ACE Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 6, 2020

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

July 05, 2019

Key Management for OSCORE Groups in ACE
draft-ietf-ace-key-groupcomm-oscore-02

Abstract

This document describes a method to request and provision keying material in group communication scenarios where the group communication is based on CoAP and secured with Object Security for Constrained RESTful Environments (OSCORE). The proposed method delegates the authentication and authorization of new client nodes that join an OSCORE group through a Group Manager server. This approach builds on the ACE framework for Authentication and Authorization, and leverages protocol-specific transport profiles of ACE to achieve communication security, proof-of-possession and server authentication.

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 January 6, 2020.

Table of Contents

1. Introduction

Object Security for Constrained RESTful Environments (OSCORE) [I-D.ietf-core-object-security] is a method for application-layer protection of the Constrained Application Protocol (CoAP) [RFC7252], using CBOR Object Signing and Encryption (COSE) [RFC8152] and enabling end-to-end security of CoAP payload and options.

As described in [I-D.ietf-core-oscore-groupcomm], OSCORE may be used to protect CoAP group communication over IP multicast [RFC7390][I-D.dijk-core-groupcomm-bis]. This relies on a Group Manager, which is responsible for managing an OSCORE group, where members exchange CoAP messages secured with OSCORE. The Group Manager can be responsible for multiple groups, coordinates the join process of new group members, and is entrusted with the distribution and renewal of group keying material.

This specification builds on the ACE framework for Authentication and Authorization [I-D.ietf-ace-oauth-authz] and defines a method to:

- Authorize a node to join an OSCORE group, and provide it with the group keying material to communicate with other group members.

- Provide updated keying material to group members upon request.

- Renew the group keying material and distribute it to the OSCORE group (rekeying) upon changes in the group membership.

A client node joins an OSCORE group through a resource server acting as Group Manager for that group. The join process relies on an Access Token, which is bound to a proof-of-possession key and authorizes the client to access a specific join resource at the Group Manager.

Messages exchanged among the participants follow the formats defined in [I-D.ietf-ace-key-groupcomm] for provisioning and renewing keying material in group communication scenarios.

In order to achieve communication security, proof-of-possession and server authentication, the client and the Group Manager leverage protocol-specific transport profiles of ACE. These include also possible forthcoming transport profiles that comply with the requirements in Appendix C of [I-D.ietf-ace-oauth-authz].

1.1. Terminology

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

Readers are expected to be familiar with the terms and concepts described in the ACE framework for authentication and authorization [I-D.ietf-ace-oauth-authz]. The terminology for entities in the considered architecture is defined in OAuth 2.0 [RFC6749]. In particular, this includes Client (C), Resource Server (RS), and Authorization Server (AS).

Readers are expected to be familiar with the terms and concepts related to the CoAP protocol described in [RFC7252][RFC7390][I-D.dijk-core-groupcomm-bis]. Note that, unless otherwise indicated, the term "endpoint" is used here following its OAuth definition, aimed at denoting resources such as /token and /introspect at the AS and /authz-info at the RS. This document does not use the CoAP definition of "endpoint", which is "An entity participating in the CoAP protocol".

Readers are expected to be familiar with the terms and concepts for protection and processing of CoAP messages through OSCORE [I-D.ietf-core-object-security] also in group communication scenarios [I-D.ietf-core-oscore-groupcomm]. These include the concept of Group Manager, as the entity responsible for a set of groups where communications are secured with OSCORE. In this specification, the Group Manager acts as Resource Server.

This document refers also to the following terminology.

- Joining node: a network node intending to join an OSCORE group, where communication is based on CoAP [RFC7390][I-D.dijk-core-groupcomm-bis] and secured with OSCORE as described in [I-D.ietf-core-oscore-groupcomm].

- Join process: the process through which a joining node becomes a member of an OSCORE group. The join process is enforced and assisted by the Group Manager responsible for that group.

- Join resource: a resource hosted by the Group Manager, associated to an OSCORE group under that Group Manager. A join resource is identifiable with the Group Identifier (Gid) of the respective group. A joining node accesses a join resource to start the join

process and become a member of that group. The URI of a join resource is fixed.

- Join endpoint: an endpoint at the Group Manager associated to a join resource.

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

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

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

- Group rekeying process: the process through which the Group Manager renews the security parameters and group keying material, and (re-)distributes them to the OSCORE group members.

1.2. Relation to Other Documents

Figure 1 overviews the main documents related to this specification. Arrows and asterisk-arrows denote normative references and informative refences, respectively.

