

ACE Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 9, 2019

F. Palombini
Ericsson AB
M. Tiloca
RISE AB
March 08, 2019

Key Provisioning for Group Communication using ACE
draft-ietf-ace-key-groupcomm-01

Abstract

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

Status of This Memo

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 September 9, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1.	Introduction	2
1.1.	Terminology	3
2.	Overview	3
3.	Authorization to Join a Group	5
3.1.	Authorization Request	6
3.2.	Authorization Response	7
3.3.	Token Post	8
4.	Key Distribution	8
4.1.	Key Distribution Request	9
4.2.	Key Distribution Response	10
5.	Removal of a Node from the Group	12
5.1.	Expired Authorization	12
5.2.	Request to Leave the Group	12
6.	Retrieval of Updated Keying Material	13
6.1.	Key Re-Distribution Request	14
6.2.	Key Re-Distribution Response	14
7.	Retrieval of Public Keys for Group Members	15
7.1.	Public Key Request	15
7.2.	Public Key Response	16
8.	Security Considerations	16
9.	IANA Considerations	16
9.1.	ACE Groupcomm Key Registry	16
9.2.	ACE Groupcomm Profile Registry	17
9.3.	Expert Review Instructions	18
10.	References	18
10.1.	Normative References	18
10.2.	Informative References	19
Appendix A.	Document Updates	20
A.1.	Version -00 to -01	20
	Acknowledgments	20
	Authors' Addresses	21

[1.](#) Introduction

This document expands the ACE framework [[I-D.ietf-ace-oauth-authz](#)] to define the format of messages used to request, distribute and renew the keying material in a group communication scenario, e.g. based on multicast [[RFC7390](#)] or on publishing-subscribing [[I-D.ietf-core-coap-pubsub](#)].

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

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

1.1. Terminology

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

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

2. Overview

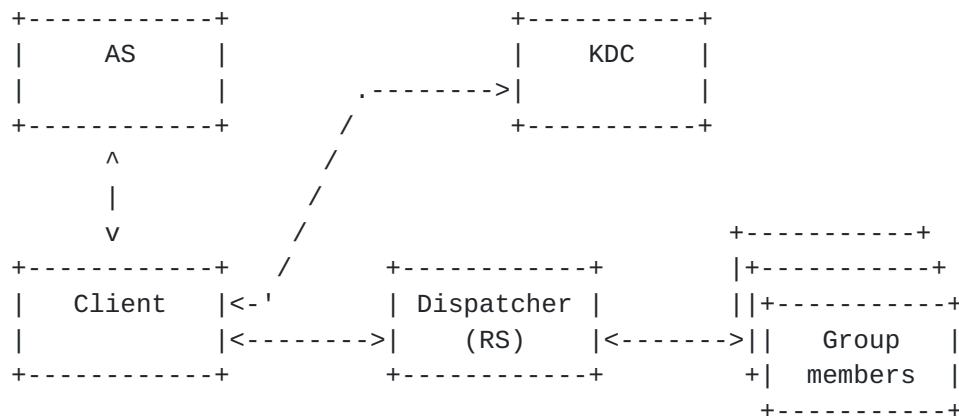


Figure 1: Key Distribution Participants

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

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

- o Authorization Server (AS): same as AS in the ACE Framework; it enforces access policies, and knows if a node is allowed to join the group with write and/or read rights.
- o Key Distribution Center (KDC): maintains the keying material to protect group communications, and provides it to Clients authorized to join the group. During the first part of the exchange ([Section 3](#)), it takes the role of the RS in the ACE Framework. During the second part ([Section 4](#)), which is not based on the ACE Framework, it distributes the keying material. In addition, it provides the latest keying material to group members when requested. If required by the application, the KDC renews and re-distributes the keying material in the group when membership changes.
- o Dispatcher: entity through which the Clients communicate with the group and which distributes messages to the group members. Examples of dispatchers are: the Broker node in a pub-sub setting; a relay node for group communication that delivers group messages as multiple unicast messages to all group members; an implicit entity as in a multicast communication setting, where messages are transmitted to a multicast IP address and delivered on the transport channel.

