ACE Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 3, 2018

M. Tiloca
RISE SICS AB
J. Park
Universitaet Duisburg-Essen
July 02, 2017

Joining of OSCOAP multicast groups in ACE
draft-tiloca-ace-oscoap-joining-00

Abstract

This document describes a method to join a multicast group where communications are based on CoAP and secured with Object Security of CoAP (OSCOAP). The proposed method delegates the authentication and authorization of client nodes that join a multicast 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.

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 http://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 3, 2018.

Copyright Notice

Table of Contents

1. Introduction

The Constrained Application Protocol (CoAP) [RFC7252] supports also group communication scenarios, where request messages can be delivered to multiple recipients using CoAP on top of IP multicast [RFC7390].

Object Security of CoAP (OSCOAP) [I-D.ietf-core-object-security] is a method for application layer protection of CoAP messages, using the CBOR Object Signing and Encryption (COSE) [I-D.ietf-cose-msg], and enabling end-to-end security of CoAP payload and options.

OSCOAP may also be used to protect group communication for CoAP over IP multicast, as described in [I-D.tiloca-core-multicast-oscoap]. This relies on a Group Manager entity, which is responsible for managing a multicast group where members exchange CoAP messages secured with OSCOAP. In particular, the Group Manager coordinates the join process of new group members and can be responsible for multiple groups.

This document builds on the ACE framework for Authentication and

Authorization [I-D.ietf-ace-oauth-authz] and specifies how a client joins an OSCOAP multicast 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.

The client and the Group Manager leverage protocol-specific profiles of ACE such as [I-D.seitz-ace-oscoap-profile] and [I-D.ietf-ace-dtls-authorize], in order to achieve communication security, proof-of-possession and server authentication.

1.1. Terminology

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

Readers are expected to be familiar with the terms and concepts described in 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). Terminology for constrained environments, such as "constrained device" and "constrained-node network", is defined in [RFC7228].

Readers are expected to be familiar with the terms and concepts related to the CoAP protocol described in [RFC7252][RFC7390]. Note that 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 OSCOAP

[I-D.ietf-core-object-security] also in group communication contexts [I-D.tiloca-core-multicast-oscoap]; and with the OSCOAP profile of ACE described in [I-D.seitz-ace-oscoap-profile].

Readers are expected to be familiar with the terms and concepts related to the DTLS protocol [RFC6347]; the support for DTLS handshake based on Raw Public Keys (RPK) [RFC7250] and on Pre-Shared Keys (PSK) [RFC4279]; and the CoAP-DTLS profile of ACE [I-D.ietf-ace-dtls-authorize].

This document refers also to the following terminology.

- Joining node: a network node intending to join an OSCOAP multicast group, where communication is based on CoAP [RFC7390] and secured with OSCOAP as described in [I-D.tiloca-core-multicast-oscoap].

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

- Join resource: a protected resource hosted by the Group Manager, associated to a multicast group under that Group Manager. A joining node accesses the join resource in order to start the join process and become a member of that group.

- Join endpoint: an endpoint hosted by 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 of CoAP (OSCOAP) [I-D.ietf-core-object-security] as described in [I-D.tiloca-core-multicast-oscoap]. A network node explicitly joins an OSCOAP multicast group, by interacting with the responsible Group Manager. Once registered in the group, the new node can securely exchange (multicast) messages with other group members.

This specification describes how a network node joins an OSCOAP multicast group leveraging 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 owns one join resource for each OSCOAP multicast group it manages. Each join resource is exported by a distinct join endpoint.

- The joining node acts as Client (C), and requests to join an OSCOAP multicast 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. Multiple Group Managers can be associated to the same AS.

If authorized to join the multicast group, the joining node receives from the AS an Access Token bound with a proof-of-possession key. After that, the joining node provides the Group Manager with the Access Token. This step involves the opening of a secure

communication channel between the joining node and the Group Manager, in case they have not already established one.

Finally, the joining node accesses the join endpoint at the Group Manager, so starting the join process to become a member of the multicast group. A same Access Token can authorize the joining node to access multiple groups under the same Group Manager. In such a case, the joining node sequentially performs multiple join processes with the Group Manager, separately for each multicast group to join and by accessing the respective join endpoint.

