                                                      P. van der Stok
                                                           Consultant
                                                        July 13, 2020

            Discovery of OSCORE Groups with the CoRE Resource Directory
                    draft-tiloca-core-oscore-discovery-06

Abstract

   Group communication over the Constrained Application Protocol (CoAP)
   can be secured by means of Group Object Security for Constrained
   RESTful Environments (Group OSCORE). At deployment time, devices may
   not know the exact OSCORE groups to join, the respective Group
   Manager, or other information required to perform the joining
   process. This document describes how a CoAP endpoint can use
   descriptions and links of resources registered at the CoRE Resource
   Directory to discover OSCORE groups and to acquire information for
   joining them through the respective Group Manager. A given OSCORE
   group may protect multiple application groups, which are separately
   announced in the Resource Directory as sets of endpoints sharing a
   pool of resources. This approach is consistent with, but not limited
   to, the joining of OSCORE groups based on the ACE framework for
   Authentication and Authorization in constrained environments.

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 14, 2021.

Table of Contents

[1](#). Introduction

The Constrained Application Protocol (CoAP) [RFC7252] supports group
communication over IP multicast [I-D.ietf-core-groupcomm-bis] to
improve efficiency and latency of communication and reduce bandwidth
requirements. A set of CoAP endpoints constitutes an application
group by sharing a common pool of resources, that can be efficiently
accessed through group communication. The members of an application
group may be members of a security group, thus sharing a common set
of keying material to secure group communication.

The security protocol Group Object Security for Constrained RESTful
Environments (Group OSCORE) [I-D.ietf-core-oscore-groupcomm] builds
on OSCORE [RFC8613] and protects CoAP messages end-to-end in group
communication contexts through CBOR Object Signing and Encryption
(COSE)
[I-D.ietf-cose-rfc8152bis-struct][I-D.ietf-cose-rfc8152bis-algs]. An
application group may rely on one or more OSCORE groups as security
groups, and a same OSCORE group may be used by multiple application
groups at the same time.

A CoAP endpoint relies on a Group Manager (GM) to join an OSCORE
group and get the group keying material. The joining process in
[I-D.ietf-ace-key-groupcomm-oscore] is based on the ACE framework for
Authentication and Authorization in constrained environments
[I-D.ietf-ace-oauth-authz], with the joining endpoint and the GM
acting as ACE Client and Resource Server, respectively. That is, the
joining endpoint accesses the group-membership resource exported by
the GM and associated with the OSCORE group to join.

Typically, devices store a static X509 IDevID certificate installed
at manufacturing time [I-D.ietf-anima-bootstrapping-keyinfra]. This
is used at deployment time during an enrollment process that provides
the devices with an Operational Certificate, possibly updated during
the device lifetime. Operational Certificates may specify
information to join OSCORE groups, especially a reference to the
group-membership resources to access at the respective GMs.

However, it is usually impossible to provide such precise information
to freshly deployed devices, as part of their (early) Operational
Certificate. This can be due to a number of reasons: (1) the OSCORE
group(s) to join and the responsible GM(s) are generally unknown at
manufacturing time; (2) an OSCORE group of interest is created, or
the responsible GM is deployed, only after the device is enrolled and
fully operative in the network; (3) information related to existing
OSCORE groups or to their GMs has changed. This requires a method
for CoAP endpoints to dynamically discover OSCORE groups and their
GM, and to retrieve relevant information about deployed groups.

To this end, CoAP endpoints can use descriptions and links of group-membership resources at GMs, to discover OSCORE groups and retrieve the information required for joining them. With the discovery process of OSCORE groups expressed in terms of links to resources, the remaining problem is the discovery of those links. The CoRE Resource Directory (RD) [I-D.ietf-core-resource-directory] allows such discovery in an efficient way, and it is expected to be used in many setups that would benefit of OSCORE group discovery.

This specification builds on this approach and describes how CoAP endpoints can use the RD to perform the link discovery steps, in order to discover OSCORE groups and retrieve the information required to join them through their GM. In short, the GM registers as an endpoint with the RD. The resulting registration resource includes one link per OSCORE group under that GM, specifying the path to the related group-membership resource to access for joining that group.

Additional descriptive information about the OSCORE group is stored with the registered link. In the RD based on Link Format [RFC6690] and defined in [I-D.ietf-core-resource-directory], this information is specified as target attributes of the registered link, and includes the identifiers of the application groups which use that OSCORE group. This enables a lookup of those application groups at the RD, where they are separately announced by a Commissioning Tool (see Appendix A of [I-D.ietf-core-resource-directory]).

When querying the RD for OSCORE groups, a CoAP endpoint can use CoAP observation [RFC7641]. This results in automatic notifications on the creation of new OSCORE groups or the update of existing groups. Thus, it facilitates the early deployment of CoAP endpoints, i.e. even before the GM is deployed and OSCORE groups are created.

Interaction examples are provided in Link Format, as well as in the Constrained RESTful Application Language CoRAL [I-D.ietf-core-coral] with reference to a CoRAL-based RD [I-D.hartke-t2trg-coral-reef]. While all the CoRAL examples use the CoRAL textual serialization format, the CBOR [RFC7049] or JSON [RFC8259] binary serialization format is used when sending such messages on the wire.

The approach in this document is consistent with, but not limited to, the joining of OSCORE groups in [I-D.ietf-ace-key-groupcomm-oscore].

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.

This specification requires readers to be familiar with the terms and
concepts discussed in [I-D.ietf-core-resource-directory] and
[RFC6690], as well as in [I-D.ietf-core-coral]. Readers should also
be familiar with the terms and concepts discussed in
[RFC7252][I-D.ietf-core-groupcomm-bis],
[I-D.ietf-core-oscore-groupcomm] and
[I-D.ietf-ace-key-groupcomm-oscore].