This document specifies the message flows and formats for:

- o Authorizing a new node to join the group ([Section 3](#)), and providing it with the group keying material to communicate with the other group members ([Section 4](#)).
- o Removing of a current member from the group ([Section 5](#)).
- o Retrieving keying material as a current group member ([Section 6](#) and [Section 7](#)).
- o Renewing and re-distributing the group keying material (rekeying) upon a membership change in the group ([Section 4.2](#) and [Section 5](#)).

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

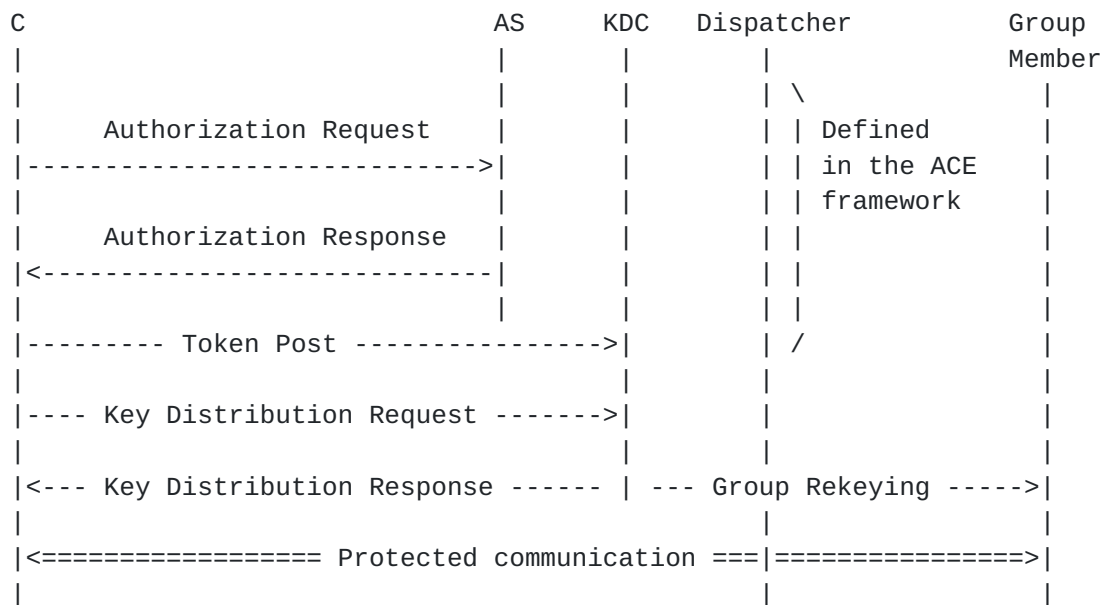


Figure 2: Message Flow Upon New Node's Joining

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

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

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

All further communications between a Client and the other group members MUST be secured using the keying material provided in [Section 4](#).

3. Authorization to Join a Group

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

As defined in [[I-D.ietf-ace-oauth-authz](#)], the Client requests from the AS an authorization to join the group through the KDC (see [Section 3.1](#)). If the request is approved and authorization is granted, the AS provides the Client with a proof-of-possession access token and parameters to securely communicate with the KDC (see

[Section 3.2](#)). Communications between the Client and the AS MUST be secured, and depends on the profile of ACE used.

Figure 3 gives an overview of the exchange described above.



Figure 3: Message Flow of Join Authorization

3.1. Authorization Request

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

- o 'grant_type', with value "client_credentials".