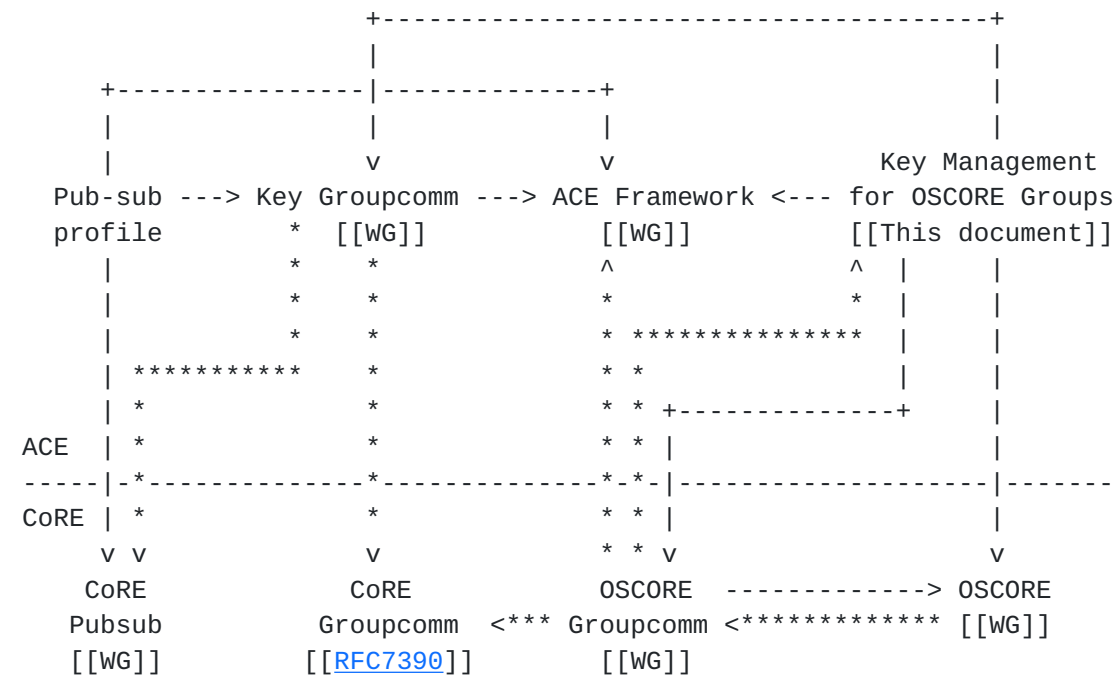

Figure 1: Related Documents

2. Protocol Overview

Group communication for CoAP over IP multicast has been enabled in [RFC7390][I-D.dijk-core-groupcomm-bis] and can be secured with Object Security for Constrained RESTful Environments (OSCORE) [I-D.ietf-core-object-security] as described in [I-D.ietf-core-oscore-groupcomm]. A network node joins an OSCORE group by interacting with the responsible Group Manager. Once registered in the group, the new node can securely exchange messages with other group members.

This specification describes how to use the ACE framework for authentication and authorization [I-D.ietf-ace-oauth-authz] to:

- Enable a node to join an OSCORE group through the Group Manager and receive the security parameters and keying material to communicate with the other members of the gorup.

- Enable members of OSCORE groups to retrieve updated group keying material from the Group Manager.

- Enable the Group Manager to renew the security parameters and group keying material, and to (re-)distribute them to the members of the OSCORE group (rekeying).

With reference to the ACE framework and the terminology defined in OAuth 2.0 [RFC6749]:

- The Group Manager acts as Resource Server (RS), and hosts one join resource for each OSCORE group it manages. Each join resource is exported by a distinct join endpoint. During the join process, the Group Manager provides joining nodes with the parameters and keying material for taking part to secure communications in the OSCORE group. The Group Manager also maintains the group keying material and performs the group rekeying process to distribute updated keying material to the group members.

- The joining node acts as Client (C), and requests to join an OSCORE group by accessing the related join endpoint at the Group Manager.

- The Authorization Server (AS) authorizes joining nodes to join OSCORE groups under their respective Group Manager. Multiple Group Managers can be associated to the same AS. The AS MAY release Access Tokens for other purposes than joining OSCORE groups under registered Group Managers. For example, the AS may also release Access Tokens for accessing resources hosted by members of OSCORE groups.

All communications between the involved entities rely on the CoAP protocol and MUST be secured.

In particular, communications between the joining node and the Group Manager leverage protocol-specific transport profiles of ACE to achieve communication security, proof-of-possession and server authentication. To this end, the AS must signal the specific transport profile to use, consistently with requirements and assumptions defined in the ACE framework [I-D.ietf-ace-oauth-authz].

With reference to the AS, communications between the joining node and the AS (/token endpoint) as well as between the Group Manager and the AS (/introspect endpoint) can be secured by different means, for instance using DTLS [RFC6347] or OSCORE [I-D.ietf-core-object-security]. Further details on how the AS secures communications (with the joining node and the Group Manager) depend on the specifically used transport profile of ACE, and are out of the scope of this specification.

2.1. Overview of the Join Process

A node performs the following steps in order to join an OSCORE group. Messages exchanged among the participants follow the formats defined in [I-D.ietf-ace-key-groupcomm], and are further specified in

Section 3 and Section 4 of this document. The Group Manager acts as the Key Distribution Center (KDC) defined in [I-D.ietf-ace-key-groupcomm].

1. The joining node requests an Access Token from the AS, in order to access a join resource on the Group Manager and hence join the associated OSCORE group (see Section 3). The joining node will start or continue using a secure communication channel with the Group Manager, according to the response from the AS.

2. The joining node transfers authentication and authorization information to the Group Manager by posting the obtained Access Token (see Section 4). After that, a joining node must have a secure communication channel established with the Group Manager, before starting to join an OSCORE group under that Group Manager (see Section 4). Possible ways to provide a secure communication channel are DTLS [RFC6347] and OSCORE [I-D.ietf-core-object-security].

3. The joining node starts the join process to become a member of the OSCORE group, by accessing the related join resource hosted by the Group Manager (see Section 4).

4. At the end of the join process, the joining node has received from the Group Manager the parameters and keying material to securely communicate with the other members of the OSCORE group.