Terminology for constrained environments, such as "constrained
device" and "constrained-node network", is defined in [RFC7228].

Consistently with the definitions from Section 2.1 of
[I-D.ietf-core-groupcomm-bis], this document also refers to the
following terminology.

o CoAP group: a set of CoAP endpoints all configured to receive CoAP
   multicast messages sent to the group's associated IP multicast
   address and UDP port. An endpoint may be a member of multiple
   CoAP groups by subscribing to multiple IP multicast addresses.

o Security group: a set of CoAP endpoints that share the same
   security material, and use it to protect and verify exchanged
   messages. A CoAP endpoint may be a member of multiple security
   groups. There can be a one-to-one or a one-to-many relation
   between security groups and CoAP groups.

   This document especially considers a security group to be an
   OSCORE group, where all members share one OSCORE Security Context
   to protect group communication with Group OSCORE
   [I-D.ietf-core-oscore-groupcomm]. However, the approach defined
   in this document can be used to support the discovery of different
   security groups than OSCORE groups.

o Application group: a set of CoAP endpoints that share a common set
   of resources. An endpoint may be a member of multiple application
   groups. An application group can be associated with one or more
   security groups, and multiple application groups can use the same
   security group. Application groups are announced in the RD by a
   Commissioning Tool, according to the RD-Groups usage pattern (see
   Appendix A of [I-D.ietf-core-resource-directory]).

2. Registration of Group Manager Endpoints

During deployment, a Group Manager (GM) can find the CoRE Resource Directory (RD) as described in Section 4 of [I-D.ietf-core-resource-directory].

Afterwards, the GM registers as an endpoint with the RD, as described in Section 5 of [I-D.ietf-core-resource-directory]. The GM SHOULD NOT use the Simple Registration approach described in Section 5.1 of [I-D.ietf-core-resource-directory].

When registering with the RD, the GM also registers the links to all the group-membership resources it has at that point in time, i.e. one for each of its security groups.

In the registration request, each link to a group-membership resource has as target the URI of that resource at the GM. Also, it specifies a number of descriptive parameters as defined in Section 2.1.

2.1. Parameters

For each registered link to a group-membership resource at a GM, the following parameters are specified together with the link.

In the RD defined in [I-D.ietf-core-resource-directory] and based on Link Format, each parameter is specified in a target attribute with the same name.

In an RD based on CoRAL, such as the one defined in [I-D.hartke-t2trg-coral-reef], each parameter is specified in a nested element with the same name.

o 'rt', with value "core.osc.mbr" (see Section 7.1).

o 'sec-gp', specifying the name of the security group of interest, as a stable and invariant identifier, such as the group name used in [I-D.ietf-ace-key-groupcomm-oscore]. This parameter MUST specify a single value.

o 'app-gp', specifying the name(s) of the application group(s) associated to the security group of interest indicated by 'sec-gp'. This parameter MUST occur once for each application group, and MUST specify only a single application group.

Optionally, the following parameters can also be specified.

o 'cs_alg', specifying the algorithm used to countersign messages in the security group. If present, this parameter MUST specify a

single value encoded as a text string, which is taken from the
'Value' column of the "COSE Algorithms" Registry
[COSE.Algorithms].

o 'cs_alg_crv', specifying the elliptic curve (if applicable) for
   the algorithm used to countersign messages in the security group.
   If present, this parameter MUST specify a single value encoded as
   a text string, which is taken from the 'Value' column of the "COSE
   Elliptic Curves" Registry [COSE.Elliptic.Curves].

o 'cs_key_kty', specifying the key type of countersignature keys
   used to countersign messages in the security group. If present,
   this parameter MUST specify a single value encoded as a text
   string, which is taken from the 'Value' column of the "COSE Key
   Types" Registry [COSE.Key.Types].

o 'cs_key_crv', specifying the elliptic curve (if applicable) of
   countersignature keys used to countersign messages in the security
   group. If present, this parameter MUST specify a single value
   encoded as a text string, which is taken from the 'Value' column
   of the "COSE Elliptic Curves" Registry defined in
   [COSE.Elliptic.Curves].

o 'cs_kenc', specifying the encoding of the public keys used in the
   security group. If present, this parameter MUST specify a single
   value encoded as a text string. This specification explicitly
   admits the signaling of COSE Keys
   [I-D.ietf-cose-rfc8152bis-struct] as encoding for public keys,
   which is indicated with "1", as taken from the 'Confirmation Key'
   column of the "CWT Confirmation Method" Registry defined in
   [RFC8747]. Future specifications may define additional values for
   this parameter.

o 'alg', specifying the AEAD algorithm used in the security group.
   If present, this parameter MUST specify a single value encoded as
   a text string, which is taken from the 'Value' column of the "COSE
   Algorithms" Registry [COSE.Algorithms].

o 'hkdf', specifying the HKDF algorithm used in the security group.
   If present, this parameter MUST specify a single value encoded as
   a text string, which is taken from the 'Value' column of the "COSE
   Algorithms" Registry defined in [COSE.Algorithms].

Note that the values registered in the COSE Registries
[COSE.Algorithms][COSE.Elliptic.Curves][COSE.Key.Types] are strongly
typed. On the contrary, Link Format is weakly typed and thus does
not distinguish between, for instance, the string value "-10" and the
integer value -10.

Therefore, in the RD defined in [I-D.ietf-core-resource-directory]
and based on Link Format, possible values registered as a string that
looks like an integer, e.g. the string "-10" in the 'Value' column of
the "COSE Algorithms" Registry [COSE.Algorithms], are not supported
by this approach. Therefore, they MUST NOT be advertised through the
corresponding parameters above.