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

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

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

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

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

3.2. Authorization Response

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

- o 'access_token', containing the proof-of-possession access token.
- o 'cnf' if symmetric keys are used, not present if asymmetric keys are used. This parameter is defined in Section 3.2 of [\[I-D.ietf-ace-oauth-params\]](#) and contains the symmetric proof-of-possession key that the Client is supposed to use with the KDC.
- o 'rs_cnf' if asymmetric keys are used, not present if symmetric keys are used. This parameter is as defined in Section 3.2 of [\[I-D.ietf-ace-oauth-params\]](#) and contains information about the public key of the KDC.
- o 'exp', contains the lifetime in seconds of the access token. This parameter MAY be omitted if the application defines how the expiration time is communicated to the Client via other means, or if it establishes a default value.

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

- o 'scope', which mirrors the 'scope' parameter in the Authorization Request (see [Section 3.1](#)). Its value is a CBOR array encoded as a byte string, containing:
 - * As first element, the identifier of the specific group or topic the Client is authorized to access.
 - * Optionally, as second element, the role (or CBOR array of roles) the Client is authorized to take in the group.

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

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

The access token MUST contain all the parameters defined above (including the same 'scope' as in this message, if present, or the

'scope' of the Authorization Request otherwise), and additionally other optional parameters the profile requires.

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

3.3. Token Post

The Client sends a CoAP POST request including the access token to the KDC, as specified in section 5.8.1 of [[I-D.ietf-ace-oauth-authz](#)]. If the specific ACE profile defines it, the Client MAY use a different endpoint than /authz-info at the KDC to post the access token to. After successful verification, the Client is authorized to receive the group keying material from the KDC and join the group.

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

4. Key Distribution

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

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

During this exchange, the Client sends a request to the AS, specifying the group it wishes to join (see [Section 4.1](#)). Then, the KDC verifies the access token and that the Client is authorized to join that group; if so, it provides the Client with the keying material to securely communicate with the member of the group (see [Section 4.2](#)).

Figure 4 gives an overview of the exchange described above.

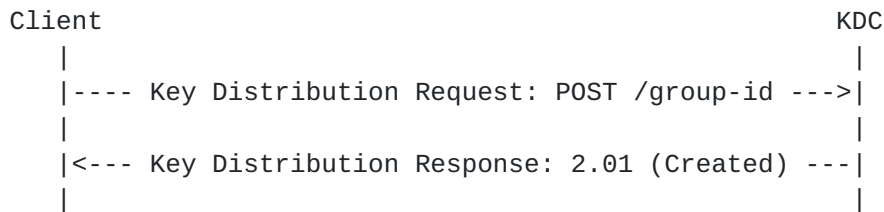


Figure 4: Message Flow of Key Distribution to a New Group Member

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

- o The Client wishes to (re-)get the current keying material, for cases such as expiration, loss or suspected mismatch, due to e.g. reboot or missed group rekeying. This is further discussed in [Section 6](#).
- o The Client wishes to (re-)get the public keys of other group members, e.g. if it is aware of new nodes joining the group after itself. This is further discussed in [Section 7](#).

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

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

[4.1](#). Key Distribution Request

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

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

- o 'get_pub_keys', if the Client wishes to receive the public keys of the other nodes in the group from the KDC. The value is an empty CBOR Array. This parameter may be present if the KDC stores the public keys of the nodes in the group and distributes them to the Client; it is useless to have here if the set of public keys of the members of the group is known in another way, e.g. it was provided by the AS.
- o 'client_cred', with value the public key or certificate of the Client. If the KDC is managing (collecting from/distributing to the Client) the public keys of the group members, this field contains the public key of the Client.
- o 'pub_keys_repos', can be present if a certificate is present in the 'client_cred' field, with value a list of public key repositories storing the certificate of the Client.

4.2. Key Distribution Response

The KDC verifies the 'scope' received in the Key Distribution Request, if present, against the 'scope' stored in the access token associated to this client. If verification fails, the KDC MUST respond with a 4.01 (Unauthorized) error message. If the Key Distribution Request is not formatted correctly (e.g. no 'scope' field present while expected, or unknown fields present), the KDC MUST respond with 4.00 (Bad Request) error message.