The AS is not necessarily expected to release Access Tokens for any other purpose than accessing join resources on registered Group Managers. In particular, the AS is not necessarily expected to release Access Tokens for accessing protected resources at members of multicast groups.

The following steps are performed for joining an OSCOAP multicast group, by leveraging the CoAP-DTLS profile of ACE [I-D.ietf-ace-dtls-authorize] or the OSCOAP profile of ACE [I-D.seitz-ace-oscoap-profile].

1. The joining node retrieves an Access Token from the AS to access a join resource on the Group Manager (Section 3). The response from the AS enables the joining node to start a secure channel with the Group Manager, if not already established. The joining node can also contact the AS for updating a previously released Access Token, in order to access further groups under the same Group Manager (Section 6).

2. Authentication and authorization information is transferred between the joining node and the Group Manager, which establish a secure channel in case one is not already set up (Section 4). That is, a joining node MUST establish a secure communication channel with a Group Manager, before joining a multicast group under that Group Manager for the first time.

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

All communications between the involved entities rely on the CoAP protocol and MUST be secured. In particular, communications between the joining node and the AS (/token endpoint) and between the Group Manager and the AS (/introspection endpoint) can be secured by different means, e.g. with DTLS [RFC6347] or with OSCOAP (see Sections 3 and 4 of [I-D.seitz-ace-oscoap-profile]).

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 a multicast 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 one described in Section 2.2 of [I-D.ietf-ace-dtls-authorize] to discover the correct AS in charge of the Group Manager.

The joining node contacts the AS, in order to request an Access Token for accessing the join resource(s) hosted by the Group Manager. In

particular, the Access Token request sent to the /token endpoint specifies the join endpoint(s) of interest at the Group Manager.

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. The same Access Token can authorize the joining node to access multiple groups under the same Group Manager.

Then, the AS provides the joining node with the Access Token, together with an Access Token response. In particular, the Access Token response indicates how to secure communications with the Group Manager, when accessing the join resource(s) for which the Access Token is valid. Specifically, the Access Token response MUST specify one of the following alternatives:

- CoAP over DTLS, i.e. coaps://, indicating to consider the CoAP-DTLS profile of ACE, with asymmetric or symmetric proof-of-possession key (see Section 3 and Section 4 of [I-D.ietf-ace-dtls-authorize], respectively).

- OSCOAP, indicating to consider the OSCOAP profile of ACE with asymmetric or symmetric proof-of-possession key, as described in Section 2.2 of [I-D.seitz-ace-oscoap-profile].

Consistently with the profiles of ACE specified in [I-D.ietf-ace-dtls-authorize] and [I-D.seitz-ace-oscoap-profile], a symmetric proof-of-possession key is generated by the AS, which uses it as proof-of-possession key bound to the Access Token, and provides it to the joining node in the Access Token response. Instead, in case of asymmetric proof-of-possession key, the joining node provides its own public key to the AS in 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 Access Token response.

4. Joining Node to Group Manager

First, the joining node establishes a secure channel with the Group Manager, according to what is specified in the Access Token response.

In particular:

- If the CoAP-DTLS profile of ACE is specified, the joining node MUST upload the Access Token to the /authz-info resource before starting the DTLS handshake. Then, the Group Manager processes the Access Token according to [I-D.ietf-ace-oauth-authz]. If this yields to a positive response, the joining node and the Group Manager establish a DTLS session, as described in Section 3 and Section 4 of [I-D.ietf-ace-dtls-authorize], in case of either asymmetric or symmetric proof-of-possession key, respectively.

- If the OSCOAP profile of ACE is specified, the joining node and the Group Manager establish an OSCOAP channel, as described in Section 2.2 of [I-D.seitz-ace-oscoap-profile]. In particular, if the EDHOC protocol [I-D.selander-ace-cose-ecdhe] is used to this end, the joining node MUST include the Access Token in the EDHOC message_1 sent to the /authz-info resource. The Group Manager processes the Access Token as specified in [I-D.ietf-ace-oauth-authz] and proceeds as defined in Section 2.2 of [I-D.seitz-ace-oscoap-profile].

