Workgroup: ACE Working Group
Internet-Draft:
draft-ietf-ace-pubsub-profile-06
Published: 13 March 2023
Intended Status: Standards Track
Expires: 14 September 2023
Authors: F. Palombini
Ericsson
C. Sengul
Brunel University
M. Tiloca
RISE AB

Publish-Subscribe Profile for Authentication and Authorization for Constrained Environments (ACE)

Abstract

This document defines an application profile for enabling secure group communication for a constrained Publish-Subscribe (pub/sub) scenario, where Publishers and Subscribers communicate through a broker, using the ACE framework. This profile relies on transport layer or application layer security 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. The document describes how to request and provision keying material for group communication, and protect the content of the pub/sub client message exchange, focusing mainly on the pub/sub scenarios using the Constrained Application Protocol (CoAP) [I-D.ietf-core-coap-pubsub].

Status of This Memo

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

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

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

This Internet-Draft will expire on 14 September 2023.

Table of Contents

1. Introduction

In the publish-subscribe (pub/sub) scenario, devices with limited reachability communicate via a broker, which enables store-and-forward messaging between these devices. This document specifies how to request, distribute and renew the keying material and configuration parameters to protect message exchanges for pub/sub communication, using [I-D.ietf-ace-key-groupcomm], which expands from the ACE framework ([RFC9200]). Message exchanges among the participants as well as message formats and processing follow the specifications for provisioning and renewing keying material in group communication scenarios in [I-D.ietf-ace-key-groupcomm].

The pub/sub communication using the Constrained Application Protocol (CoAP) [RFC7252] is specified in [I-D.ietf-core-coap-pubsub].This document gives detailed specifications for CoAP pub/sub, and describes how it can be applicable to MQTT [MQTT-OASIS-Standard-v5]; similar adaptations can extend to other transport protocols as well.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [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 [RFC9200], and Authorization Information Format (AIF) [RFC9237] to express authorization information. In particular, analogously to [RFC9200], terminology for entities in the architecture such as Client (C), Resource Server (RS), and Authorization Server (AS) is defined in OAuth 2.0 [RFC6749].

* The terms and concept related to the message formats and processing, specified in [I-D.ietf-ace-key-groupcomm], for provisioning and renewing keying material in group communication scenarios.

* The terms and concepts of pub/sub group communication, as described in [I-D.ietf-core-coap-pubsub].

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

A principal interested to participate 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 may be a set of nodes registered to a pub/sub topic.

2. Application Profile Overview

This document describes how to use [I-D.ietf-ace-key-groupcomm] and [RFC9200] to perform authentication, authorization and key distribution actions as overviewed in Section 2 of [I-D.ietf-ace-key-groupcomm], when the considered group is pub/sub clients belonging to the same security group.

Pub/sub clients communicate within their application groups mapped to a collection of pub/sub topics. The pub/sub topics may consist of one or more sub-topics, which may have their own sub-topics, forming a hierarchy. The applications decide how to map this hierarchy into different application groups, and 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.

The architecture of the scenario is shown in Figure 1. A Client can act both as a publisher and a subscriber, publishing to some topics, and subscribing to others. However, for the simplicity of presentation, this profile describes Publisher and Subscriber Clients separately. The Broker acts as the ACE RS, and also corresponds to the Dispatcher in [I-D.ietf-ace-key-groupcomm]. The Clients communicate with The Key Distribution Center (KDC) to join security groups, and obtain the group keying material.

Both Publisher and Subscriber Clients use the same pub/sub communication protocol and the same transport profile of ACE in their interaction with the broker. The pub/sub communication protocol considered in this document is CoAP, as described in [I-D.ietf-core-coap-pubsub], but the specification can apply to other pub/sub protocols such as MQTT [MQTT-OASIS-Standard-v5], or other transport protocols. All clients MUST use CoAP when communicating to the KDC.

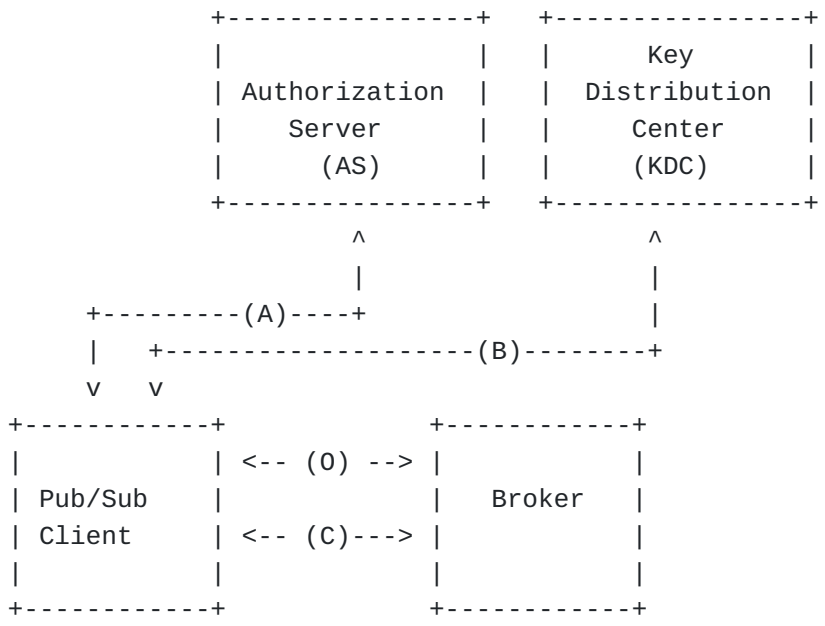

Figure 1: Architecture for Pub/Sub with Authorization Server and Key Distribution Center

All communications between the involved entities MUST be secured. This profile expects the establishment of a secure connection between a Client and Broker, using an ACE transport profile such as DTLS [RFC9202] or OSCORE [RFC9203] (A and C). Once the client establishes a secure association with KDC with the help of AS, it can request to join the security groups of its pub/sub topics (A and B), and can communicate securely with the other group members, using the keying material provided by the KDC.