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

- o 'kty', identifying the key type of the 'key' parameter. The set of values can be found in the "Key Type" column of the "ACE Groupcomm Key" Registry. Implementations MUST verify that the key type matches the profile being used, if present, as registered in the "ACE Groupcomm Key" registry.
- o 'key', containing the keying material for the group communication, or information required to derive it.

The exact format of the 'key' value MUST be defined in applications of this specification. Additionally, documents specifying the key format MUST register it in the "ACE Groupcomm Key" registry, including its name, type and profile to be used with, as defined in the "ACE Groupcomm Key" registry, defined in [Section 9.1](#).

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

Figure 5: Key Type Values

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

- o 'profile', with value an identifier that MUST be used to uniquely identify itself. The identifier MUST be registered in the "ACE Groupcomm Profile" Registry.
- o 'exp', with value the expiration time of the keying material for the group communication, encoded as a CBOR unsigned integer or floating-point number.
- o 'pub_keys', may only be present if 'get_pub_keys' was present in the Key Distribution Request. This parameter is a CBOR Byte String, which encodes the public keys of all the group members paired with the respective member identifiers. In case public keys in the group are represented as COSE Keys, the CBOR Byte String encodes a COSE_KeySet (see [RFC8152]), which contains the public keys of all the members of the group. In particular, each COSE Key in the COSE_KeySet includes the identifier of the corresponding group member as value of its 'kid' key parameter. Alternative specific encodings of this parameter MUST be defined in applications of this specification.
- o 'group_policies', with value a list of parameters indicating how the group handles specific management aspects. This includes, for instance, approaches to achieve synchronization of sequence numbers among group members. The exact format of this parameter is specific to the profile.
- o 'mgt_key_material', with value the administrative keying material to participate in the group rekeying performed by the KDC. The exact format and content depend on the specific rekeying scheme used in the group, which may be specified in the profile.

Specific profiles need to specify how exactly the keying material is used to protect the group communication.

If the application requires backward security, the KDC SHALL generate new group keying material and securely distribute it to all the current group members, using the message format defined in this

section. Application profiles may define alternative message formats.

5. Removal of a Node from the Group

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

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

5.1. Expired Authorization

If the node is not authorized anymore, the AS can directly communicate that to the KDC. Alternatively, the access token might have expired. If Token introspection is provided by the AS, the KDC can use it as per Section 5.7 of [[I-D.ietf-ace-oauth-authz](#)], in order to verify that the access token is still valid.

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

5.2. Request to Leave the Group

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

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

- o 'leave', with value an empty CBOR array.
- o 'scope', with value the specific resource that the Client is authorized to access (i.e. group or topic identifier) and wants to leave, encoded as in [Section 3.1](#). The 'role' field is omitted.

Additionally, the Leave request MAY contain the following parameters, which, if included, MUST have the corresponding values:

- o 'client_cred', with value the identifier of the public key or certificate of the Client. This field is used if the KDC is managing (collecting from/distributing to the Client) the public keys of the group members.

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

If the Leave Request is not formatted correctly (e.g. no 'scope' field present, or unknown fields present), the KDC MUST respond with a 4.00 (Bad Request) error message. Otherwise, the KDC MUST remove the leaving node from the list of group members, if the KDC keeps track of that.

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

6. Retrieval of Updated Keying Material

A node stops using the group keying material upon its expiration, according to the 'exp' parameter specified in the retained COSE Key. Then, if it wants to continue participating in the group communication, the node has to request new updated keying material to the KDC.

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

Note that policies can be set up so that the Client sends a request to the KDC only after a given number of unsuccessfully decrypted incoming messages.

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

- o Can maintain an Observable resource to send notifications to Clients when the keying material is updated. Such a notification

would have the same payload as the Key Re-Distribution Response defined in [Section 6.2](#).

