ACE Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 6, 2018

M. Tiloca
RISE SICS AB
J. Park
Universitaet Duisburg-Essen
March 05, 2018

Joining OSCORE groups in ACE
draft-tiloca-ace-oscoap-joining-03

Abstract

This document describes a method to join a group where communications are based on CoAP and secured with Object Security for Constrained RESTful Environments (OSCORE). The proposed method delegates the authentication and authorization of 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 profiles of ACE to achieve communication security, proof-of-possession and server authentication.

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 also to protect CoAP group communication over IP multicast [RFC7390]. This relies on a Group Manager entity, which is responsible for managing an OSCORE group, where members exchange CoAP messages secured with OSCORE. In particular, the Group Manager coordinates the join process of new group members and can be responsible for multiple groups.

This specification builds on the ACE framework for Authentication and Authorization [I-D.ietf-ace-oauth-authz] and defines how a client joins an OSCORE group through a resource server acting as Group Manager. The client acting as joining node relies on an Access Token, which is bound to a proof-of-possession key and authorizes the access to a specific join resource at the Group Manager. Messages exchanged among the participants follow the formats defined in

[I-D.palombini-ace-key-groupcomm] for provisioning 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 profiles of ACE. These include [I-D.ietf-ace-dtls-authorize] and [I-D.ietf-ace-oscore-profile], as well as possible forthcoming 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]. Message exchanges are presented as RESTful protocol interactions, for which HTTP [RFC7231] provides useful terminology.

The terminology for entities in the considered architecture is defined in OAuth 2.0 [RFC6749] and [I-D.ietf-ace-actors]. 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]. 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].

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] 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.

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

2. Protocol Overview

Group communication for CoAP over IP multicast has been enabled in [RFC7390] 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 explicitly 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 a network node joins an OSCORE group by using the ACE framework for authentication and authorization [I-D.ietf-ace-oauth-authz]. 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 group.

- 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) enables and enforces the authorized access of joining nodes to join endpoints at the Group Manager, and hence the access to the related OSCORE groups. Multiple Group Managers can be associated to the same AS. The AS is not necessarily expected to release Access Tokens for any other purpose than accessing join resources on registered Group Managers. However, the AS may be configured also to release Access Tokens for accessing resources at members of OSCORE groups.

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

Communications between the joining node and the AS (/token endpoint) as well as between the Group Manager and the AS (/introspection endpoint) can be secured by different means, for instance by means of 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 profile of ACE, and are out of the scope of this specification.

The following steps are performed for joining an OSCORE group. Messages exchanged among the participants follow the formats defined in [I-D.palombini-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) referred in [I-D.palombini-ace-key-groupcomm].

1. The joining node requests an Access Token from the AS to access a join resource on the Group Manager and hence the associated OSCORE group (see Section 3). The response from the AS enables the joining node to start a secure channel with the Group Manager, if not already established.

2. The joining node transfers authentication and authorization information to the Group Manager by posting the obtained Access Token. Then, the joining node and the Group Manager have to establish a secure channel in case one is not already set up (see Section 4). That is, a joining node must establish a secure communication channel with a Group Manager, before joining an OSCORE group under that Group Manager for the first time.

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 in the OSCORE group.

3. Joining Node to Authorization Server

This section considers a joining node that intends to contact the Group Manager for the first time. That is, the joining node has never attempted before to join an OSCORE group under that Group Manager. Also, the joining node and the Group Manager do not have a secure communication channel established.

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 2 of [I-D.ietf-ace-dtls-authorize] to discover the correct AS in charge of the Group Manager. As an alternative, the joining node may look up in a Resource Directory service [I-D.ietf-core-resource-directory].

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.palombini-ace-key-groupcomm]. In particular:

- The "scope" parameter MUST be present and includes:
  - in the first element, the Group Identifier (Gid) of the group to join under the Group Manager. This identifier may not fully coincide with the Gid currently associated to the respective group, e.g. if it includes a dynamic component.
  - in the second element, which MUST be present, the role(s) that the joining node intends to have in the group it intends to join. Roles and their combinations are defined in [I-D.ietf-core-oscore-groupcomm], and indicated as "multicaster", "listener" and "purelistener". Multiple roles are specified in the form of a CBOR array.
- The "aud" parameter MUST be present and is set to the address of the Group Manager.
- The "get_pub_keys" parameter is present only if the Group Manager is configured to store the public keys of the group members and, at the same time, the joining node wants to retrieve such public keys during the joining process (see Section 5). In any other case, this parameter MUST NOT be present.

