e.g., the access token is instead transferred during the establishment of a secure communication association): see Section 4.1.1.2.

* REQ16: Define the initial value of the version number for the group keying material: the initial value MUST be set to 0 (see Section 4.1.2).

* REQ17: Specify the format of the group keying material that is conveyed in the 'key' parameter: see Section 4.1.2.

* REQ18: Specify the acceptable values of the 'gkty' parameter. For each of them, register a corresponding entry in the "ACE Groupcomm Key Types" IANA registry if such an entry does not exist already: Group_PubSub_Keying_Material, see Section 4.1.2 and Section 8.1.

* REQ19: Specify and register the application profile identifier: coap_group_pubsub_app (see Section 4.1.2 and Section 8.2) (see Section 8.2).

* REQ20: If used, specify the format and default values of the entries of the CBOR map to include in the 'group_policies' parameter: not applicable.

* REQ21: Specify the approaches used to compute and verify the PoP evidence to include in the 'kdc_cred_verify' parameter and which of those approaches is used in which case. If external signature verifiers are supported, specify how those provide a nonce to the KDC to be used for computing the PoP evidence: see Section 4.1.2.

* REQ22: Specify the communication protocol that members of the group use to communicate with each other (e.g., CoAP for group communication): CoAP [RFC7252], used for Pub-Sub communications as defined in [I-D.ietf-core-coap-pubsub].

* REQ23: Specify the security protocol that members of the group use to protect the group communication (e.g., Group OSCORE). This must provide encryption, integrity, and replay protection: Publishers in a group use a symmetric group key to protect published topic data as a COSE_Encrypt0 object, per the AEAD algorithm specified by the KDC. A Publisher also produces a COSE countersignature of the COSE_Encrypt0 object by using its private key, per the signature algorithm specified by the KDC.

* REQ24: Specify how the communication is secured between the Client and the KDC. Optionally, specify a transport profile of ACE [RFC9200] to use between the Client and the KDC: ACE transport profile such as for DTLS [RFC9202] or OSCORE [RFC9203].

* REQ25: Specify the format of the node identifiers of group members: the Sender ID defined in Section 4.1.2.

* REQ26: Specify policies at the KDC to handle node identifiers that are included in the 'get_creds' parameter but are not associated with any current group member: see Section 4.2.1.

* REQ27: Specify the format of (newly generated) individual keying material for group members or of the information to derive such keying material, as well as the corresponding CBOR map key that has to be registered in the "ACE Groupcomm Parameters" registry: see Section 4.2.2.

* REQ28: Specify which CBOR tag is used for identifying the semantics of binary scopes, or register a new CBOR tag if a suitable one does not exist already: see Section 3.5 and Section 8.5.

* REQ29: Categorize newly defined parameters according to the same criteria of Section 8 of [RFC9594]: a Publisher Client MUST support 'group_SenderId' in 'key'; see Section 4.1.2.

* REQ30: Define whether Clients must, should, or may support the conditional parameters defined in Section 8 of [RFC9594] and under which circumstances: a Publisher Client MUST support the client_cred', 'cnonce', and 'client_cred_verify' parameters (see Section 4.1.1). A Publisher Client that provides the access token to the KDC through the /authz-info endpoint MUST support the parameter 'kdcchallenge' (see Section 3.6).

A.2. Optional-to-Address Requirements

* OPT1: Optionally, if the textual format of scope is used, specify CBOR values to use for abbreviating the role identifiers in the group: not applicable.

* OPT2: Optionally, specify the additional parameters used in the exchange of Token Transfer Request and Response: none are defined.

* OPT3: Optionally, specify the negotiation of parameter values for signature algorithm and signature keys, if the 'sign_info' parameter is not used: see Section 3.6.

* OPT4: Optionally, specify possible or required payload formats for specific error cases: see Section 4.1.3.

* OPT5: Optionally, specify additional identifiers of error types as values of the 'error-id' field within the Custom Problem Detail entry 'ace-groupcomm-error': no.

* OPT6: Optionally, specify the encoding of the 'creds_repo' parameter if the default one is not used: no.

* OPT7: Optionally, specify the functionalities implemented at the resource hosted by the Client at the URI indicated in the 'control_uri' parameter, including the encoding of exchanged messages and other details: no.