A CoAP endpoint that queries the RD to discover security groups and
their group-membership resource to access (see Section 4) would
benefit from the information above as follows.

o The values of 'cs_alg', 'cs_alg_crv', 'cs_key_kty', 'cs_key_crv'
   and 'cs_kenc' related to a group-membership resource provide an
   early knowledge of the format and encoding of public keys used in
   the security group. Thus, the CoAP endpoint does not need to ask
   the GM for this information as a preliminary step before the
   joining process, or to perform a trial-and-error joining exchange
   with the GM. Hence, the CoAP endpoint is able to provide the GM
   with its own public key in the correct expected format and
   encoding at the very first step of the joining process.

o The values of 'cs_alg', 'alg' and 'hkdf' related to a group-
   membership resource provide an early knowledge of the algorithms
   used in the security group. Thus, the CoAP endpoint is able to
   decide whether to actually proceed with the joining process,
   depending on its support for the indicated algorithms.

2.2. Relation Link to Authorization Server

For each registered link to a group-membership resource, the GM MAY
additionally specify the link to the ACE Authorization Server (AS)
[I-D.ietf-ace-oauth-authz] associated to the GM, and issuing
authorization credentials to join the security group as described in
[I-D.ietf-ace-key-groupcomm-oscore].

The link to the AS has as target the URI of the resource where to
send an authorization request to.

In the RD defined in [I-D.ietf-core-resource-directory] and based on
Link Format, the link to the AS is separately registered with the RD,
and includes the following parameters as target attributes.

o 'rel', with value "authorization_server".

o 'anchor', with value the target of the link to the group-
   membership resource at the GM.

In an RD based on CoRAL, such as the one defined in
[I-D.hartke-t2trg-coral-reef], this is mapped (as describe there) to
a link from the registration resource to the AS, using the
<http://www.iana.org/assignments/relation/authorization_server> link
relation type.

2.3. Registration Example

The example below shows a GM with endpoint name "gm1" and address
2001:db8::ab that registers with the RD.

The GM specifies the value of the 'sec-gp' parameter for accessing
the security group with name "feedca570000", and used by the
application group with name "group1" specified with the value of the
'app-gp' parameter. The countersignature algorithm used in the
security group is EdDSA, with elliptic curve Ed25519 and keys of type
OKP. Public keys used in the security group are encoded as COSE Keys
[I-D.ietf-cose-rfc8152bis-struct].

In addition, the GM specifies the link to the ACE Authorization
Server associated to the GM, to which a CoAP endpoint should send an
Authorization Request for joining the corresponding security group
[I-D.ietf-ace-key-groupcomm-oscore].

2.3.1. Example in Link Format

```
Request: GM -> RD

Req: POST coap://rd.example.com/rd?ep=gm1
Content-Format: 40
Payload:
</group-oscore/feedca570000>;ct=41;rt="core.osc.mbr";
                            sec-gp="feedca570000";app-gp="group1";
                            cs_alg="-8";cs_alg_crv="6";
                            cs_key_kty="1";cs_key_crv=6";
                            cs_kenc="1",
<coap://as.example.com/token>;
      rel="authorization-server";
      anchor="coap://[2001:db8::ab]/group-oscore/feedca570000"

Response: RD -> GM

Res: 2.01 Created
Location-Path: /rd/4521
```

2.3.2. Example in CoRAL

```
Request: GM -> RD

Req: POST coap://rd.example.com/rd?ep=gm1
Content-Format: TBD123456 (application/coral+cbor)

Payload:
#using <http://coreapps.org/core.oscore-discovery#>
#using reef = <http://coreapps.org/reef#>
#using iana = <http://www.iana.org/assignments/relation/>

#base <coap://[2001:db8::ab]/>
reef:rd-item </group-oscore/feedca570000> {
   reef:rt "core.osc.mbr"
   sec-gp "feedca570000"
   app-gp "group1"
   cs_alg -8
   cs_alg_crv 6
   cs_key_kty 1
   cs_key_crv 6
   cs_kenc 1
   iana:authorization-server <coap://as.example.com/token>
}

Response: RD -> GM

Res: 2.01 Created
Location-Path: /rd/4521
```

3. Addition and Update of Security Groups

The GM is responsible to refresh the registration of all its group-
membership resources in the RD. This means that the GM has to update
the registration within its lifetime as per Section 5.3.1 of
[I-D.ietf-core-resource-directory], and has to change the content of
the registration when a group-membership resource is added/removed,
or if its parameters have to be changed, such as in the following
cases.

o The GM creates a new security group and starts exporting the
   related group-membership resource.

o The GM dismisses an security group and stops exporting the related
   group-membership resource.

o Information related to an existing security group changes, e.g.
   the list of associated application groups.

To perform an update of its registrations, the GM can re-register
with the RD and fully specify all links to its group-membership
resources.

Alternatively, the GM can perform a PATCH/iPATCH [RFC8132] request to
the RD, as per Section 5.3.3 of [I-D.ietf-core-resource-directory].
This requires new media-types to be defined in future standards, to
apply a new document as a patch to an existing stored document.

3.1. Addition Example

The example below shows how the GM from Section 2 re-registers with
the RD. When doing so, it specifies:

o The same previous group-membership resource associated to the
   security group with name "feedca570000".

o An additional group-membership resource associated to the security
   group with name "ech0ech00000" and used by the application group
   "group2".

o A third group-membership resource associated with the security
   group with name "abcdef120000" and used by two application groups,
   namely "group3" and "group4".

Furthermore, the GM relates the same Authorization Server also to the
security groups "ech0ech00000" and "abcdef120000".