5. The joining node and the Group Manager maintain the secure channel, to support possible future communications.

All further communications between the joining node and the Group Manager MUST be secured, for instance with the same secure channel mentioned in step 2.

2.2. Overview of the Group Rekeying Process

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

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

Parameters and keying material include a new Group Identifier (Gid) for the group and a new Master Secret for the OSCORE Common Security Context of that group (see Section 2 of [I-D.ietf-core-oscore-groupcomm]).

The Group Manager MUST support the Group Rekeying Process described in Section 7. Future application profiles may define alternative message formats and distribution schemes to perform group rekeying.

3. Joining Node to Authorization Server

This section describes how the joining node interacts with the AS in order to be authorized to join an OSCORE group under a given Group Manager. In particular, it considers a joining node that intends to contact that Group Manager for the first time.

The message exchange between the joining node and the AS consists of the messages Authorization Request and Authorization Response defined in Section 3 of [I-D.ietf-ace-key-groupcomm].

In case the specific AS associated to the Group Manager is unknown to the joining node, the latter can rely on mechanisms like the Unauthorized Resource Request message described in Section 5.1.1 of [I-D.ietf-ace-oauth-authz] to discover the correct AS to contact.

3.1. Authorization Request

The joining node contacts the AS, in order to request an Access Token for accessing the join resource hosted by the Group Manager and associated to the OSCORE group. The Access Token request sent to the /token endpoint follows the format of the Authorization Request message defined in Section 3.1 of [I-D.ietf-ace-key-groupcomm]. In particular:

- The 'scope' parameter MUST be present and MUST include:
  - in the first element, either the Group Identifier (Gid) of the group to join under the Group Manager, or a value from which the Group Manager can derive the Gid of the group to join. It is up to the application to define how the Group Manager possibly performs the derivation of the full Gid. Appendix C of [I-D.ietf-core-oscore-groupcomm] provides an example of structured Gid, composed of a fixed part, namely Group Prefix, and a variable part, namely Group Epoch.
  - in the second element, the role (encoded as a text string) or CBOR array of roles that the joining node intends to have in the group it intends to join. Accepted values of roles are:

"requester", "responder", and "monitor". Possible combinations are: ["requester" , "responder"]; ["requester" , "monitor"].

- The 'audience' parameter MUST be present and is set to the identifier of the Group Manager.

3.2. Authorization Response

The AS is responsible for authorizing the joining node to join specific OSCORE groups, according to join policies enforced on behalf of the respective Group Manager.

In case of successful authorization, the AS releases an Access Token bound to a proof-of-possession key associated to the joining node.

Then, the AS provides the joining node with the Access Token as part of an Access Token response, which follows the format of the Authorization Response message defined in Section 3.2 of [I-D.ietf-ace-key-groupcomm].

The 'exp' parameter MUST be present. Other means for the AS to specify the lifetime of Access Tokens are out of the scope of this specification.

The AS must include the 'scope' parameter in the response when the value included in the Access Token differs from the one specified by the joining node in the request. In such a case, the second element of 'scope' MUST be present and includes the role or CBOR array of roles that the joining node is actually authorized to take in the group, encoded as specified in Section 3.1 of this document.

Also, the 'profile' parameter indicates the specific transport profile of ACE to use for securing communications between the joining node and the Group Manager (see Section 5.6.4.3 of [I-D.ietf-ace-oauth-authz]).

In particular, if symmetric keys are used, the AS generates a proof-of-possession key, binds it to the Access Token, and provides it to the joining node in the 'cnf' parameter of the Access Token response. Instead, if asymmetric keys are used, the joining node provides its own public key to the AS in the 'req_cnf' parameter of the Access Token request. Then, the AS uses it as proof-of-possession key bound to the Access Token, and provides the joining node with the Group Manager's public key in the 'rs_cnf' parameter of the Access Token response.

4. Joining Node to Group Manager

The following subsections describe the interactions between the joining node and the Group Manager, i.e. the Access Token post and the Request-Response exchange to join the OSCORE group.

4.1. Token Post

The joining node posts the Access Token to the /authz-info endpoint at the Group Manager, according to the Token post defined in Section 3.3 of [I-D.ietf-ace-key-groupcomm].

At this point in time, the joining node might not have all the necessary information concerning the public keys in the OSCORE group, as well as concerning the algorithm and related parameters for computing countersignatures in the OSCORE group. In such a case, the joining node MAY use the 'sign_info' and 'pub_key_enc' parameters defined in Section 3.3 of [I-D.ietf-ace-key-groupcomm] to ask for such information.

Alternatively, the joining node may retrieve this information by other means, e.g. by using the approach described in [I-D.tiloca-core-oscore-discovery].

If the Access Token is valid, the Group Manager responds to the POST request with a 2.01 (Created) response, according to what is specified in the signalled transport profile of ACE. The Group Manager MUST use the Content-Format "application/ace+cbor" defined in Section 8.14 of [I-D.ietf-ace-oauth-authz].

The payload of the 2.01 (Created) response is a CBOR map, which MUST include the 'cnonce' parameter defined in section 5.1.2 of [I-D.ietf-ace-oauth-authz], and MAY include the 'sign_info' parameter as well as the 'pub_key_enc' parameter.