(C) corresponds to the exchange between the Client and the Broker, where the Client sends its access token to the Broker and establishes a secure connection with the Broker. Depending on the Information received in (A), the connection set-up may involve, for example, a DTLS handshake, or other protocols. Depending on the application, the set up phase may be skipped: for example, if OSCORE is used directly.

In addition, this document describes an Optional Discovery though Broker (O), where an anonymous Clients MAY discover the topic categories, topics resources, the AS and the KDC from the Broker.

It must be noted that Clients maintain two different security associations. On the one hand, the Publisher and the Subscriber clients have a security association with the Broker, which, as the ACE RS, verifies that the Clients are authorized (Security Association 1). On the other hand, the Publisher has a security association with the Subscriber, to protect the publication content (Security Association 2) while sending it through the broker. The

Security Association 1 is set up using AS and a transport profile of [RFC9200], the Security Association 2 is set up using AS, KDC and [I-D.ietf-ace-key-groupcomm].

Given that the publication content is protected, the Broker MAY accept unauthorised Subscribers. In this case, the Subscriber client MAY skip setting up Security Association 1 with the Broker and connect to it as an anonymous client to subscribe to topics of interest at the Broker.

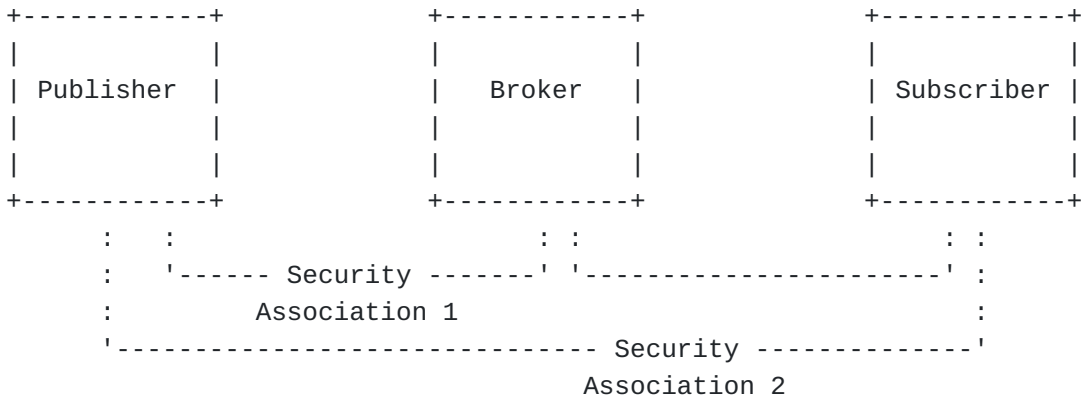

Figure 2: Security Associations between Publisher, Broker, Subscriber pairs.

In summary, this profile describes how:

1. A Client gets the authorization to join a security group, and providing it with the group keying material to communicate with other group members.

2. A Client retrieves group keying material to publish protected publications to the Broker or read protected publications.

3. A Client retrieves authentication credentials of other group members, and provides and updates own authentication credentials.

4. A Client is removed from the group.

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

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

3. Getting Authorisation to Join a Pub/sub security group (A)

Figure Figure 3 provides a high level overview of the message flow for a node getting authorisation to join a group. This message flow is expanded in the subsequent sections.

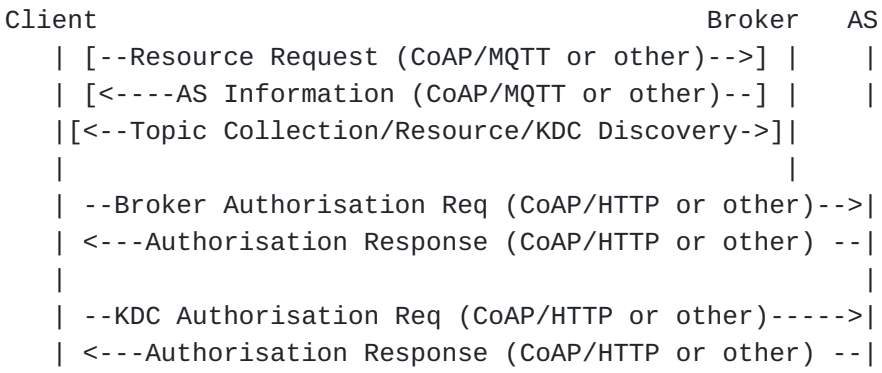

Figure 3: Authorisation Flow

Since [RFC9200] recommends the use of CoAP and CBOR, this document describes the exchanges assuming CoAP and CBOR are used. However, using HTTP instead of CoAP is possible, using the corresponding parameters and methods. Analogously, JSON [RFC8259] can be used instead of CBOR, using the conversion method specified in Sections 6.1 and 6.2 of [RFC8949]. In case JSON is used, the Content Format or Media Type of the message has to be changed accordingly. Exact definition of these exchanges are considered out of scope for this document.