- o Can send the payload of the Key Re-Distribution Response in a multicast request to the members of the group.
- o Can send unicast requests to each Client over a secure channel, with the Key Re-Distribution Response as payload.
- o Can act as a publisher in a pub-sub scenario, and update the keying material by publishing on a specific topic on a broker, which all the members of the group are subscribed to.

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

6.1. Key Re-Distribution Request

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

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

6.2. Key Re-Distribution Response

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

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

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

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

7. Retrieval of Public Keys for Group Members

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

Figure 6 gives an overview of the exchange described above.

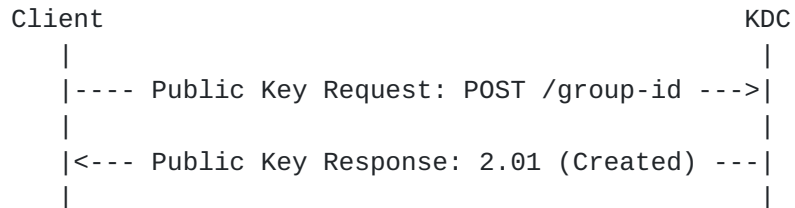


Figure 6: Message Flow of Public Key Request-Response

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

7.1. Public Key Request

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

- o 'get_pub_keys', which has as value a CBOR array including either:
 - * no elements, i.e. an empty array, in order to request the public key of all current group members; or
 - * N elements, each of which is the identifier of a group member, in order to request the public key of the specified nodes.
- o 'scope', which contains only the identifier of the specific group or topic, encoded as in [Section 3.1](#). That is, the role field is not present.

7.2. Public Key Response

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

- o 'pub_keys', which contains either:
 - * the public keys of all the members of the group, if the 'get_pub_keys' parameter of the Public Key request was an empty array; or
 - * the public keys of the group members with the identifiers specified in the 'get_pub_keys' parameter of the Public Key request.

The KDC ignores possible identifiers included in the 'get_pub_keys' parameter of the Public Key request if they are not associated to any current group member.

8. Security Considerations

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

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

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

9. IANA Considerations

This document has the following actions for IANA.

9.1. ACE Groupcomm Key Registry

This specification establishes the IANA "ACE Groupcomm Key" Registry. The Registry has been created to use the "Expert Review Required" registration procedure [[RFC8126](#)]. Expert review guidelines are provided in [Section 9.3](#).

The columns of this Registry are:

- o Name: This is a descriptive name that enables easier reference to the item. The name **MUST** be unique. It is not used in the encoding.
- o Key Type Value: This is the value used to identify the keying material. These values **MUST** be unique. The value can be a positive integer, a negative integer, or a string.
- o Profile: This field may contain a descriptive string of a profile to be used with this item. This should be a value that is in the Name column of the "ACE Groupcomm Profile" Registry.
- o Description: This field contains a brief description of the keying material.
- o References: This contains a pointer to the public specification for the format of the keying material, if one exists.

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

9.2. ACE Groupcomm Profile Registry

This specification establishes the IANA "ACE Groupcomm Profile" Registry. The Registry has been created to use the "Expert Review Required" registration procedure [[RFC8126](#)]. Expert review guidelines are provided in [Section 9.3](#). It should be noted that, in addition to the expert review, some portions of the Registry require a specification, potentially a Standards Track RFC, be supplied as well.

The columns of this Registry are:

- o Name: The name of the profile, to be used as value of the profile attribute.
- o Description: Text giving an overview of the profile and the context it is developed for.
- o CBOR Value: CBOR abbreviation for this profile name. Different ranges of values use different registration policies [[RFC8126](#)]. Integer values from -256 to 255 are designated as Standards Action. Integer values from -65536 to -257 and from 256 to 65535 are designated as Specification Required. Integer values greater than 65535 are designated as Expert Review. Integer values less than -65536 are marked as Private Use.