The 'cnonce' parameter includes a nonce N generated by the Group Manager. The joining node may use this nonce in order to prove the possession of its own private key, upon joining the group (see Section 4.2).

If present in the response:

- 'sign_alg', i.e. the first element of the 'sign_info' parameter, takes value from Tables 5 and 6 of [RFC8152].

- 'sign_parameters', i.e. the second element of the 'sign_info' parameter, takes values from the "Counter Signature Parameters" Registry (see Section 9.1 of [I-D.ietf-core-oscore-groupcomm]).

Its structure depends on the value of 'sign_alg'. If no parameters of the counter signature algorithm are specified, 'sign_parameters' MUST be encoding the CBOR simple value Null.

- 'sign_key_parameters', i.e. the third element of the 'sign_info' parameter, takes values from the "Counter Signature Key Parameters" Registry (see Section 9.2 of [I-D.ietf-core-oscore-groupcomm]). Its structure depends on the value of 'sign_alg'. If no parameters of the key used with the counter signature algorithm are specified, 'sign_key_parameters' MUST be encoding the CBOR simple value Null.

- 'pub_key_enc' takes value from Figure 2, as a public key encoding in the "ACE Public Key Encoding" Registry (see Section 11.2 of [I-D.ietf-ace-key-groupcomm]).

Name	Value	Description	Reference
COSE_Key	1	Public key encoded as COSE Key	{{RFC8152}}

Figure 2: ACE Public Key Encoding Values

Note that the CBOR map specified as payload of the 2.01 (Created) response may include further parameters, e.g. according to the signalled transport profile of ACE.

Finally, the joining node establishes a secure channel with the Group Manager, according to what is specified in the Access Token response and the signalled transport profile of ACE.

4.2. Join Request

Once a secure communication channel with the Group Manager has been established, the joining node requests to join the OSCORE group, by accessing the related join resource at the Group Manager.

In particular, the joining node sends to the Group Manager a confirmable CoAP request, using the method POST and targeting the join endpoint associated to that group. This Join Request follows the format and processing of the Key Distribution Request message defined in Section 4.1 of [I-D.ietf-ace-key-groupcomm]. In particular:

- The 'type' parameter is set to 1 ("key distribution").

- The 'get_pub_keys' parameter is present only if the joining node wants to retrieve the public keys of the group members from the Group Manager during the join process (see Section 6). Otherwise, this parameter MUST NOT be present.

- The 'client_cred' parameter, if present, includes the public key of the joining node. In case the joining node knows the encoding of public keys in the OSCORE group, as well as the countersignature algorithm and possible associated parameters used in the OSCORE group, the included public key MUST be in a consistent format. This parameter MAY be omitted if: i) the joining node is asking to access the group exclusively as monitor; or ii) the Group Manager already acquired this information, for instance during a past join process. In any other case, this parameter MUST be present.

Furthermore, the CBOR map specified as payload of the Join Request MAY also include the following additional parameter, which MUST be present if the 'client_cred' parameter is present.

- The 'client_cred_verify' parameter, which is encoded as a CBOR byte string and contains a signature computed by the joining node, in order to prove possession of its own private key. The signature is computed over the nonce N received in the 2.01 (Created) response to the Token POST (see Section 4.1). In particular, the joining node MUST use the COSE_CounterSignature0 object [RFC8152], with the Sig_structure containing the nonce N as payload; and an empty external_aad. The joining node computes the signature by using the same private key and countersignature algorithm it intends to use for signing messages in the OSCORE group.

4.3. Join Response

The Group Manager processes the Join Request according to [I-D.ietf-ace-oauth-authz] and Section 4.2 of [I-D.ietf-ace-key-groupcomm]. Also, the Group Manager MUST return a 4.00 (Bad Request) response in case the Join Request includes the 'client_cred' parameter but does not include the 'client_cred_verify' parameter.

If the request processing yields a positive outcome, the Group Manager performs the further following checks.

- In case the Join Request includes the 'client_cred' parameter, the Group Manager checks that the public key of the joining node has an accepted format. That is, the public key has to be encoded as expected in the OSCORE group, and has to be consistent with the

counter signature algorithm and possible associated parameters used in the OSCORE group. The join process fails if the public key of the joining node does not have an accepted format.

- In case the Join Request does not include the 'client_cred' parameter, the Group Manager checks whether it is storing a public key for the joining node, which is consistent with the encoding, counter signature algorithm and possible associated parameters used in the OSCORE group. The join process fails if the Group Manager either: i) does not store a public key with an accepted format for the joining node; or ii) stores multiple public keys with an accepted format for the joining node.

- In case the Join Request includes the 'client_cred_verify' parameter, the Group Manager verifies the signature contained in the parameter. To this end, it considers: i) as signed value, the nonce N previously provided in the 2.01 (Created) response to the Token POST (see Section 4.1); ii) the countersignature algorithm used in the OSCORE group; and iii) the public key of the joining node, either retrieved from the 'client_cred' parameter, or as stored from a past join process. The join process fails if the Group Manager does not successfully verify the signature.

If the join process has failed, the Group Manager MUST reply to the joining node with a 4.00 (Bad Request) response. The payload of this response is a CBOR map, which includes a 'sign_info' parameter and a 'pub_key_enc' parameter, formatted as in the Token POST response in Section 4.1.

Upon receiving this response, the joining node SHOULD send a new Join Request to the Group Manager, which contains:

- The 'client_cred' parameter, including a public key in a format consistent with the encoding, countersignature algorithm and possible associated parameters indicated by the Group Manager.