Once the secure channel with the Group Manager has been established, the joining node requests to join the OSCOAP multicast groups of interest, by accessing the related join resources at the Group Manager. That is, the joining node performs multiple join processes with the Group Manager, separately for each multicast group to join and by accessing the respective join endpoint.

In particular, for each multicast group to join, the joining node sends to the Group Manager a confirmable CoAP request, using the method POST and targeting the join endpoint associated to that multicast group. The request payload specifies the intended role(s) of the joining node in the multicast group, i.e. multicaster and/or (pure) listener [I-D.tiloca-core-multicast-oscoap].

The Group Manager processes the request according to [I-D.ietf-ace-oauth-authz]. If this yields to a positive response, the Group Manager updates the group membership by registering the joining node as a new member of the group. Then, the Group Manager

replies to the joining node including the following pieces of

information in the CoAP response payload:

- An OSCOAP endpoint ID, if the joining node is not configured exclusively as pure listener (see Section 3 of [I-D.tiloca-core-multicast-oscoap]). The Group Manager ensures that each OSCOAP endpoint ID in use is unique within a same multicast group.

- The OSCOAP Security Common Context associated to the joined multicast group (see Section 4 of [I-D.tiloca-core-multicast-oscoap]).

From then on, the joining node is registered as a member of the multicast group, and can exchange group messages secured with OSCOAP as described in Section 5 of [I-D.tiloca-core-multicast-oscoap].

5. Public Keys of Joining Nodes

Source authentication of OSCOAP messages exchanged within the multicast group is ensured by means of digital counter signatures [I-D.tiloca-core-multicast-oscoap]. Therefore, group members must be able to retrieve each other's public key from a trusted key repository, in order to verify the authenticity of incoming group messages. As also discussed in Section 7.4 of [I-D.tiloca-core-multicast-oscoap], the Group Manager can be configured to store public keys of group members and provide them upon request.

Upon joining a multicast 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, if at least one of the following conditions hold.

- The joining node joins a group exclusively as pure listener.

- The joining node joins a group where only group authentication of messages is provided (see Appendix C of [I-D.tiloca-core-multicast-oscoap]).

In case the Group Manager is not configured to store public keys of group members, a joining node SHOULD specify to the Group Manager the address of a trusted key repository where its own public key is available. In particular, upon performing a join process with a given Group Manager for the first time, the joining node additionally includes this information in the payload of the POST request

targeting the join endpoint. The Group Manager can then redirect group members to the correct key repository in case of need.

Instead, in case the Group Manager is configured to store public keys of group members, two main cases can occur.

- The joining node and the Group Manager have used 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 have used 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 includes its own public key in the payload of the POST request targeting the join endpoint. Then, the Group Manager MUST verify that the joining node actually owns the associated private key, for instance by performing a proof-of-possession challenge-response.

Furthermore, if the Group Manager is configured as key repository, it SHOULD provide a joining node with the public keys of the current members in the joined group. In particular, when providing the OSCOAP Endpoint ID and the OSCOAP Security Common Context as described in Section 4, the Group Manager additionally includes the following material in the response to the joining node:

- The public keys of the non-pure listeners currently in the joined multicast group, if the joining node is configured (also) as multicaster.

- The public keys of the multicasters currently in the joined multicast group, if the joining node is configured (also) as non-pure listener.

6. Updating Authorization Information

At any point in time, a node might want to join further OSCOAP multicast groups under the same Group Manager. In such a case, the joining node requests from the AS an updated Access Token for accessing the new multicast groups of interest.

The joining node uploads the new Access Token to the /authz-info resource at the Group Manager, using the already established secure channel. After that, the joining node performs the joining process

described in Section 4, separately for each multicast group to join.

Since the joining node and the Group Manager already share a secure communication channel, they are not required to establish a new one. However, according to the specific profile of ACE in use, the joining node and the Group Manager may leverage the new Access Token to establish a new secure communication channel or update the currently existing one. For instance, Section 4.2 of [I-D.ietf-ace-dtls-authorize] describes how the new Access Token can be used to renegotiate an existing DTLS session or to establish a new one by performing a new DTLS handshake.