3.1. AS Discovery at the Broker (Optional)

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

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

Figure 4: AS Information example

3.2. Topic and KDC Discovery at the Broker

A Broker can offer a topic discovery entry point to enable clients to find topics of interest. The resource entry point thus represents a collection of related resources as specified in [RFC6690] and is

identified by the resource type "core.ps.coll". A topic collection is a group of topic configuration resources that define topic properties and are identified by the resource type "core.ps.conf". An anonymous pub/sub client MAY request a collection of the topics present in the broker by making a CoAP GET request to the collection URI. An anonymous pub/sub client MAY read the configuration of a topic by making a CoAP GET request to the topic configuration URI. (ToDo: Consider a discovery token to be consumed by the Broker for topic collection, and topic configuration?)

(ToDo: Instead of defining "core.ps.gm", need to extend Topic Configuration Representation in core-coap-pubsub to include KDC?) 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 and its sub-resources at Broker, e.g., by using a link-format document [RFC6690]. Applications can use this common resource type to discover links to group-membership resources for joining pub/sub groups.

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

The Client sends two Authorisation Requests to the AS for two audiences: the Broker and the KDC, respectively. AS handles authorisation requests for topics a Client is allowed to Publish or Subscribe to the Broker, corresponding to an application group. The client sends a request to the KDC to join the security group(s) corresponding to those application groups to be able protect the message content with the group key.

Communications between the Client and the AS MUST be secured, according to what is defined by the used transport profile of ACE. This section builds on Section 3 of [I-D.ietf-ace-key-groupcomm] and defined only additions or modifications to that specification.

Both Authorisation Requests include the following fields (Section 3.1 of [I-D.ietf-ace-key-groupcomm]):

* 'scope': Optional. If present, specifies the name of the topic groups, that the Client requests to access. This parameter is a CBOR byte string that encodes a CBOR array, whose format SHOULD follow the data model AIF-PUBSUB-GROUPCOMM defined below.

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

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

For the Broker, the scope represents pub/sub topics i.e., the application group, and for the KDC, the scope represents the

corresponding security group. This document expects a one-to-one mapping between the application group and the security group, and the client uses the same scope for both requests. If there is not a one-to-one mapping, the client MUST ask for the correct scopes in its Authorization Requests, and the correct policies regarding both sets of scopes MUST be available to the AS. How the client discovers the (application group, security group) association is out of scope of this document.

3.3.1. Format of Scope

The 'scope' parameter SHOULD follow the AIF format (REQ1). However, if the ACE transport profile, supports another 'scope' format, then implementations MAY use this format.