- The 'client_cred_verify' parameter, including a signature computed as described in Section 4.2, by using the public key indicated in the current 'client_cred' parameter, with the countersignature algorithm and possible associated parameters indicated by the Group Manager.

Otherwise, in case of success, the Group Manager updates the group membership by registering the joining node as a new member of the OSCORE group.

Then, the Group Manager replies to the joining node providing the updated security parameters and keying meterial necessary to

participate in the group communication. This Join Response follows the format and processing of the Key Distribution success Response message defined in Section 4.2 of [I-D.ietf-ace-key-groupcomm]. In particular:

- The 'kty' parameter identifies a key of type "Group_OSCORE_Security_Context object", defined in Section 9.1 of this specification.

- The 'key' parameter includes what the joining node needs in order to set up the OSCORE Security Context as per Section 2 of [I-D.ietf-core-oscore-groupcomm]. This parameter has as value a Group_OSCORE_Security_Context object, which is defined in this specification and extends the OSCORE_Security_Context object encoded in CBOR as defined in Section 3.2.1 of [I-D.ietf-ace-oscore-profile]. In particular, it contains the additional parameters 'cs_alg', 'cs_params', 'cs_key_params' and 'cs_key_enc' defined in Section 9.2 of this specification. More specifically, the 'key' parameter is composed as follows.

  * The 'ms' parameter MUST be present and includes the OSCORE Master Secret value.

  * The 'clientId' parameter, if present, has as value the OSCORE Sender ID assigned to the joining node by the Group Manager. This parameter is not present if the node joins the group exclusively as monitor, according to what specified in the Access Token (see Section 3.2). In any other case, this parameter MUST be present.

  * The 'hkdf' parameter, if present, has as value the KDF algorithm used in the group.

  * The 'alg' parameter, if present, has as value the AEAD algorithm used in the group.

  * The 'salt' parameter, if present, has as value the OSCORE Master Salt.

  * The 'contextId' parameter MUST be present and has as value the Group Identifier (Gid) associated to the OSCORE group.

  * The 'rpl' parameter, if present, specifies the OSCORE Replay Window Size and Type value.

  * The 'cs_alg' parameter MUST be present and specifies the algorithm used to countersign messages in the group. This parameter takes values from Tables 5 and 6 of [RFC8152].

  * The 'cs_params' parameter MAY be present and specifies the additional parameters for the counter signature algorithm. This parameter is a CBOR map whose content depends on the counter signature algorithm, as specified in Section 2 and Section 9.1 of [I-D.ietf-core-oscore-groupcomm].

  * The 'cs_key_params' parameter MAY be present and specifies the additional parameters for the key used with the counter signature algorithm. This parameter is a CBOR map whose content depends on the counter signature algorithm, as specified in Section 2 and Section 9.2 of [I-D.ietf-core-oscore-groupcomm].

  * The 'cs_key_enc' parameter MAY be present and specifies the encoding of the public keys of the group members. This parameter is a CBOR integer, whose value is taken from Figure 2, as a public key encoding in the "ACE Public Key Encoding" Registry (see Section 11.2 of [I-D.ietf-ace-key-groupcomm]). If this parameter is not present, COSE_Key (1) MUST be assumed as default value.

- The 'profile' parameter MUST be present and has value coap_group_oscore_app (TBD), which is defined in Section 9.3 of this specification.

- The 'exp' parameter MUST be present and specifies the expiration time in seconds after which the OSCORE Security Context derived from the 'key' parameter is not valid anymore.

- The 'pub_keys' parameter is present only if the 'get_pub_keys' parameter was present in the Join Request. If present, this parameter includes the public keys of the group members that are relevant to the joining node. That is, it includes: i) the public keys of the responders currently in the group, in case the joining node is configured (also) as requester; and ii) the public keys of the requesters currently in the group, in case the joining node is configured (also) as responder or monitor.

- The 'group_policies' parameter SHOULD be present and includes a list of parameters indicating particular policies enforced in the group. For instance, its field "Sequence Number Synchronization Method" can indicate the method to achieve synchronization of sequence numbers among group members (see Appendix E of [I-D.ietf-core-oscore-groupcomm]), as indicated by the corresponding value from the "Sequence Number Synchronization Method" Registry defined in Section 11.8 of [I-D.ietf-ace-key-groupcomm].

Finally, the joining node uses the information received in the Join Response to set up the OSCORE Security Context, as described in Section 2 of [I-D.ietf-core-oscore-groupcomm]. From then on, the joining node can exchange group messages secured with OSCORE as described in [I-D.ietf-core-oscore-groupcomm].

If the application requires backward security, the Group Manager SHALL generate updated security parameters and group keying material, and provide it to all the current group members (see Section 7).

When the OSCORE Security Context expires, as specified by the 'exp' parameter of the Join Response, the node considers it invalid and to be renewed. Then, the node retrieves updated security parameters and keying material, by exchanging with the Group Manager a shortened Join Request sent to the same Join Resource with the 'type' parameter set to 3 ("update key") and a shortened Join Response message, according to the approach defined in Section 6 of [I-D.ietf-ace-key-groupcomm]. Finally, the node uses the updated security parameters and keying material to set up the new OSCORE Security Context as described in Section 2 of [I-D.ietf-core-oscore-groupcomm].