7. Security Considerations

This document relies on the security considerations included in Section 7 of [I-D.tiloca-core-multicast-oscoap], as to different management aspects related to OSCOAP multicast groups:

- Management of group keying material (Section 7.2). This includes the need to revoke and renew the keying material currently used in the multicast group, upon changes in the group membership. In particular, renewing the keying material is required upon a new node joining the multicast group, in order to preserve backward security. The Group Manager is responsible to enforce rekeying policies and accordingly update the keying material within the multicast groups of its competence.

- Synchronization of sequence numbers (Section 7.3). This concerns how a listener node that has just joined a multicast group can synchronize with the sender sequence number of multicasters in the same group. To this end, the new listener node performs a challenge-response with a multicaster node, leveraging the Repeat Option for CoAP [I-D.amsuess-core-repeat-request-tag].

- Provisioning of public keys (Section 7.4). 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 5 of this specification leverages and builds on such considerations.

Further security considerations are (going to be) inherited from the

ACE framework for Authentication and Authorization [I-D.ietf-ace-oauth-authz], as well as from the CoAP-DTLS profile [I-D.ietf-ace-dtls-authorize] and the OSCOAP profile [I-D.seitz-ace-oscoap-profile] of ACE.

8. IANA Considerations

This document has no actions for IANA.

9. Acknowledgments

The authors sincerely thank Goeran Selander, Santiago Aragon, Ludwig Seitz and Martin Gunnarsson for their comments and feedback.

10. References

10.1. Normative 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-05 (work in progress), March 2017.

[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-00 (work in progress), June 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-06 (work in progress), March 2017.

[I-D.ietf-core-object-security]
Selander, G., Mattsson, J., Palombini, F., and L. Seitz, "Object Security of CoAP (OSCOAP)", draft-ietf-core-object-security-04 (work in progress), July 2017.

[I-D.seitz-ace-oscoap-profile]
Seitz, L., Gunnarsson, M., and F. Palombini, "OSCOAP profile of ACE", draft-seitz-ace-oscoap-profile-03 (work in progress), June 2017.

[I-D.tiloca-core-multicast-oscoap]
Tiloca, M., Selander, G., and F. Palombini, "Secure group communication for CoAP", draft-tiloca-core-multicast-oscoap-02 (work in progress), July 2017.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://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, <http://www.rfc-editor.org/info/rfc7252>.

10.2. Informative References

[I-D.amsuess-core-repeat-request-tag]
Amsuess, C., Mattsson, J., and G. Selander, "Repeat And Request-Tag", draft-amsuess-core-repeat-request-tag-00 (work in progress), July 2017.

[I-D.ietf-cose-msg]
Schaad, J., "CBOR Object Signing and Encryption (COSE)", draft-ietf-cose-msg-24 (work in progress), November 2016.

[I-D.selander-ace-cose-ecdhe]
Selander, G., Mattsson, J., and F. Palombini, "Ephemeral Diffie-Hellman Over COSE (EDHOC)", draft-selander-ace-cose-ecdhe-06 (work in progress), April 2017.

[RFC4279] Eronen, P., Ed. and H. Tschofenig, Ed., "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)", RFC 4279, DOI 10.17487/RFC4279, December 2005, <http://www.rfc-editor.org/info/rfc4279>.

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

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

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

[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, <http://www.rfc-editor.org/info/rfc7231>.

[RFC7250] Wouters, P., Ed., Tschofenig, H., Ed., Gilmore, J., Weiler, S., and T. Kivinen, "Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", RFC 7250, DOI 10.17487/RFC7250, June 2014, <http://www.rfc-editor.org/info/rfc7250>.

[RFC7390] Rahman, A., Ed. and E. Dijk, Ed., "Group Communication for the Constrained Application Protocol (CoAP)", RFC 7390, DOI 10.17487/RFC7390, October 2014, <http://www.rfc-editor.org/info/rfc7390>.

Authors' Addresses

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

Phone: +46 70 604 65 01
Email: marco.tiloca@ri.se

Jiye Park
Universitaet Duisburg-Essen
Schuetzenbahn 70
Essen 45127
Germany

Phone: +49 201 183-7634
Email: ji-ye.park@uni-due.de