Based on the generic AIF model

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

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

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

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

- The permission set ("Tperm") is a CBOR unsigned integer with value, specifying the Client role, based on the operations the Client can execute on Topic Data in the group. The set of numbers representing the permissions is converted into a single number by taking two to the power of each method number and computing the inclusive OR of the binary representations of all the power values. The roles a Client is allowed are Publish (1), Subscribe (or Read) (2) and Delete (3). An Admin(0) role is also defined, which is reserved for expressing permissions for Administrators of Pub/Sub groups. For Pub/Sub client communication, the scope entry MUST NOT include the Admin permission i.e., the least significant bit of "Tperm" always set to 0.

```
AIF-PUBSUB-GROUPCOMM = AIF-Generic<pubsub-topic, pubsub-perm>
 pubsub-topic = tstr ; Pub/sub topic name
                     ; (the associated security group)

 pubsub-perm = uint . bits pubsub-roles

 pubsub-roles = &(
  Admin: 0,
  Pub: 1,
  Sub: 2,
  Delete: 3
 )

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

Figure 5: Pub/Sub scope using the AIF format

3.4. Authorisation response

The AS responds with an Authorization Response to each request, containing claims, as defined in Section 5.8.2 of [RFC9200] and Section 3.2 of [I-D.ietf-ace-key-groupcomm] with the following additions:

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

- The AS MUST include the 'scope' parameter, when the value included in the Access Token differs from the one specified by the joining node in the Authorization Request. In such a case, the second element of each scope entry MUST be present, and specifies the set of roles that the joining node is actually authorized to take in for that scope entry, encoded as specified in Section 3.3.

ToDo: Extend the authorisation response to describe the token returned, and do a MUST on the Audience claim to indicate the response is for KDC or Broker?

Furthermore, the AS MAY use the extended format of scope defined in Section 7 of [I-D.ietf-ace-key-groupcomm] for the 'scope' claim of the Access Token. In such a case, the AS MUST use the CBOR tag with tag number TAG_NUMBER, associated with the CoAP Content-Format CF_ID for the media type application/aif+cbor registered in Section 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 3.3.1 of this document.

3.5. Token Transfer to KDC

After receiving a token from the AS, the Client transfers the token to the KDC using one of the methods defined Section 3.3 [I-D.ietf-ace-key-groupcomm]. This typically includes sending a POST request to the authz-info endpoint. However, if using the DTLS transport profile of ACE [RFC9202] 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 DTLS ClientKeyExchange message. In addition to that, the following applies.

In the token transfer response to the Publisher Clients, i.e., the Clients whose scope of the access token includes the "Pub" role, the KDC MUST include the parameter 'kdcchallenge' in the CBOR map. 'kdcchallange' is a challenge N_S generated by the KDC, and is RECOMMENDED to be a 8-byte long random nonce. Later when joining the group, the Publisher Client can use the 'kdcchallenge' as part of proving possession of its private key (see [I-D.ietf-ace-key-groupcomm]). If a Publisher Client 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 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

The Clients uses the following KDC resources to enable group communication:

KDC resource	Description	Operations
/ace-group	Required. Contains a set of group names, each corresponding to one of the specified group identifiers	FETCH (All Clients)
/ace-group/ GROUPNAME	Required. Contains symmetric group keying material associated with GROUPNAME	GET, POST (All)
/ace-group/ GROUPNAME/creds	Required. Contains the authentication credentials of all the Publisher members of the group with name GROUPNAME	GET, FETCH (All)
/ace-group/ GROUPNAME/num	Required. Contains the current version number for the symmetric group keying material of the group with name GROUPNAME	GET (All)
/ace-group/ GROUPNAME/ nodes/NODENAME	Required. Contains the group keying material for that group member NODENAME in GROUPNAME.	GET, DELETE (All). PUT not supported.
/ace-group/ GROUPNAME/ nodes/NODENAME/ cred	Required. Authentication credential for NODENAME in the group GROUPNAME	POST (Pub)
/ace-group/ GROUPNAME/kdc-cred	MUST be hosted if a group re-keying mechanism is used. Contains the authentication credential of the KDC for the group with name GROUPNAME.	GET (All)

KDC resource	Description	Operations
/ace-group/ GROUPNAME/ policies	Optional. Contains the group policies of the group with	
name GROUPNAME.	GET (All)	

Table 1

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

4.1. Joining a Security Group

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

On a successful join, the Clients receive the symmetric COSE Key received from the KDC to protect the payload of a published topic data.

The message exchange between the joining node and the KDC follows what's defined in Section 4.3.1.1 of [I-D.ietf-ace-key-groupcomm] and only additions or modifications to that specification are defined in this document.

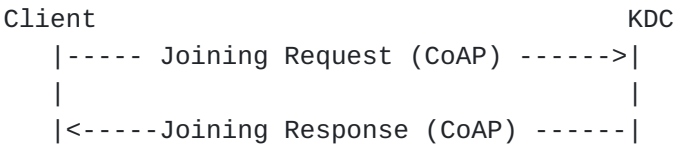

Figure 6: Join Flow

4.1.1. Join Request

After establishing a secure communication, the Client sends a Join Request to the KDC as described in Section 4.3 of [I-D.ietf-ace-key-groupcomm]. More specifically, the Client sends a POST request to the /ace-group/GROUPNAME endpoint, with Content-Format "application/ace-groupcomm+cbor". The payload MUST contain

the following information formatted as a CBOR map, which MUST be encoded as defined in Section 4.3.1 of [I-D.ietf-ace-key-groupcomm]:

* 'scope': Required. MUST be set to the specific group that the Client is attempting to join, i.e., the group name, and the roles it wishes to have in the group. This value corresponds to one scope entry, as defined in Section 3.3.1.

* 'get_creds': Optional, present if the Subcriber Client wants to retrieve the public keys of all the Publisher Clients 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 no 'role_filter' is necessary as KDC returns the authentication credentials of Publisher Clients by default.

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

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

* 'client_cred_verify': Optional, MUST be present if 'client_cred' is present. 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 client_cred', 'cnonce', 'client_cred_verify' parameters.