Furthermore, as discussed in Section 2.2 of [I-D.ietf-core-oscore-groupcomm], the node may at some point experience a wrap-around of its own Sender Sequence Number in the group. When this happens, the node MUST send to the Group Manager a shortened Join Request message to the same Join Resource, with the 'type' parameter set to 4 ("new"). Upon receiving this request message, the Group Manager either rekeys the whole OSCORE group as discussed in Section 7, or generates a new Sender ID for that node and replies with a shortened Join Response message where:

- Only the parameters 'type', 'kty', 'key', 'profile' and 'exp' are present.

- The 'clientId' parameter of the 'key' parameter specifies the new Sender ID of the node.

5. Leaving of a Group Member

A node may be removed from the OSCORE group, due to expired or revoked authorization, or after its own request to the Group Manager.

If the application requires forward security, the Group Manager SHALL generate updated security parameters and group keying material, and provide it to the remaining group members (see Section 7). The leaving node must not be able to acquire the new security parameters and group keying material distributed after its leaving.

Same considerations in Section 5 of [I-D.ietf-ace-key-groupcomm] apply here as well, considering the Group Manager acting as KDC. In particular, a node requests to leave the OSCORE group as described in Section 5.2 of [I-D.ietf-ace-key-groupcomm], i.e. by sending to the Group Manager a request to the same Join Resource with the 'type' parameter set to 2 ("leave").

6. Public Keys of Joining Nodes

Source authentication of OSCORE messages exchanged within the group is ensured by means of digital counter signatures (see Sections 2 and 3 of [I-D.ietf-core-oscore-groupcomm]). Therefore, group members must be able to retrieve each other's public key from a trusted key repository, in order to verify source authenticity of incoming group messages.

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

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

- The joining node is going to join the group exclusively as monitor. That is, it is not going to send messages to the group, and hence to produce signatures with its own private key. In this case, the joining node is not required to provide its own public key to the Group Manager, which thus does not have to perform any check related to the public key encoding, or to a countersignature algorithm and possible associated parameters for that joining node.

- The Group Manager already acquired the public key of the joining node during a past join process. In this case, the joining node MAY not provide again its own public key to the Group Manager, in order to limit the size of the Join Request. The joining node MUST provide its own public key again if it has provided the Group Manager with multiple public keys during past join processes, intended for different OSCORE groups. If the joining node provides its own public key, the Group Manager performs consistency checks as in Section 4.3 and, in case of success, considers it as the public key associated to the joining node in the OSCORE group.

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

  1. The proof-of-possession key is consistent with the encoding as well as with the counter signature algorithm and possible associated parameters used in the OSCORE group. Then, the Group Manager considers the proof-of-possession key as the public key associated to the joining node in the OSCORE group. If the joining node is aware that the proof-of-possession key is also valid for the OSCORE group, it MAY not provide it again as its own public key to the Group Manager. The joining node MUST provide its own public key again if it has provided the Group Manager with multiple public keys during past join processes, intended for different OSCORE groups. If the joining node provides its own public key in the 'client_cred' parameter of the Join Request (see Section 4.2), the Group Manager performs consistency checks as in Section 4.3 and, in case of success, considers it as the public key associated to the joining node in the OSCORE group.

  2. The proof-of-possession key is not consistent with the encoding or with the counter signature algorithm and possible associated parameters used in the OSCORE group. In this case, the joining node MUST provide a different consistent public key to the Group Manager in the 'client_cred' parameter of the Join Request (see Section 4.2). Then, the Group Manager performs consistency checks on this latest provided public key as in Section 4.3 and, in case of success, considers it as the public key associated to the joining node in the OSCORE group.

- The joining node and the Group Manager use a symmetric proof-of-possession key to establish a secure communication channel. In this case, upon performing a join process with that Group Manager for the first time, the joining node specifies its own public key in the 'client_cred' parameter of the Join Request targeting the join endpoint (see Section 4.2).

Furthermore, as described in Section 4.2, the joining node may have explicitly requested the Group Manager to retrieve the public keys of the current group members, i.e. by including the 'get_pub_keys' parameter in the Join Request. In this case, the Group Manager includes also such public keys in the 'pub_keys' parameter of the Join Response (see Section 4.3).

Later on as a group member, the node may need to retrieve the public keys of other group members. The node can do that by exchanging with the Group Manager a shortened Join Request sent to the same Join

Resource with the 'type' parameter set to 5 ("pub keys") and a shortened Join Response, according to the approach defined in Section 7 of [I-D.ietf-ace-key-groupcomm].

7. Group Rekeying Process

In order to rekey the OSCORE group, the Group Manager distributes a new Group ID of the group and a new OSCORE Master Secret for that group. When doing so, the Group Manager MAY take a best effort to preserve the same unchanged Sender IDs for all group members. This avoids affecting the retrieval of public keys from the Group Manager as well as the verification of message countersignatures.

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

The Group Manager uses the same format of the Join Response message in Section 4.3. In particular:

- Only the parameters 'type', 'kty', 'key', 'profile' and 'exp' are present.
- The 'ms' parameter of the 'key' parameter specifies the new OSCORE Master Secret value.
- The 'contextId' parameter of the 'key' parameter specifies the new Group ID.

The Group Manager separately sends a group rekeying message to each group member to be rekeyed. Each rekeying message MUST be secured with the pairwise secure communication channel between the Group Manager and the group member used during the join process.