3.2. Authorization Response

The AS is responsible for authorizing the joining node, accordingly to group join policies enforced on behalf of the 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.palombini-ace-key-groupcomm].

The "exp" parameter MUST be present, since defining the lifetime of Access Tokens is out of the scope of this specification.

In case the value of "scope" specified in the Access Token differs from the value originally included in the Access Token request, the Access Token response MUST include the "scope" parameter, whose second element MUST be present and includes the role(s) 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 profile of ACE to use for securing communications between the joining node and the Group Manager (see Section 5.6.4.4 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 "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

First, the joining node posts the Access Token to the /authz-info endpoint at the Group Manager, in accordance with the Token post defined in Section 3.3 of [I-D.palombini-ace-key-groupcomm]. Then, the joining node establishes a secure channel with the Group Manager, according to what specified in the Access Token response and to the signalled profile of ACE.

4.1. 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 of the Key Distribution Request message defined in Section 4.1 of [I-D.palombini-ace-key-groupcomm]. In particular:

- The "get_pub_keys" parameter can be present only if included also in the Authorization Request previously sent to the AS. In such a case, its value is the same as in the Authorization Request. Otherwise, this parameter MUST NOT be present.

- The "client_cred" parameter, if present, includes the public key or certificate of the joining node. Specifically, it includes the public key of the joining node if the Group Manager is configured to store the public keys of the group members, or the certificate of the joining node otherwise. This parameter MAY be omitted if: i) public keys are used as proof-of-possession keys between the joining node and the Group Manager; or ii) the joining node is asking to access the group exclusively as pure listener; or iii) the Group Manager already acquired this information during a previous join process. In any other case, this parameter MUST be present.

- The "pub_keys_repos" parameter MAY be present if the "client_cred" parameter is both present and with value a certificate of the joining node. If present, this parameter contains the list of public key repositories storing the certificate of the joining node. In any other case, this parameter MUST NOT be present.

4.2. Join Response

The Group Manager processes the request according to [I-D.ietf-ace-oauth-authz]. If this yields to a positive outcome, 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 information necessary to participate in the group communication. This join response follows the format of the Key Distribution success Response message defined in Section 4.2 of [I-D.palombini-ace-key-groupcomm]. In particular:

- The "key" parameter includes what the joining node needs in order to set up the OSCORE Security Context as in Section 2 of [I-D.ietf-core-oscore-groupcomm]. In particular:

  * The "kty" parameter has value "Symmetric".

  * The "k" parameter includes the OSCORE Master Secret.

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

  * The "kid" parameter, if present, has as value the identifier of the key in the parameter "k".

  * The "base IV" parameter, if present, has as value the OSCORE Common IV.

  * The "clientID" parameter MUST be present and has as value the OSCORE Endpoint ID assigned to the joining node by the Group Manager.

  * The "serverID" parameter MUST be present and has as value the Group Identifier (Gid) currently associated to the group.

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

  * The "slt" parameter, if present, has as value the OSCORE Master Salt.

  * The "cs_alg" parameter MUST be present and has as value the countersignature algorithm used in the group.

- 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 non-pure listeners currently in the group, in case the joining node is configured (also) as multicaster; and ii) the public keys of the multicasters currently in the group, in case the joining node is configured (also) as listener or pure listener.

- The "group_policies" parameter SHOULD be present and includes a list of parameters indicating particular policies enforced in the group. For instance, it can indicate the method to achieve synchronization of sequence numbers among group members (see Appendix E of [I-D.ietf-core-oscore-groupcomm]), as well as the

rekeying protocol used to renew the keying material in the group (see Section 2.1 of [I-D.ietf-core-oscore-groupcomm]).

- The "mgt_key_material" parameter SHOULD be present and includes the administrative keying material that the joining node requires to participate in the rekeying process led by the Group Manager. The exact content and format depend on the specific rekeying protocol used in the group.

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 Section 4 of [I-D.ietf-core-oscore-groupcomm].

5. Public Keys of Joining Nodes

Source authentication of OSCORE messages exchanged within the group is ensured by means of digital counter signatures [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 the source authenticity of incoming group messages.

Upon joining an OSCORE group, a joining node is expected to make its own public key available to the other group members, either through the Group Manager or through another trusted, publicly available, key repository. However, this is not required for a node that joins a group exclusively as pure listener.