3.1.1. Example in Link Format

Request: GM -> RD

```
Req: POST coap://rd.example.com/rd?ep=gm1
Content-Format: 40
Payload:
</group-oscore/feedca570000>;ct=41;rt="core.osc.mbr";
                             sec-gp="feedca570000";app-gp="group1";
                             cs_alg="-8";cs_alg_crv="6";
                             cs_key_kty="1";cs_key_crv=6";
                             cs_kenc="1",
</group-oscore/ech0ech00000>;ct=41;rt="core.osc.mbr";
                             sec-gp="ech0ech00000";app-gp="group2";
                             cs_alg="-8";cs_alg_crv="6";
                             cs_key_kty="1";cs_key_crv=6";
                             cs_kenc="1",
</group-oscore/abcdef120000>;ct=41;rt="core.osc.mbr";
                             sec-gp="abcdef120000";app-gp="group3";
                             app-gp="group4";cs_alg="-8";
                             cs_alg_crv="6";cs_key_kty="1";
                             cs_key_crv=6";cs_kenc="1",
<coap://as.example.com/token>;
     rel="authorization-server";
     anchor="coap://[2001:db8::ab]/group-oscore/feedca570000",
<coap://as.example.com/token>;
     rel="authorization-server";
     anchor="coap://[2001:db8::ab]/group-oscore/ech0ech00000",
<coap://as.example.com/token>;
     rel="authorization-server";
     anchor="coap://[2001:db8::ab]/group-oscore/abcdef120000"

Response: RD -> GM

Res: 2.04 Changed
Location-Path: /rd/4521
```

3.1.2. Example in CoRAL

```
Request: GM -> RD
```

```
Req: POST coap://rd.example.com/rd?ep=gm1
Content-Format: TBD123456 (application/coral+cbor)

Payload:
#using <http://coreapps.org/core.oscore-discovery#>
#using reef = <http://coreapps.org/reef#>
#using iana = <http://www.iana.org/assignments/relation/>

#base <coap://[2001:db8::ab]/>
reef:rd-item </group-oscore/feedca570000> {
   reef:rt "core.osc.mbr"
   sec-gp "feedca570000"
   app-gp "group1"
   cs_alg -8
   cs_alg_crv 6
   cs_key_kty 1
   cs_key_crv 6
   cs_kenc 1
   iana:authorization-server <coap://as.example.com/token>
}
reef:rd-item </group-oscore/ech0ech00000> {
   reef:rt "core.osc.mbr"
   sec-gp "ech0ech00000"
   app-gp "group2"
   cs_alg -8
   cs_alg_crv 6
   cs_key_kty 1
   cs_key_crv 6
   cs_kenc 1
   iana:authorization-server <coap://as.example.com/token>
}
reef:rd-item </group-oscore/abcdef120000> {
   reef:rt "core.osc.mbr"
   sec-gp "abcdef120000"
   app-gp "group3"
   app-gp "group4"
   cs_alg -8
   cs_alg_crv 6
   cs_key_kty 1
   cs_key_crv 6
   cs_kenc 1
   iana:authorization-server <coap://as.example.com/token>
}

Response: RD -> GM

Res: 2.04 Changed
Location-Path: /rd/4521
```

4. **Discovery of Security Groups**

   A CoAP endpoint that wants to join a security group, hereafter called
   the joining node, might not have all the necessary information at
   deployment time. Also, it might want to know about possible new
   security groups created afterwards by the respective Group Managers.

   To this end, the joining node can perform a resource lookup at the RD
   as per Section 6.1 of [I-D.ietf-core-resource-directory], to retrieve
   the missing pieces of information needed to join the security
   group(s) of interest. The joining node can find the RD as described
   in Section 4 of [I-D.ietf-core-resource-directory].

   The joining node uses the following parameter value for the lookup
   filtering.

   o 'rt' = "core.osc.mbr" (see Section 7.1).

   The joining node may additionally consider the following parameters
   for the lookup filtering, depending on the information it has already
   available.

   o 'sec-gp', specifying the name of the security group of interest.
      This parameter MUST specify a single value.

   o 'ep', specifying the registered endpoint of the GM.

   o 'app-gp', specifying the name(s) of the application group(s)
      associated with the security group of interest. This parameter
      MAY be included multiple times, and each occurrence MUST specify
      the name of one application group.

   The response from the RD may include links to a group-membership
   resource specifying multiple application groups, as all using the
   same security group. In this case, the joining node is already
   expected to know the exact application group of interest.

   Furthermore, the response from the RD may include the links to
   different group-membership resources, all specifying a same
   application group of interest for the joining node, if the
   corresponding security groups are all used by that application group.

   In this case, application policies on the joining node should define
   how to determine the exact security group to join, depending on what
   exactly the endpoint is intended to do in the application group of
   interest. Later on, the joining node will be anyway able to join
   only security groups for which it is actually authorized to be a
   member (see [I-D.ietf-ace-key-groupcomm-oscore]).

Note that, with RD-based discovery, including the 'app-gp' parameter
multiple times would result in finding only the group-membership
resource that serves all the specified application groups, i.e. not
any group-membership resource that serves either. Therefore, a
joining node needs to perform N separate queries with different
values for 'app-gp', in order to safely discover the (different)
group-membership resource(s) serving the N application groups.

4.1. Discovery Example #1

Consistently with the examples in Section 2 and Section 3, the
examples below consider a joining node that wants to join the
security group associated with the application group "group1", but
that does not know the name of the security group, the responsible GM
and the group-membership resource to access.