4.1.1.1. Client Credentials-'client_cred'

One of the following cases can occur when a new node attempts to join a pub/sub group.

* The joining node requests to join the group exclusively as a Subscriber or for Delete, i.e., it is not going to send messages to the group. In this case, the joining node is not required to provide its own authentication credential to the 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 parameters 'cnonce' and 'client_cred_verify'.

* The joining node has a Publisher role, and

  - the KDC already acquired the authentication credential of the joining node either during a past group joining process, or

during establishing a secure communication association, and the joining node and the KDC use a symmetric proof-of-possession key. If the authentication credential and the proof-of-possession key are compatible with the signature or ECDH algorithm, and possible associated parameters, then the key can be used for the authentication credential in pub/sub groups. In this case, the joining node MAY choose not to provide again its own authentication credential to the KDC, in order to limit the size of the Join Request.

- the KDC hasn't acquired an authentication credential. Then, the joining node MUST provide a compatible authentication credential in the 'client_cred' parameter of the Join Request (see Section 4.1.1).

Finally, the joining node MUST provide its own authentication credential again if it has provided the KDC with multiple authentication credentials during past joining processes intended for different pub/sub groups. If the joining node provides its own authentication credential, the KDC performs consistency checks as per Section 4.1.1 and, in case of success, considers it as the authentication credential associated with the joining node in the pub/sub group.

4.1.1.2. Proof-of-Possession

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 concatenated with N_C using the private key corresponding to the public key in the 'client_cred' paramater.

The N_S may be either:

* The challenge received from the KDC in the 'kdcchallenge' parameter of the 2.01 (Created) response to the Token Transfer Request (see Section 3.5).

* If the Publisher Client used a symmetric proof-of-possession key in the DTLS handshake [RFC9202] with the KDC, then it is an exporter value computed as defined in Section 7.5 of [RFC8446]. Specifically, N_S is exported from the DTLS session between the joining node and the KDC, using an empty 'context_value', 32 bytes as 'key_length', and the exporter label "EXPORTER-ACE-Sign-Challenge-coap-group-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, N_S MUST be this new challenge parameter.

4.1.2. Join Response

On receiving the Join Request, the KDC processes the request as defined in Section 4.3.1 of [I-D.ietf-ace-key-groupcomm], and may return a success or error response.

If 'client_cred' field is present, the KDC verifies signature in the 'client_cred_verify'. 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 already stored authentication credential from previous interactions with the joining node. The KDC verifies the PoP evidence, which is a signature, by using the public key of the joining node, as well as the signature algorithm used in the group and possible corresponding parameters.

For a Publisher Client, the KDC assigns an available Sender ID that has not been used in the group. The KDC MUST NOT assign a Sender ID to the joining node if the node doesn't have a Publisher role. The Sender ID MUST be unique, and MAY be short. ToDo: SenderID Size from groupcomm oscore? - the maximum length of Sender ID in bytes equals the length of the AEAD nonce minus 6; for AES-CCM-16-64-128 the maximum length of Sender ID is 7 bytes.

In the case of any join request error, the KDC and the Client attempting the join follow the procedure defined in Section 4.1.3.

In the case of success, the Client is added to the list of current members, if not already a member. The Client is assigned a NODENAME and a sub-resource /ace-group/GROUPNAME/nodes/NODENAME. NODENAME is associated to the access token and secure session of the Client. Publishers' client credentials are also associated with tuple containing NODENAME, GROUPNAME, sender ID and access token. The KDC responds with a Join Response with response code 2.01 (Created) if the Client has been added to the list of group members, and 2.04 (Changed) otherwise (e.g., if the Client is re-joining). The Content-Format is "application/ace-groupcomm+cbor". The payload (formatted as a CBOR map) MUST contain the following fields from the

Join Response and encode them as defined in Section 4.3.1 of [I-D.ietf-ace-key-groupcomm]:

* 'gkty': the key type "Group_PubSub_COSE_Key" for the 'key' parameter defined in Section 8.2 of this document.

* 'key': The keying material for group communication includes 'group_SenderId' if the Client is a Publisher, and a "COSE_Key". The "COSE_Key" object is defined in [RFC9052] [RFC9053] and contains:

  - 'kty' with value 4 (symmetric)

  - 'kid' with value defined by the KDC

  - 'alg' with value defined by the KDC

  - 'Base IV' with value defined by the KDC

  - 'k', the value of the symmetric key (REQ17)

* 'num': MUST be initialised to 0 as the version number of the keying material.

* 'exp', MUST be present.

* 'ace-groupcomm-profile' parameter MUST be present and has value "coap_group_pubsub_app" (PROFILE_TBD), which is defined in Section 8.1.1 of this document.

* 'creds', MUST be present, if the 'get_creds' parameter was present. Otherwise, it MUST NOT be present. The KDC provides the authentication credentials of all the Publisher Clients in the group.

* 'peer_roles' MUST be present if 'creds' is also present. Otherwise, it MUST NOT be present. (ToDo: Requested a change for this, and see how the Groupcomm draft is updated.)

* 'peer_identifiers' MUST be present if 'creds' is also present. Otherwise, it MUST NOT be present. The identifiers are the Publisher Sender IDs whose authentication credential is specified in the 'creds' parameter (REQ 25).

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

* 'kdc_nonce', MUST be present, if 'kdc_cred' is present and encoded as a CBOR byte string, and including a dedicated nonce

N_KDC generated by the KDC. For N_KDC, it is RECOMMENDED to use a 8-byte long random nonce.

* 'kdc_cred_verify' MUST be present, if 'kdc_cred' is present and encoded as a CBOR byte string. The PoP evidence is computed over the nonce N_KDC, which is specified in the 'kdc_nonce' parameter and taken as PoP input. KDC MUST compute the 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': MAY be omitted, if the KDC uses the "Point-to-Point" group rekeying scheme registered in Section 11.12 of [I-D.ietf-ace-key-groupcomm] as the default rekeying scheme in the group (OPT9). In any other case, the 'group_rekeying' parameter MUST be included.