This approach requires group members to act (also) as servers, in order to correctly handle unsolicited group rekeying messages from the Group Manager. In particular, if a group member and the Group Manager use OSCORE [I-D.ietf-core-object-security] to secure their pairwise communications, the group member MUST create a Replay Window in its own Recipient Context upon establishing the OSCORE Security Context with the Group Manager, e.g. by means of the OSCORE profile of ACE [I-D.ietf-ace-oscore-profile].

Group members and the Group Manager SHOULD additionally support alternative rekeying approaches that do not require group members to act (also) as servers. A number of such approaches are defined in Section 6 of [I-D.ietf-ace-key-groupcomm], and are based on the following rationale:

- A group member queries the Group Manager for updated group keying material, by sending a dedicated request to the same Join Resource targeted when joining the group. Like for the case discussed in Section 4.3 where the OSCORE Security Context expires, the group member exchanges with the Group Manager a shortened Join Request sent to the same Join Resource with the 'type' parameter set to 3 ("update key") and a shortened Join Response message, according to the approach defined in Section 6 of [I-D.ietf-ace-key-groupcomm].

- A group member subscribes for updates to the join resource and its associated group keying material on the Group Manager. This can rely on CoAP Observe [RFC7641] or on a full-fledged Pub-Sub model [I-D.ietf-core-coap-pubsub] with the Group Manager acting as Broker.

Either case, the Group Manager provides the (updated) group keying material as specified above in this section.

8. Security Considerations

The method described in this document leverages the following management aspects related to OSCORE groups and discussed in the sections of [I-D.ietf-core-oscore-groupcomm] referred below.

- Management of group keying material (see Section 2.1 of [I-D.ietf-core-oscore-groupcomm]). The Group Manager is responsible for the renewal and re-distribution of the keying material in the groups of its competence (rekeying). According to the specific application requirements, this can include rekeying the group upon changes in its membership. In particular, renewing the keying material is required upon a new node's joining or a current node's leaving, in case backward security and forward security have to be preserved, respectively.

- Provisioning and retrieval of public keys (see Section 2 of [I-D.ietf-core-oscore-groupcomm]). The Group Manager acts as key repository of public keys of group members, and provides them upon request.

- Synchronization of sequence numbers (see Section 5 of [I-D.ietf-core-oscore-groupcomm]). This concerns how a responder node that has just joined an OSCORE group can synchronize with the sequence number of requesters in the same group.

Before sending the Join Response, the Group Manager MUST verify that the joining node actually owns the associated private key. To this end, the Group Manager can rely on the proof-of-possession challenge-response defined in Section 4. Alternatively, the joining node can

use its own public key as asymmetric proof-of-possession key to establish a secure channel with the Group Manager, e.g. as in Section 3.2 of [I-D.ietf-ace-dtls-authorize]. However, this requires such proof-of-possession key to be consistent with the encoding as well as with the countersignature algorithm and possible associated parameters used in the OSCORE group.

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

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

9. IANA Considerations

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

This document has the following actions for IANA.

9.1. ACE Groupcomm Key Registry

IANA is asked to register the following entry in the "ACE Groupcomm Key" Registry defined in Section 11.5 of [I-D.ietf-ace-key-groupcomm].

- Name: Group_OSCORE_Security_Context object
- Key Type Value: TBD
- Profile: "coap_group_oscore_app", defined in Section 9.3 of this specification.
- Description: A Group_OSCORE_Security_Context object encoded as described in Section 4.3 of this specification.
- Reference: [[This specification]]

9.2. OSCORE Security Context Parameters Registry

IANA is asked to register the following entries in the "OSCORE Security Context Parameters" Registry defined in Section 9.2 of [I-D.ietf-ace-oscore-profile].

- Name: cs_alg
- CBOR Label: TBD
- CBOR Type: tstr / int
- Registry: COSE Algorithm Values (ECDSA, EdDSA)
- Description: OSCORE Counter Signature Algorithm Value
- Reference: [[This specification]]
- Name: cs_params
- CBOR Label: TBD
- CBOR Type: map
- Registry: Counter Signatures Parameters
- Description: OSCORE Counter Signature Algorithm Additional Parameters
- Reference: [[This specification]]
- Name: cs_key_params
- CBOR Label: TBD
- CBOR Type: map
- Registry: Counter Signatures Key Parameters
- Description: OSCORE Counter Signature Key Additional Parameters
- Reference: [[This specification]]
- Name: cs_key_enc
- CBOR Label: TBD
- CBOR Type: integer

- Registry: ACE Public Key Encoding
- Description: Encoding of Public Keys to be used with the OSCORE Counter Signature Algorithm
- Reference: [[This specification]]

9.3. ACE Groupcomm Profile Registry

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

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

9.4. Sequence Number Synchronization Method Registry

IANA is asked to register the following entries in the "Sequence Number Synchronization Method" Registry defined in Section 11.8 of [I-D.ietf-ace-key-groupcomm].

- Name: Best effort
- Value: 1
- Description: No action is taken.
- Reference: [I-D.ietf-core-oscore-groupcomm] (Appendix E.1).
- Name: Baseline
- Value: 2
- Description: The first received request sets the baseline reference point, and is discarded with no delivery to the application.
- Reference: [I-D.ietf-core-oscore-groupcomm] (Appendix E.2).
- Name: Echo challenge-response

- Value: 3

- Description: Challenge response using the Echo Option for CoAP from [I-D.ietf-core-echo-request-tag].