4.1.1. Example in Link Format

```
Request: Joining node -> RD

Req: GET coap://rd.example.com/rd-lookup/res
  ?rt=core.osc.mbr&app-gp=group1

Response: RD -> Joining node

Res: 2.05 Content
Payload:
<coap://[2001:db8::ab]/group-oscore/feedca570000>;rt="core.osc.mbr";
    sec-gp="feedca570000";app-gp="group1";
    cs_alg="-8";cs_alg_crv="6";cs_key_kty="1";
    cs_key_crv=6";cs_kenc="1";anchor="coap://[2001:db8::ab]"
```

To retrieve the multicast IP address of the CoAP group used by the
application group "group1", the joining node performs an endpoint
lookup as shown below. The following assumes that the application
group "group1" had been previously registered as per Appendix A of
[I-D.ietf-core-resource-directory], with ff35:30:2001:db8::23 as
multicast IP address of the associated CoAP group.

```
Request: Joining node -> RD

Req: GET coap://rd.example.com/rd-lookup/ep
  ?et=core.rd-group&ep=group1

Response: RD -> Joining node
```

```
   Res: 2.05 Content
   Payload:
   </rd/501>;ep="group1";et="core.rd-group";
            base="coap://[ff35:30:2001:db8::23]"
```

4.1.2. Example in CoRAL

```
   Request: Joining node -> RD

   Req: GET coap://rd.example.com/rd-lookup/res
     ?rt=core.osc.mbr&app-gp=group1
   Accept: TBD123456 (application/coral+cbor)


   Response: RD -> Joining node

   Res: 2.05 Content
   Content-Format: TBD123456 (application/coral+cbor)


   Payload:
   #using <http://coreapps.org/core.oscore-discovery#>
   #using reef = <http://coreapps.org/reef#>
   #using iana = <http://www.iana.org/assignments/relation/>

   #base <coap://[2001:db8::ab]/>
   reef:rd-item </group-oscore/feedca570000> {
      reef:rt "core.osc.mbr"
      sec-gp "feedca570000"
      app-gp "group1"
      cs_alg -8
      cs_alg_crv 6
      cs_key_kty 1
      cs_key_crv 6
      cs_kenc 1
      iana:authorization-server <coap://as.example.com/token>
   }
```

   To retrieve the multicast IP address of the CoAP group used by the
   application group "group1", the joining node performs an endpoint
   lookup as shown below. The following assumes that the application
   group "group1" had been previously registered, with
   ff35:30:2001:db8::23 as multicast IP address of the associated CoAP
   group.

```
   Request: Joining node -> RD

   Req: GET coap://rd.example.com/rd-lookup/ep
     ?et=core.rd-group&ep=group1
   Accept: TBD123456 (application/coral+cbor)
```

```
Response: RD -> Joining node

Res: 2.05 Content
Content-Format: TBD123456 (application/coral+cbor)

Payload:
#using <http://coreapps.org/core.oscore-discovery#>
#using reef = <http://coreapps.org/reef#>

reef:rd-unit <./rd/501> {
   reef:ep="group1"
   reef:et="core.rd-group"
   reef:base <coap://[ff35:30:2001:db8::23]>
}
```

4.2. Discovery Example #2

Consistently with the examples in Section 2 and Section 3, the
examples below consider a joining node that wants to join the
security group with name "feedca570000", but that does not know the
responsible GM, the group-membership resource to access, and the
associated application groups.

The examples also show how the joining node uses CoAP observation
[RFC7641], in order to be notified of possible changes to the
parameters of the group-membership resource. This is also useful to
handle the case where the security group of interest has not been
created yet, so that the joining node can receive the requested
information when it becomes available.

4.2.1. Example in Link Format

```
Request: Joining node -> RD

Req: GET coap://rd.example.com/rd-lookup/res
  ?rt=core.osc.mbr&sec-gp=feedca570000
Observe: 0

Response: RD -> Joining node

Res: 2.05 Content
Observe: 24
Payload:
<coap://[2001:db8::ab]/group-oscore/feedca570000>;rt="core.osc.mbr";
    sec-gp="feedca570000";app-gp="group1";
    cs_alg="-8";cs_alg_crv="6";cs_key_kty="1";
    cs_key_crv=6";cs_kenc="1";anchor="coap://[2001:db8::ab]"
```

Depending on the search criteria, the joining node performing the resource lookup can get large responses. This can happen, for instance, when the lookup request targets all the group-membership resources at a specified GM, or all the group-membership resources of all the registered GMs, as in the example below.

Request: Joining node -> RD

Req: GET coap://rd.example.com/rd-lookup/res?rt=core.osc.mbr

Response: RD -> Joining node

```
Res: 2.05 Content
Payload:
<coap://[2001:db8::ab]/group-oscore/feedca570000>;rt="core.osc.mbr";
    sec-gp="feedca570000";app-gp="group1";
    cs_alg="-8";cs_alg_crv="6";cs_key_kty="1";
    cs_key_crv=6";cs_kenc="1";anchor="coap://[2001:db8::ab]",
<coap://[2001:db8::ab]/group-oscore/ech0ech00000>;rt="core.osc.mbr";
    sec-gp="ech0ech00000";app-gp="group2";
    cs_alg="-8";cs_alg_crv="6";cs_key_kty="1";
    cs_key_crv=6";cs_kenc="1";anchor="coap://[2001:db8::ab]",
<coap://[2001:db8::ab]/group-oscore/abcdef120000>;rt="core.osc.mbr";
    sec-gp="abcdef120000";app-gp="group3";
    app-gp="group4";cs_alg="-8";cs_alg_crv="6";
    cs_key_kty="1";cs_key_crv=6";cs_kenc="1";
    anchor="coap://[2001:db8::ab]"
```

Therefore, it is RECOMMENDED that a joining node which performs a resource lookup with the CoAP Observe option specifies the value of the parameter 'sec-gp' in its GET request sent to the RD.

4.2.2. Example in CoRAL

Request: Joining node -> RD

```
Req: GET coap://rd.example.com/rd-lookup/res
  ?rt=core.osc.mbr&sec-gp=feedca570000
Accept: TBD123456 (application/coral+cbor)
Observe: 0
```

Response: RD -> Joining node

```
   Res: 2.05 Content
   Observe: 24
   Content-Format: TBD123456 (application/coral+cbor)

   Payload:
   #using <http://coreapps.org/core.oscore-discovery#>
   #using reef = <http://coreapps.org/reef#>
   #using iana = <http://www.iana.org/assignments/relation/>

   #base <coap://[2001:db8::ab]/>
   reef:rd-item </group-oscore/feedca570000> {
       reef:rt "core.osc.mbr"
       sec-gp "feedca570000"
       app-gp "group1"
       cs_alg -8
       cs_alg_crv 6
       cs_key_kty 1
       cs_key_crv 6
       cs_kenc 1
       iana:authorization-server <coap://as.example.com/token>
   }
```

5. Use Case Example With Full Discovery

In this section, the discovery of security groups is described to
support the installation process of a lighting installation in an
office building. The described process is a simplified version of
one of many processes.