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

9.3. Expert Review Instructions

The IANA Registries established in this document are defined as expert review. This section gives some general guidelines for what the experts should be looking for, but they are being designated as experts for a reason so they should be given substantial latitude.

Expert reviewers should take into consideration the following points:

- o Point squatting should be discouraged. Reviewers are encouraged to get sufficient information for registration requests to ensure that the usage is not going to duplicate one that is already registered and that the point is likely to be used in deployments. The zones tagged as private use are intended for testing purposes and closed environments, code points in other ranges should not be assigned for testing.
- o Specifications are required for the standards track range of point assignment. Specifications should exist for specification required ranges, but early assignment before a specification is available is considered to be permissible. Specifications are needed for the first-come, first-serve range if they are expected to be used outside of closed environments in an interoperable way. When specifications are not provided, the description provided needs to have sufficient information to identify what the point is being used for.
- o Experts should take into account the expected usage of fields when approving point assignment. The fact that there is a range for standards track documents does not mean that a standards track document cannot have points assigned outside of that range. The length of the encoded value should be weighed against how many code points of that length are left, the size of device it will be used on, and the number of code points left that encode to that size.

10. References

10.1. Normative References

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

Seitz, L., Selander, G., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "Authentication and Authorization for Constrained Environments (ACE) using the OAuth 2.0 Framework (ACE-OAuth)", [draft-ietf-ace-oauth-authz-22](#) (work in progress), March 2019.

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

Seitz, L., "Additional OAuth Parameters for Authorization in Constrained Environments (ACE)", [draft-ietf-ace-oauth-params-04](#) (work in progress), February 2019.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 8126](#), DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.

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

10.2. Informative References**[I-D.ietf-core-coap-pubsub]**

Koster, M., Keranen, A., and J. Jimenez, "Publish-Subscribe Broker for the Constrained Application Protocol (CoAP)", [draft-ietf-core-coap-pubsub-06](#) (work in progress), January 2019.

[I-D.ietf-core-object-security]

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

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

- [RFC2094] Harney, H. and C. Muckenhirn, "Group Key Management Protocol (GKMP) Architecture", [RFC 2094](#), DOI 10.17487/RFC2094, July 1997, <<https://www.rfc-editor.org/info/rfc2094>>.
- [RFC2627] Wallner, D., Harder, E., and R. Agee, "Key Management for Multicast: Issues and Architectures", [RFC 2627](#), DOI 10.17487/RFC2627, June 1999, <<https://www.rfc-editor.org/info/rfc2627>>.
- [RFC7390] Rahman, A., Ed. and E. Dijk, Ed., "Group Communication for the Constrained Application Protocol (CoAP)", [RFC 7390](#), DOI 10.17487/RFC7390, October 2014, <<https://www.rfc-editor.org/info/rfc7390>>.

Appendix A. Document Updates

RFC EDITOR: PLEASE REMOVE THIS SECTION.

A.1. Version -00 to -01

- o Changed name of 'req_aud' to 'audience' in the Authorization Request ([Section 3.1](#)).
- o Defined error handling on the KDC (Sections [4.2](#) and [6.2](#)).
- o Updated format of the Key Distribution Response as a whole ([Section 4.2](#)).
- o Generalized format of 'pub_keys' in the Key Distribution Response ([Section 4.2](#)).
- o Defined format for the message to request leaving the group ([Section 5.2](#)).
- o Mentioned methods for group rekeying initiated by the KDC ([Section 6](#)).
- o Added security consideration on replay protection ([Section 8](#)).
- o New IANA registries "ACE Groupcomm Key Registry" and "ACE Groupcomm Profile Registry" ([Section 9](#)).

Acknowledgments

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

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

Authors' Addresses

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

Email: francesca.palombini@ericsson.com

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

Email: marco.tiloca@ri.se