* OPT8: Optionally, specify the behavior of the POST handler of group-membership resources, for the case when it fails to retrieve an authentication credential for the specific Client: The KDC MUST reply with a 4.00 (Bad Request) error response to the Join Request (see Section 4.1.3).

* OPT9: Optionally, define a default group rekeying scheme to refer to in case the 'rekeying_scheme' parameter is not included in the Join Response: the "Point-to-Point" rekeying scheme registered in Section 11.13 of [RFC9594].

* OPT10: Optionally, specify the functionalities implemented at the resource hosted by the Client at the URI indicated in the 'control_group_uri' parameter, including the encoding of exchanged messages and other details: no.

* OPT11: Optionally, specify policies that instruct Clients to retain messages and for how long, if those are unsuccessfully decrypted: no.

* OPT12: Optionally, specify for the KDC to perform a group rekeying when receiving a Key Renewal Request, together with or instead of renewing individual keying material: the KDC SHOULD NOT perform a group rekeying, unless already scheduled to occur shortly (see Section 4.2.2).

* OPT13: Optionally, specify how the identifier of a group member's authentication credential is included in requests sent to other group members: no.

* OPT14: Optionally, specify additional information to include in rekeying messages for the "Point-to-Point" group rekeying scheme (see Section 6 of [RFC9594]): no.

Appendix B. Document Updates

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

B.1. Version -10 to -11

* Recommended /ps/TOPICNAME as path ot topic resources at the Broker.

* The request for a new Sender ID uses the method POST.

* Fixed description of ACE Group Error with identifier 4.

* Aligned requirement formulation with that in RFC 9594.

* Updated references.

* Clarifications and editorial improvements.

B.2. Version -09 to -10

* More details on the scope format.

* More details in the encoding of the 'key' parameter in the Join Response.

* More details on exchanges between group members and KDC.

* More details on the rekeying process and rekeying messages.

* Defined replay checks at the Subscriber.

* Improved examples.

* Improved security considerations.

* Revised IANA considerations.

* Aligned the list of profile requirements with draft-ietf-ace-key-groupcomm.

* Clarifications and editorial improvements.

B.3. Version -08 to -09

* Improved terminology section.

* Generalized scope format for future, admin-related extensions.

* Improved definition of permissions in the format of scope.

* Clarified alternative computing of N_S Challenge when DTLS is used.

* Use of the parameter ’exi’ in the Join Response.

* Use of RFC 9290 instead of the custom format of error responses.

* Fixed construction of the COSE_Encrypt0 object.

* Fixed use of the resource type "core.ps.gm".

* Updated formulation of profile requirements.

* Clarification and editorial improvements.

B.4. Version -07 to -08

* Revised presentation of the scope format.

* Revised presentation of the Join Request-Response exchange.

* The ’cnonce’ parameter must be present in the Join Request.

* The ’kid’ of the group key is used as Group Identifier.

* Relaxed inclusion of the ’peer_roles’ parameter.

* More detailed description of the encryption and signing operations.

* Defined construction of the AEAD nonce.

* Clarifications and editorial improvements.

B.5. Version -06 to -07

* Revised abstract and introduction.
* Clarified use of "application groups".
* Revised use of protocols and transport profiles with Broker and KDC.
* Revised overview of the profile and its security associations.
* Revised presentation of authorization flow.
* Subscribers cannot be anonymous anymore.
* Revised scope definition.
* Revised Join Response.
* Revised COSE countersignature, COSE encrypt objects.
* Further clarifications, fixes and editorial improvements.

Acknowledgments

The authors wish to thank Ari Ker辰nen, John Preu Mattsson, Jim Schaad, Ludwig Seitz, and Gran Selander for the useful discussion and reviews that helped shape this document.

This work was supported by the Sweden’s Innovation Agency VINNOVA within the EUREKA CELTIC-NEXT projects CRITISEC and CYPRESS; and by the H2020 project SIFIS-Home (Grant agreement 952652).

Authors’ Addresses

Francesca Palombini
Ericsson
Email: francesca.palombini@ericsson.com

Cigdem Sengul
Brunel University
Email: csengul@acm.org

Marco Tiloca
RISE AB
Email: marco.tiloca@ri.se