- Reference: [I-D.ietf-core-oscore-groupcomm] (Appendix E.3).

9.5. ACE Public Key Encoding Registry

This specification registers the value defined in Figure 2 in the "ACE Public Key Encoding" IANA Registry.

10. References

10.1. Normative References

[I-D.ietf-ace-key-groupcomm]
Palombini, F. and M. Tiloca, "Key Provisioning for Group Communication using ACE", draft-ietf-ace-key-groupcomm-02 (work in progress), July 2019.

[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-24 (work in progress), March 2019.

[I-D.ietf-ace-oscore-profile]
Palombini, F., Seitz, L., Selander, G., and M. Gunnarsson, "OSCORE profile of the Authentication and Authorization for Constrained Environments Framework", draft-ietf-ace-oscore-profile-07 (work in progress), February 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.

[I-D.ietf-core-oscore-groupcomm]
Tiloca, M., Selander, G., Palombini, F., and J. Park, "Group OSCORE - Secure Group Communication for CoAP", draft-ietf-core-oscore-groupcomm-05 (work in progress), July 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>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <https://www.rfc-editor.org/info/rfc7252>.

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

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

10.2. Informative References

[I-D.dijk-core-groupcomm-bis]
Dijk, E., Wang, C., and M. Tiloca, "Group Communication for the Constrained Application Protocol (CoAP)", draft-dijk-core-groupcomm-bis-00 (work in progress), March 2019.

[I-D.ietf-ace-dtls-authorize]
Gerdes, S., Bergmann, O., Bormann, C., Selander, G., and L. Seitz, "Datagram Transport Layer Security (DTLS) Profile for Authentication and Authorization for Constrained Environments (ACE)", draft-ietf-ace-dtls-authorize-08 (work in progress), April 2019.

[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-08 (work in progress), March 2019.

[I-D.ietf-core-echo-request-tag]
Amsuess, C., Mattsson, J., and G. Selander, "CoAP: Echo, Request-Tag, and Token Processing", draft-ietf-core-echo-request-tag-05 (work in progress), May 2019.

[I-D.tiloca-core-oscore-discovery]
Tiloca, M., Amsuess, C., and P. Stok, "Discovery of OSCORE Groups with the CoRE Resource Directory", draft-tiloca-core-oscore-discovery-02 (work in progress), March 2019.

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

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

[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>.

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

Appendix A. Profile Requirements

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

- Communication protocol that the members of the group must use: CoAP, possibly over IP multicast.
- Security protocols that the group members must use to protect their communication: Group OSCORE.
- Specify the encoding and value of the identifier of group and role of 'scope': see Section 3.1.
- Profile identifier: coap_group_oscore_app
- Acceptable values of 'kty': Group_OSCORE_Security_Context object
- Specify the format and content of 'group_policies' entries: three values are defined and registered, as content of the entry "Sequence Number Synchronization Method" (see Section 9.4).
- (Optional) specify the format and content of 'mgt_key_material': no.
- (Optional) specify the transport profile of ACE [I-D.ietf-ace-oauth-authz] to use between Client and Group Manager: any transport profile of ACE that complies with the requirements in Appendix C of [I-D.ietf-ace-oauth-authz].

- (Optional) specify the encoding of public keys, of 'client_cred', and of 'pub_keys' if COSE_Keys are not used: no.

- (Optional) specify the acceptable values for parameters related to signature algorithm and signature keys: 'sign_alg' takes value from Tables 5 and 6 of [RFC8152]; 'sign_parameters' takes values from the "Counter Signature Parameters" Registry (see Section 9.1 of [I-D.ietf-core-oscore-groupcomm]); 'sign_key_parameters' takes values from the "Counter Signature Key Parameters" Registry (see Section 9.2 of [I-D.ietf-core-oscore-groupcomm]); 'pub_key_enc' takes value from Figure 2 in Section 4.1.

- (Optional) specify the negotiation of parameter values for signature algorithm and signature keys, if 'sign_info' and 'pub_key_enc' are not used: pre-knowledge by using the approach based on the CoRE Resource Directory described in [I-D.tiloca-core-oscore-discovery].

Appendix B. Document Updates

RFC EDITOR: PLEASE REMOVE THIS SECTION.

B.1. Version -01 to -02

- Editorial fixes.

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

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

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

- Added parameters to indicate the encoding of public keys.

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

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

- Code 4.00 (Bad request), in responses to joining nodes providing an invalid public key (Section 4.3).

- Clarifications on provisioning and checking of public keys (Sections 4 and 6).

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

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

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

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

B.2. Version -00 to -01

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

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

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

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

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

Acknowledgments

The authors sincerely thank Santiago Aragon, Stefan Beck, Martin Gunnarsson, Rikard Hoeglund, Jim Schaad, Ludwig Seitz, Goeran Selander and Peter van der Stok for their comments and feedback.

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

Authors' Addresses

Marco Tiloca
RISE AB
Isafjordsgatan 22
Kista SE-164 29 Stockholm
Sweden

Email: marco.tiloca@ri.se

Jiye Park
Universitaet Duisburg-Essen
Schuetzenbahn 70
Essen 45127
Germany

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

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

Email: francesca.palombini@ericsson.com