To generate the keying material, the KDC starts at the same Base IV and Partial IV, and different keys are derived for each sender, based on their Sender ID, sent as the 'group_SenderId' inside the 'key' parameter. A Publisher Client MUST support 'group_SenderId' parameter (REQ29).

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 4.2.4). In such a case, the joining node is not able to access secure communication in the pubsub group prior its joining.

Upon receiving the Join Response, the joining node retrieves the KDC's authentication credential from the 'kdc_cred' parameter. The joining node MUST verify the proof-of-possession (PoP) evidence, which is a signature, specified in 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 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 does not include both the 'cnonce' and 'client_cred_verify' parameters.

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

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

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

* The 'scope' parameter is not present in the Join Request, or it is present and specifies any set of roles not included in the role list as defined in Section 3.3.1.

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, which encodes 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, possibly replacing the currently stored value.

On 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 Group Manager MUST return a 5.03 (Service Unavailable) response to a Publisher's join request in case there are currently no Sender IDs available.

4.2. Other Group Operations through the KDC

4.2.1. Querying for Group Information

* '/ace-group': All Clients send FETCH requests to retrieve a set of group names associated with their group identifiers. Each element of the CBOR array 'gid' is a CBOR byte string (REQ13), which encodes the Gid of the group for which the group name and the URI to the group-membership resource are provided. ToDo: Support or not?

* '/ace-group/GROUPNAME': All Clients can use GET requests 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.

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

*'/ace-group/GROUPNAME/num': All group member Clients use GET requests 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 use GET requests to retrieve the current authentication credential of the KDC.

4.2.2. Updating Authentication Credentials

A Publisher Client can contact the KDC to upload a new authentication credential to use in the group, and replace the currently stored one. To this end, it sends a CoAP POST request to the /ace-group/GROUPNAME/nodes/NODENAME/cred. The KDC replaces the stored authentication credential of this Client (identified by NODENAME) with the one specified in the request at the KDC, for the group identified by GROUPNAME.

4.2.3. Removal from a Group

A Client can actively request to leave the group. In this case, the Client sends a CoAP DELETE request to the endpoint /ace-group/GROUPNAME/nodes/NODENAME at the KDC, where GROUPNAME is the group name and NODENAME is its node name. KDC can also remove a group member due to any of the reasons described in Section 5 of [I-D.ietf-ace-key-groupcomm].

4.2.4. Rekeying a Group

KDC MUST trigger a group rekeying as described in Section 6 of [I-D.ietf-ace-key-groupcomm] due to a change in the group membership or the current group keying material approaching its expiration time. KDC MAY trigger regularly scheduled update of the group keying material.

Upon generating the new group keying material and before starting its distribution, the KDC MUST increment the version number of the group keying material. The KDC MUST preserve the current value of the Sender ID of each member in that group.

Default rekeying scheme is Point-to-point (Section 6.1 of [I-D.ietf-ace-key-groupcomm]), where KDC individually targets each

node to rekey, using the pairwise secure communication association with that node.

If the group rekeying is performed due to one or multiple Publisher Clients that have joined the group, then a rekeying message includes sender IDs, and authentication credentials that those Clients use in the group, together with their roles. This information is specified by means of the parameters 'creds', 'peer_roles' and 'peer_identifiers', like done in the Join Response message.

5. PubSub Protected Communication (C)

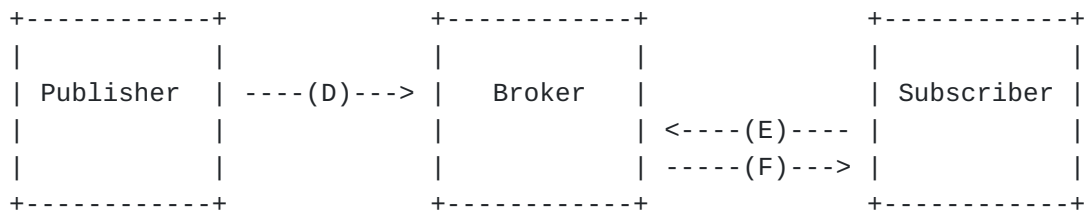

Figure 7: Secure communication between Publisher and Subscriber

(D) corresponds to the publication of a topic on the Broker, using a CoAP PUT. The publication (the resource representation) is protected with COSE ([RFC9052][RFC9053]) by the Publisher. The (E) message is the subscription of the Subscriber, and uses a CoAP GET with the Observe option set to 0 (zero) [I-D.ietf-core-coap-pubsub]. The subscription MAY be unprotected. The (F) message is the response from the Broker, where the publication is protected with COSE by the Publisher. (ToDo: Add Delete to the flow?)

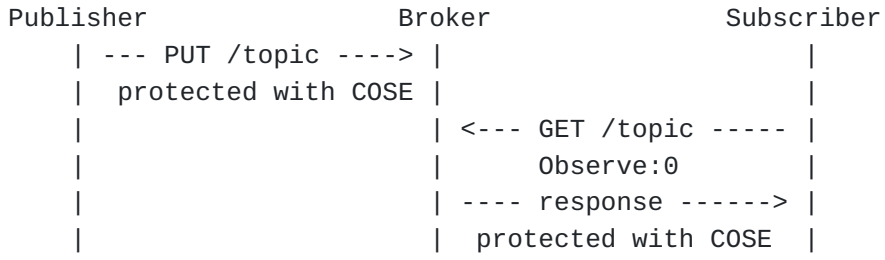

Figure 8: Example of protected communication for CoAP

5.1. Using COSE Objects To Protect The Resource Representation

The Publisher uses the symmetric COSE Key received from the KDC to protect the payload of the Publish operation (Section 4.3 of [I-D.ietf-core-coap-pubsub]). Specifically, the COSE Key is used to create a COSE_Encrypt0 object with an AEAD algorithm specified by the KDC. The AEAD key lengths, AEAD nonce length, and maximum Sender Sequence Number (Partial IV) are algorithm dependent.

The Publisher uses the private key corresponding to the public key sent to the KDC to countersign the COSE Object as specified in [RFC9052] [RFC9053]. The payload is replaced by the COSE object before the publication is sent to the Broker.

The Subscriber uses the 'kid' in the 'countersignature' field in the COSE object to retrieve the right public key to verify the countersignature. It then uses the symmetric key received from KDC to verify and decrypt the publication received in the payload from the Broker (in the case of CoAP the publication is received by the CoAP Notification).

The COSE object is constructed in the following way (as described in [RFC9052] [RFC9053]).

The protected Headers MUST contain:

- *alg, an AEAD algorithm specified by the KDC, the same as received in the symmetric COSE Key

The unprotected Headers MUST contain:

- *kid, with the value the same as in the symmetric COSE Key received

- *the Partial IV, with value a Sender Sequence Number that is incremented for every message sent. 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 IV, generated following the construction in Section 5.2 of [RFC8613] using the sender ID, Partial IV, and Base IV from the symmetric COSE Key received.