The process described in this section is intended as an example and
does not have any particular ambition to serve as recommendation or
best practice to adopt. That is, it shows a possible workflow
involving a Commissioning Tool (CT) used in a certain way, while it
is not meant to prescribe how the workflow should necessarily be.

Assume the existence of four luminaires that are members of two
application groups. In the first application group, the four
luminaires receive presence messages and light intensity messages
from sensors or their proxy. In the second application group, the
four luminaires and several other pieces of equipment receive
building state schedules.

Each of the two application groups is associated to a different
security group and to a different CoAP group with its own dedicated
multicast IP address.

The Fairhair Alliance describes how a new device is accepted and
commissioned in the network [Fairhair], by means of its certificate

stored during the manufacturing process. When commissioning the new
device in the installation network, the new device gets a new
identity defined by a newly allocated certificate, following the
BRSKI specification.

Section 7.3 of [I-D.ietf-core-resource-directory] describes how the
CT assigns an endpoint name based on the CN field, (CN=ACME) and the
serial number of the certificate (serial number = 123x, with 3 < x <
8). Corresponding ep-names ACME-1234, ACME-1235, ACME-1236 and
ACME-1237 are also assumed.

It is common practice that locations in the building are specified
according to a coordinate system. After the acceptance of the
luminaires into the installation network, the coordinate of each
device is communicated to the CT. This can be done manually or
automatically.

The mapping between location and ep-name is calculated by the CT.
For instance, on the basis of grouping criteria, the CT assigns: i)
application group "grp_R2-4-015" to the four luminaires; and ii)
application group "grp_schedule" to all schedule requiring devices.
Also, the device with ep name ACME-123x has been assigned IP address:
[2001:db8:4::x]. The RD is assigned IP address: [2001:db8:4:ff].
The used multicast addresses are: [ff05::5:1] and [ff05::5:2].

The following assumes that each device is pre-configured with the
name of the two application groups it belongs to. Additional
mechanisms can be defined in the RD, for supporting devices to
discover the application groups they belong to.

Appendix A provides this same use case example in CoRAL.

*** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***

The CT defines the application group "grp_R2-4-015", with resource
/light and base address [ff05::5:1], as follows.

Request: CT -> RD

Req: POST coap://[2001:db8:4::ff]/rd
  ?ep=grp_R2-4-015&et=core.rd-group&base=coap://[ff05::5:1]
Content-Format: 40
Payload:
</light>;rt="oic.d.light"

Response: RD -> CT

```
Res: 2.01 Created
Location-Path: /rd/501
```

Also, the CT defines a second application group "grp_schedule", with
resource /schedule and base address [ff05::5:2], as follows.

Request: CT -> RD

```
Req: POST coap://[2001:db8:4::ff]/rd
  ?ep=grp_schedule&et=core.rd-group&base=coap://[ff05::5:2]
Content-Format: 40
Payload:
</schedule>;rt="oic.r.time.period"
```

Response: RD -> CT

```
Res: 2.01 Created
Location-Path: /rd/502
```

*** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***

Finally, the CT defines the corresponding security groups. In
particular, assuming a Group Manager responsible for both security
groups and with address [2001:db8::ab], the CT specifies:

Request: CT -> RD

```
Req: POST coap://[2001:db8:4::ff]/rd
  ?ep=gm1&base=coap://[2001:db8::ab]
Content-Format: 40
Payload:
</group-oscore/feedca570000>;ct=41;rt="core.osc.mbr";
                             sec-gp="feedca570000";
                             app-gp="grp_R2-4-015",
</group-oscore/feedsc590000>;ct=41;rt="core.osc.mbr";
                             sec-gp="feedsc590000";
                             app-gp="grp_schedule"
```

Response: RD -> CT

```
Res: 2.01 Created
Location-Path: /rd/4521
```

*** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***

The device with IP address [2001:db8:4::x] can retrieve the multicast
IP address of the CoAP group used by the application group
"grp_R2-4-015", by performing an endpoint lookup as shown below.

Request: Joining node -> RD

Req: GET coap://[2001:db8:4::ff]/rd-lookup/ep
  ?et=core.rd-group&ep=grp_R2-4-015

Response: RD -> Joining node

Res: 2.05 Content
Content-Format: 40
Payload:
</rd/501>;ep="grp_R2-4-015";et="core.rd-group";
        base="coap://[ff05::5:1]"

Similarly, to retrieve the multicast IP address of the CoAP group
used by the application group "grp_schedule", the device performs an
endpoint lookup as shown below.

Request: Joining node -> RD

Req: GET coap://[2001:db8:4::ff]/rd-lookup/ep
  ?et=core.rd-group&ep=grp_schedule

Response: RD -> Joining node

Res: 2.05 Content
Content-Format: 40
Payload:
</rd/502>;ep="grp_schedule";et="core.rd-group";
        base="coap://[ff05::5:2]"

*** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***

Consequently, the device learns the security groups it has to join.
In particular, it does the following for app-gp="grp_R2-4-015".