As also discussed in Section 6 of [I-D.ietf-core-oscore-groupcomm], it is recommended that the Group Manager is configured to store the public keys of the group members and to provide them upon request. If so, three cases can occur when a new node joins a group.

- The Group Manager already acquired the public key of the joining node during a previous join process. In this case, the joining node is not required to provide again its own public key to the Group Manager.

- The joining node and the Group Manager use an asymmetric proof-of-possession key to establish a secure communication channel. In this case, the Group Manager stores the proof-of-possession key conveyed in the Access Token as the public key of the joining node.

- 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 4.1).

Before sending the join response, the Group Manager should verify that the joining node actually owns the associated private key, for instance by performing a proof-of-possession challenge-response, whose details are out of the scope of this specification.

Furthermore, as described in Section 4.1, the joining node may have explicitly requested the Group Manager to retrieve the public keys of the current group members, i.e. through 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 4.2).

On the other hand, in case the Group Manager is not configured to store public keys of group members, the joining node provides the Group Manager with its own certificate in the "client_cred" parameter of the join request targeting the join endpoint (see Section 4.1). Then, the Group Manager validates and handles the certificate, for instance as described in Appendix D.2 of [I-D.ietf-core-oscore-groupcomm].

6. 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 (Section 2.1). This includes the need to revoke and renew the keying material currently used in the OSCORE group, upon changes in the group membership. In particular, renewing the keying material is required upon a new node joining the group, in order to preserve backward security. That is, the Group Manager should renew the keying material before completing the join process and sending a join response. Such a join response provides the joining node with updated the keying material just established in the group. The Group Manager is responsible to enforce rekeying policies and accordingly update the keying material in the groups of its competence (Section 6).

- Synchronization of sequence numbers (Section 5). This concerns how a listener node that has just joined an OSCORE group can synchronize with the sequence number of multicasters in the same group.

- Provisioning and retrieval of public keys (Appendix D.2). This provides guidelines about how to ensure the availability of group members' public keys, possibly relying on the Group Manager as trusted key repository (Section 6).

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

7. IANA Considerations

This document has no actions for IANA.

8. Acknowledgments

The authors sincerely thank Santiago Aragon, Stefan Beck, Martin Gunnarsson, Francesca Palombini, Jim Schaad, Ludwig Seitz and Goeran Selander for their comments and feedback.

The work on this document has been partly supported by the EIT-Digital High Impact Initiative ACTIVE.

9. References

9.1. Normative References

[I-D.ietf-ace-dtls-authorize]
Gerdes, S., Bergmann, O., Bormann, C., Selander, G., and L. Seitz, "Datagram Transport Layer Security (DTLS) Profiles for Authentication and Authorization for Constrained Environments (ACE)", draft-ietf-ace-dtls-authorize-02 (work in progress), October 2017.

[I-D.ietf-ace-oauth-authz]
Seitz, L., Selander, G., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "Authentication and Authorization for Constrained Environments (ACE)", draft-ietf-ace-oauth-authz-10 (work in progress), February 2018.

[I-D.ietf-ace-oscore-profile]
Seitz, L., Palombini, F., and M. Gunnarsson, "OSCORE profile of the Authentication and Authorization for Constrained Environments Framework", draft-ietf-ace-oscore-profile-00 (work in progress), December 2017.

[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-08 (work in progress), January 2018.

[I-D.ietf-core-oscore-groupcomm]
Tiloca, M., Selander, G., Palombini, F., and J. Park, "Secure group communication for CoAP", draft-ietf-core-oscore-groupcomm-01 (work in progress), March 2018.

[I-D.palombini-ace-key-groupcomm]
Palombini, F. and M. Tiloca, "Key Provisioning for Group Communication using ACE", draft-palombini-ace-key-groupcomm-00 (work in progress), March 2018.

[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>.

[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>.

9.2. Informative References

[I-D.ietf-ace-actors]
Gerdes, S., Seitz, L., Selander, G., and C. Bormann, "An architecture for authorization in constrained environments", draft-ietf-ace-actors-06 (work in progress), November 2017.

[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-12 (work in progress), October 2017.

[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>.

[RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <https://www.rfc-editor.org/info/rfc7231>.

[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>.

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

Authors' Addresses

Marco Tiloca
RISE SICS 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