- *the counter signature

  - -the algorithm (protected),

  - -the kid, the sender ID (unprotected)

  - -the signature computed as specified in [RFC9052] [RFC9053].

- *The ciphertext, computed over the plaintext that MUST contain the message payload.

The 'external_aad' is an empty string.

The encryption and decryption operations are described in [RFC9052] [RFC9053].

6. Applicability to MQTT PubSub Profile

The steps MQTT clients go through would be similar to the CoAP clients, and the payload of the MQTT PUBLISH messages will be protected using COSE. The MQTT clients needs to use CoAP to communicate to the KDC, to join security groups, and be part of the pair-wise rekeying initiated by the KDC.

Authorisation Server (AS) Discovery is defined in Section 2.2.6.1 of [I-D.ietf-ace-mqtt-tls-profile] for MQTT v5 clients (and not supported for MQTT v3 clients). $SYS/ has been widely adopted as a prefix to topics that contain Server-specific information or control APIs, and may be used for topic and KDC discovery.

Differently for MQTT, the Client sends an authorisation request to the AS using AIF-MQTT data model for representing the requested scopes is described in Section 3 of the [I-D.ietf-ace-mqtt-tls-profile]. In the authorisation response, the 'profile' claim is set to "mqtt_pubsub_app" as defined in Section 8.1.2.

Both Publisher and Subscriber Clients MUST authorise to the Broker with their respective tokens (described in [I-D.ietf-ace-mqtt-tls-profile]) i.e., anonymous Subscribers are not supported in the profile. A Publisher Client sends PUBLISH messages for a given topic and protects the payload with the corresponding key for the associated security group. The Broker validates the PUBLISH message by verifying its topic in the stored token. A Subscriber Client may send SUBSCRIBE messages with one or multiple topic filters. A topic filter may correspond to multiple topics. The Broker validates the SUBSCRIBE message by checking the stored token for the Client. The Broker forwards all PUBLISH messages to all authorised Subscribers, including the retained messages.

7. Security Considerations

All the security considerations in [I-D.ietf-ace-key-groupcomm] apply.

In the profile described above, when the Publisher and Subscriber use asymmetric crypto, which would make the message exchange quite heavy for small constrained devices. Moreover, all Subscribers must be able to access the public keys of all the Publishers to a specific topic to verify the publications.

Even though Access Tokens have expiration times, an Access Token may need to be revoked before its expiration time (see [I-D.draft-ietf-ace-revoked-token-notification-03] for a list of possible circumstances). Clients can be excluded from future publications through re-keying for a certain topic. This could be

set up to happen on a regular basis, for certain applications. How this could be done is out of scope for this work. The method described in [I-D.draft-ietf-ace-revoked-token-notification-03] MAY be used to allow an Authorization Server to notify the KDC about revoked Access Tokens.

The Broker is only trusted with verifying that the Publisher is authorized to publish, but is not trusted with the publications itself, which it cannot read nor modify. In this setting, caching of publications on the Broker is still allowed.

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

TODO: expand on security and privacy considerations

8. IANA Considerations

8.1. ACE Groupcomm Profile Registry

The following registrations are done for the "ACE Groupcomm Profile" Registry following the procedure specified in [I-D.ietf-ace-key-groupcomm].

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

8.1.1. CoAP Profile Registration

Name: coap_group_pubsub_app

Description: Profile for delegating client authentication and authorization for publishers and subscribers in a CoAP pub/sub setting scenario in a constrained environment.

CBOR Key: TBD

Reference: [[This document]]

8.1.2. MQTT Profile Registration

Name: mqtt_pubsub_app

Description: Profile for delegating client authentication and authorization for publishers and subscribers in a MQTT pub/sub setting scenario in a constrained environment.

CBOR Key: TBD

Reference: [[This document]]

8.2. ACE Groupcomm Key Registry

The following registrations are done for the "ACE Groupcomm Key Types" defined in Section 11.7 of [I-D.ietf-ace-key-groupcomm].

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

Name: Group_PubSub_COSE_Key

Key Type Value: GROUPCOMM_KEY_TBD

Profile: coap_group_pubsub_app, defined in Section 8.1.1 of this document.

Description: COSE_Key object

References: [RFC9052] [RFC9053], [[This document]]

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 a Broker.

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: Pub/sub topic name, corresponding to the security group

*Reference: [[This document]]

*Parameter: Tperm

*Name: pubsub-perm

*Description/Specification: Permissions corresponding to the roles in pub/sub group

*Reference: [[This document]]

8.5. CoAP Content-Format

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

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

*Encoding: -

*ID: 294 (suggested)

*Reference: [RFC-XXXX]

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

*Encoding: -

*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-coap-group-pubsub-app

*DTLS-OK: Y

*Recommended: N

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

9. References

9.1. Normative References

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

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

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

[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>.

[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>.

[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>.

[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>.

[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>.

[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>.

9.2. Informative References

[I-D.draft-ietf-ace-revoked-token-notification-03] Tiloca, M., Seitz, L., 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-03, 24 October 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-ace-revoked-token-notification-03>.

[I-D.ietf-ace-key-groupcomm-oscore] Tiloca, M., Park, J., and F. Palombini, "Key Management for OSCORE Groups in ACE", Work in Progress, Internet-Draft, draft-ietf-ace-key-groupcomm-oscore-16, 6 March 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-ace-key-groupcomm-oscore-16>.