Request: Joining node -> RD

Req: GET coap://[2001:db8:4::ff]/rd-lookup/res
  ?rt=core.osc.mbr&app-gp=grp_R2-4-015

Response: RD -> Joining Node

Res: 2.05 Content
Content-Format: 40
Payload:
<coap://[2001:db8::ab]/group-oscore/feedca570000>;
    rt="core.osc.mbr";sec-gp="feedca570000";
    app-gp="grp_R2-4-015";anchor="coap://[2001:db8::ab]"

   Similarly, the device does the following for app-gp="grp_schedule".

   Req: GET coap://[2001:db8:4::ff]/rd-lookup/res
     ?rt=core.osc.mbr&app-gp=grp_schedule

   Response: RD -> Joining Node

   Res: 2.05 Content
   Content-Format: 40
   Payload:
   <coap://[2001:db8::ab]/group-oscore/feedsc590000>;
       rt="core.osc.mbr";sec-gp="feedsc590000";
       app-gp="grp_schedule";anchor="coap://[2001:db8::ab]"

   *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***

   After this last discovery step, the device can ask permission to join
   the security groups, and effectively join them through the Group
   Manager, e.g. according to [I-D.ietf-ace-key-groupcomm-oscore].

6. Security Considerations

   The security considerations as described in Section 8 of
   [I-D.ietf-core-resource-directory] apply here as well.

7. IANA Considerations

   This document has the following actions for IANA.

7.1. Resource Types

   IANA is asked to enter the following value into the Resource Type
   (rt=) Link Target Attribute Values subregistry within the Constrained
   Restful Environments (CoRE) Parameters registry defined in [RFC6690].

Value	Description	Reference
core.osc.mbr	Group-membership resource of an OSCORE Group Manager	[[this document]]

8. References

8.1. Normative References

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

[COSE.Elliptic.Curves]
          IANA, "COSE Elliptic Curves",
          <https://www.iana.org/assignments/cose/
          cose.xhtml#elliptic-curves>.

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

[I-D.ietf-core-coral]
          Hartke, K., "The Constrained RESTful Application Language
          (CoRAL)", draft-ietf-core-coral-03 (work in progress),
          March 2020.

[I-D.ietf-core-groupcomm-bis]
          Dijk, E., Wang, C., and M. Tiloca, "Group Communication
          for the Constrained Application Protocol (CoAP)", draft-
          ietf-core-groupcomm-bis-00 (work in progress), March 2020.

[I-D.ietf-core-oscore-groupcomm]
          Tiloca, M., Selander, G., Palombini, F., and J. Park,
          "Group OSCORE - Secure Group Communication for CoAP",
          draft-ietf-core-oscore-groupcomm-09 (work in progress),
          June 2020.

[I-D.ietf-core-resource-directory]
          Shelby, Z., Koster, M., Bormann, C., Stok, P., and C.
          Amsuess, "CoRE Resource Directory", draft-ietf-core-
          resource-directory-24 (work in progress), March 2020.

[I-D.ietf-cose-rfc8152bis-algs]
          Schaad, J., "CBOR Object Signing and Encryption (COSE):
          Initial Algorithms", draft-ietf-cose-rfc8152bis-algs-11
          (work in progress), July 2020.

[I-D.ietf-cose-rfc8152bis-struct]
          Schaad, J., "CBOR Object Signing and Encryption (COSE):
          Structures and Process", draft-ietf-cose-rfc8152bis-
          struct-11 (work in progress), July 2020.

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

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

[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>.

[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>.

[RFC8747] Jones, M., Seitz, L., Selander, G., Erdtman, S., and H.
          Tschofenig, "Proof-of-Possession Key Semantics for CBOR
          Web Tokens (CWTs)", RFC 8747, DOI 10.17487/RFC8747, March
          2020, <https://www.rfc-editor.org/info/rfc8747>.

8.2. Informative References

[Fairhair]
          FairHair Alliance, "Security Architecture for the Internet
          of Things (IoT) in Commercial Buildings", White Paper, ed.
          Piotr Polak , March 2018, <https://openconnectivity.org/
          wp-content/uploads/2019/11/
          fairhair_security_wp_march-2018.pdf>.

[I-D.hartke-t2trg-coral-reef]
          Hartke, K., "Resource Discovery in Constrained RESTful
          Environments (CoRE) using the Constrained RESTful
          Application Language (CoRAL)", draft-hartke-t2trg-coral-
          reef-04 (work in progress), May 2020.

[I-D.ietf-ace-key-groupcomm-oscore]
          Tiloca, M., Park, J., and F. Palombini, "Key Management
          for OSCORE Groups in ACE", draft-ietf-ace-key-groupcomm-
          oscore-07 (work in progress), June 2020.

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

   [I-D.ietf-anima-bootstrapping-keyinfra]
              Pritikin, M., Richardson, M., Eckert, T., Behringer, M.,
              and K. Watsen, "Bootstrapping Remote Secure Key
              Infrastructures (BRSKI)", draft-ietf-anima-bootstrapping-
              keyinfra-41 (work in progress), April 2020.

   [RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object
              Representation (CBOR)", RFC 7049, DOI 10.17487/RFC7049,
              October 2013, <https://www.rfc-editor.org/info/rfc7049>.

   [RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for
              Constrained-Node Networks", RFC 7228,
              DOI 10.17487/RFC7228, May 2014,
              <https://www.rfc-editor.org/info/rfc7228>.

   [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>.

   [RFC8132] van der Stok, P., Bormann, C., and A. Sehgal, "PATCH and
              FETCH Methods for the Constrained Application Protocol
              (CoAP)", RFC 8132, DOI 10.17487/RFC8132, April 2017,
              <https://www.rfc-editor.org/info/rfc8132>.

   [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/info/rfc8259>.

   [RFC8613] Selander, G., Mattsson, J., Palombini, F., and L. Seitz,
              "Object Security for Constrained RESTful Environments
              (OSCORE)", RFC 8613, DOI 10.17487/RFC8613, July 2019,
              <https://www.rfc-editor.org/info/rfc8613>.