[I-D.ietf-ace-mqtt-tls-profile] Sengul, C. and A. Kirby, "Message Queuing Telemetry Transport (MQTT)-TLS profile of Authentication and Authorization for Constrained

Environments (ACE) Framework", Work in Progress, Internet-Draft, draft-ietf-ace-mqtt-tls-profile-17, 23 March 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-ace-mqtt-tls-profile-17>.

[MQTT-OASIS-Standard-v5] Banks, A., Briggs, E., Borgendale, K., and R. Gupta, "OASIS Standard MQTT Version 5.0", 2017, <http://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html>.

[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>.

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

*REQ1: Specify the format and encoding of 'scope'. : See Section 3.3.1.

*REQ2: If the AIF format of 'scope' is used, register its specific instance of "Toid" and "Tperm" as Media Type parameters and a corresponding Content-Format, as per the guidelines in [RFC9237].:See Section 8.4.

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

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

*REQ5: If used, specify the acceptable values for 'sign_key_parameters' : 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].

*REQ6: Specify the acceptable formats for authentication credentials and, if used, the acceptable values for 'cred_fmt': Acceptable formats explicitly provide the public key as well as the comprehensive set of information related to the public key algorithm. Takes value 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 and if this has to be provided through the 'kdc_cred' parameter : Optional, see Section 4.1.2 of this document.

*REQ9: Specify if any part of the KDC interface as defined in [I-D.ietf-ace-key-groupcomm] is not supported by the KDC: Some left optional, see Section 4 of this document.

*REQ10: Register a Resource Type for the root url-path, which is used to discover the correct url to access at the KDC : the Resource Type (rt=) Link Target Attribute value "core.ps.gm" is registered in Section Section 8.3. ToDo: This possibly will not stay as the final method, KDC discovery done differently through topic discovery?

*REQ11: Define what specific actions (e.g., CoAP methods) are allowed on each resource provided by the KDC interface, depending on whether the Client is a current group member; the roles that a Client is authorized to take as per the obtained access token; and the roles that the Client has as current group member.: See Section 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 identifier: CBOR byte string, see Section 4.2.1.

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

*REQ15: Specify how the nonce N_S is generated, if the token is not provided to the KDC through the Token Transfer Request to the authz-info endpoint (e.g., if it is used directly to validate TLS instead): See Section 4.1.1.2 in this document.

*REQ16: Define the initial value of the 'num' parameter: The initial value MUST be set to 0.

*REQ17: Specify the format of the 'key' parameter: See Section 4.1.2.

*REQ18: Specify the acceptable values of the 'gkty' parameter: Group_PubSub_COSE_Key, see Section 8.2.

*REQ19: Specify and register the application profile identifier: coap_group_pubsub_app, see Section 8.1.1.

*REQ20: If used, specify the format and content of 'group_policies' and its entries. Specify the policies default values: ToDo.

*REQ21: Specify the approaches used to compute and verify the PoP evidence to include in 'kdc_cred_verify', and which of those approaches is used in which case: see Section 4.1.2.

*REQ22: Specify the communication protocol the members of the group must use.: CoAP [RFC7252], and for pub/sub communication [I-D.ietf-core-coap-pubsub]

*REQ23: Specify the security protocol the group members must use to protect their communication. This must provide encryption, integrity and replay protection.: Symmetric COSE Key is used to create a COSE_Encrypt0 object with an AEAD algorithm specified by the KDC.

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

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

*REQ26: Specify policies at the KDC to handle ids that are not included in 'get_creds'.: See Section 4.2.1.

*REQ27: Specify the format of newly-generated individual keying material for group members, or of the information to derive it, and corresponding CBOR label.: Not applicable.

*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 3.4 and Section 8.5 of this document.

*REQ29: Categorize newly defined parameters according to the same criteria of Section 8 of [I-D.ietf-ace-key-groupcomm].: 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 [I-D.ietf-ace-key-groupcomm], and under which circumstances.: A Publisher Client MUST support client_cred', 'cnonce', 'client_cred_verify' parameters; see Section 4.1.1. A Publisher Client that provides the token to the KDC, through the authz-info endpoint, MUST support the parameter 'kdcchallenge'; see Section 3.5.

*OPT1: Optionally, if the textual format of 'scope' is used, specify CBOR values to use for abbreviating the role identifiers in the group: No.

*OPT2: Optionally, specify the additional parameters used in the exchange of Token Transfer Request and Response : No.

*OPT3: Optionally, specify the negotiation of parameter values for signature algorithm and signature keys, if 'sign_info' is not used: See Section 3.5.

*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' field in an error response from the KDC: No.

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

*OPT7: Optionally, specify the functionalities implemented at the 'control_uri' resource hosted at the Client, including message exchange encoding and other details.: No.

*OPT8: Optionally, specify the behavior of the handler in case of failure to retrieve an authentication credential for the specific

node: 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 11.12 of [I-D.ietf-ace-key-groupcomm].

*OPT10: Optionally, specify the functionalities implemented at the 'control_group_uri' resource hosted at the Client, including message exchange encoding and other details. : No.

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

*OPT12: Optionally, specify for the KDC to perform group rekeying (together or instead of renewing individual keying material) when receiving a Key Renewal Request: ToDo.

*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: ToDo.

*OPT15: Optionally, specify if Clients must or should support any of the parameters defined as optional in [I-D.ietf-ace-key-groupcomm]: No.

Acknowledgments

The author wishes to thank Ari Keränen, John Preuß Mattsson, Ludwig Seitz, Göran Selander, and Jim Schaad for the useful discussion and reviews that helped shape this document.

The work on this document has been partly supported 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