Appendix A. Use Case Example With Full Discovery (CoRAL)

   This section provides the same use case example of Section 5, but
   specified in CoRAL [I-D.ietf-core-coral].

   *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***

The CT defines the application group "grp_R2-4-015", with resource
/light and base address [ff05::5:1], as follows.

Request: CT -> RD

Req: POST coap://[2001:db8:4::ff]/rd
Content-Format: TBD123456 (application/coral+cbor)

Payload:
#using reef = <http://coreapps.org/reef#>

#base <coap://[ff05::5:1]/>
reef:ep "grp_R2-4-015"
reef:et "core.rd-group"
reef:rd-item </light> {
    reef:rt "oic.d.light"
}

Response: RD -> CT

Res: 2.01 Created
Location-Path: /rd/501

Also, the CT defines a second application group "grp_schedule", with
resource /schedule and base address [ff05::5:2], as follows.

Request: CT -> RD

Req: POST coap://[2001:db8:4::ff]/rd?ep=grp_schedule&et=core.rd-group
Content-Format: TBD123456 (application/coral+cbor)

Payload:
#using reef = <http://coreapps.org/reef#>

#base <coap://[ff05::5:2]/>
reef:rd-item </schedule> {
    reef:rt "oic.r.time.period"
}

Response: RD -> CT

Res: 2.01 Created
Location-Path: /rd/502

*** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***

Finally, the CT defines the corresponding security groups. In
particular, assuming a Group Manager responsible for both security
groups and with address [2001:db8::ab], the CT specifies:

Request: CT -> RD

Req: POST coap://[2001:db8:4::ff]/rd?ep=gm1
Content-Format: TBD123456 (application/coral+cbor)

Payload:
#using <http://coreapps.org/core.oscore-discovery#>
#using reef = <http://coreapps.org/reef#>

#base <coap://[2001:db8::ab]/>
reef:rd-item </group-oscore/feedca570000> {
   reef:ct 41
   reef:rt "core.osc.mbr"
   sec-gp "feedca570000"
   app-gp "grp_R2-4-015"
}
reef:rd-item </group-oscore/feedsc590000> {
   reef:ct 41
   reef:rt "core.osc.mbr"
   sec-gp "feedsc590000"
   app-gp "grp_schedule"
}

Response: RD -> CT

Res: 2.01 Created
Location-Path: /rd/4521

*** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***

The device with IP address [2001:db8:4::x] can retrieve the multicast
IP address of the CoAP group used by the application group
"grp_R2-4-015", by performing an endpoint lookup as shown below.

Request: Joining node -> RD

Req: GET coap://[2001:db8:4::ff]/rd-lookup/ep
  ?et=core.rd-group&ep=grp_R2-4-015

Response: RD -> Joining node

```
Res: 2.05 Content
Content-Format: TBD123456 (application/coral+cbor)

Payload:
#using reef = <http://coreapps.org/reef#>

#base <coap://[2001:db8:4::ff]/rd/>
reef:rd-unit <501> {
   reef:ep "grp_R2-4-015"
   reef:et "core.rd-group"
   reef:base <coap://[ff05::5:1]/>
}
```

Similarly, to retrieve the multicast IP address of the CoAP group
used by the application group "grp_schedule", the device performs an
endpoint lookup as shown below.

Request: Joining node -> RD

```
Req: GET coap://[2001:db8:4::ff]/rd-lookup/ep
  ?et=core.rd-group&ep=grp_schedule
```

Response: RD -> Joining node

```
Res: 2.05 Content
Content-Format: TBD123456 (application/coral+cbor)

Payload:
#using reef = <http://coreapps.org/reef#>

#base <coap://[2001:db8:4::ff]/rd/>
reef:rd-unit <501> {
   reef:ep "grp_schedule"
   reef:et "core.rd-group"
   reef:base <coap://[ff05::5:2]/>
}
```

*** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***

Consequently, the device learns the security groups it has to join.
In particular, it does the following for app-gp="grp_R2-4-015".

Request: Joining node -> RD

```
Req: GET coap://[2001:db8:4::ff]/rd-lookup/res
  ?rt=core.osc.mbr&app-gp=grp_R2-4-015
```

Response: RD -> Joining Node

```
Res: 2.05 Content
Content-Format: TBD123456 (application/coral+cbor)

Payload:
#using <http://coreapps.org/core.oscore-discovery#>
#using reef = <http://coreapps.org/reef#>

#base <coap://[2001:db8::ab]/>
reef:rd-item </group-oscore/feedca570000> {
   reef:rt "core.osc.mbr"
   sec-gp "feedca570000"
   app-gp "grp_R2-4-015"
}

Similarly, the device does the following for app-gp="grp_schedule".

Req: GET coap://[2001:db8:4::ff]/rd-lookup/res
  ?rt=core.osc.mbr&app-gp=grp_schedule

Response: RD -> Joining Node

Res: 2.05 Content
Content-Format: TBD123456 (application/coral+cbor)

Payload:
#using <http://coreapps.org/core.oscore-discovery#>
#using reef = <http://coreapps.org/reef#>

#base <coap://[2001:db8::ab]/>
reef:rd-item </group-oscore/feedsc590000> {
   reef:rt "core.osc.mbr"
   sec-gp "feedsc590000"
   app-gp "grp_schedule"
}

*** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
```

After this last discovery step, the device can ask permission to join the security groups, and effectively join them through the Group Manager, e.g. according to [I-D.ietf-ace-key-groupcomm-oscore].

Acknowledgments

Authors' Addresses

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

Email: marco.tiloca@ri.se

Christian Amsuess
Hollandstr. 12/4
Vienna 1020
Austria

Email: christian@amsuess.com

Peter van der Stok
Consultant

Phone: +31-492474673 (Netherlands), +33-966015248 (France)
Email: consultancy@vanderstok.org
URI: www.vanderstok.org