CoRE Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 16, 2018

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

February 12, 2018

Secure group communication for CoAP
draft-ietf-core-oscore-groupcomm-00

Abstract

This document describes a method for protecting group communication over the Constrained Application Protocol (CoAP). The proposed approach relies on Object Security for Constrained RESTful Environments (OSCORE) and the CBOR Object Signing and Encryption (COSE) format. All security requirements fulfilled by OSCORE are maintained for multicast OSCORE request messages and related OSCORE response messages. Source authentication of all messages exchanged within the group is ensured, by means of digital signatures produced through private keys of sender devices and embedded in the protected CoAP messages.

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 August 16, 2018.

Copyright Notice

Table of Contents

1. Introduction

The Constrained Application Protocol (CoAP) [RFC7252] is a web transfer protocol specifically designed for constrained devices and networks [RFC7228].

Group communication for CoAP [RFC7390] addresses use cases where deployed devices benefit from a group communication model, for example to reduce latencies and improve performance. Use cases include lighting control, integrated building control, software and firmware updates, parameter and configuration updates, commissioning of constrained networks, and emergency multicast (see Appendix A). Furthermore, [RFC7390] recognizes the importance to introduce a secure mode for CoAP group communication. This specification defines such a mode.

Object Security for Constrained RESTful Environments (OSCORE)[I-D.ietf-core-object-security] describes a security protocol based on the exchange of protected CoAP messages. OSCORE builds on CBOR Object Signing and Encryption (COSE) [RFC8152] and provides end-to-end encryption, integrity, and replay protection between a sending endpoint and a receiving endpoint across intermediary nodes. To this end, a CoAP message is protected by including payload (if any), certain options, and header fields in a COSE object, which finally replaces the authenticated and encrypted fields in the protected message.

This document describes multicast OSCORE, providing end-to-end security of CoAP messages exchanged between members of a multicast group. In particular, the described approach defines how OSCORE should be used in a group communication context, while fulfilling the same security requirements. That is, end-to-end security is assured for multicast CoAP requests sent by multicaster nodes to the group and for related CoAP responses sent as reply by multiple listener nodes. Multicast OSCORE provides source authentication of all CoAP messages exchanged within the group, by means of digital signatures produced through private keys of sender devices and embedded in the protected CoAP messages. As in OSCORE, it is still possible to simultaneously rely on DTLS to protect hop-by-hop communication between a multicaster node and a proxy (and vice versa), and between a proxy and a listener node (and vice versa).

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 CoAP [RFC7252]; group communication for CoAP [RFC7390]; COSE and counter signatures [RFC8152].

Readers are also expected to be familiar with the terms and concepts for protection and processing of CoAP messages through OSCORE, such as "Security Context", "Master Secret" and "Master Salt", defined in [I-D.ietf-core-object-security].

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

This document refers also to the following terminology.

- Keying material: data that is necessary to establish and maintain secure communication among member of a multicast group. This includes, for instance, keys and IVs [RFC4949].

- Group Manager (GM): entity responsible for creating a multicast group, establishing and provisioning Security Contexts among authorized group members, as well as managing the joining of new group members and the leaving of current group members. A GM can be responsible for multiple multicast groups. Besides, a GM is not required to be an actual group member and to take part in the group communication. The GM is also responsible for renewing/ updating Security Contexts and related keying material in the multicast groups of its competence. Each endpoint in a multicast group securely communicates with the respective GM.

- Multicaster: member of a multicast group that sends multicast CoAP messages intended for all members of the group. In a 1-to-N multicast group, only a single multicaster transmits data to the group; in an M-to-N multicast group (where M and N do not necessarily have the same value), M group members are multicasters. According to [RFC7390], any possible proxy entity is supposed to know about the multicasters in the group and to not perform aggregation of response messages. Also, every multicaster expects and is able to handle multiple response messages associated to a given multicast request message that it has previously sent to the group.

- Listener: member of a multicast group that receives multicast CoAP messages when listening to the multicast IP address associated to the multicast group. A listener may reply back, by sending a

response message to the multicaster which has sent the multicast message.

- Pure listener: member of a multicast group that is configured as listener and never replies back to multicasters after receiving multicast messages.

- Endpoint ID: identifier assigned by the Group Manager to an endpoint upon joining the group as a new member, unless configured exclusively as pure listener. The Group Manager generates and manages Endpoint IDs in order to ensure their uniqueness within a same multicast group. That is, within a single multicast group, the same Endpoint ID cannot be associated to more endpoints at the same time. Endpoint IDs are not necessarily related to any protocol-relevant identifiers, such as IP addresses.

- Group request: multicast CoAP request message sent by a multicaster in the group to all listeners in the group through multicast IP, unless otherwise specified.

- Source authentication: evidence that a received message in the group originated from a specifically identified group member. This also provides assurances that the message was not tampered with either by a different group member or by a non-group member.

2. Assumptions and Security Objectives

This section presents a set of assumptions and security objectives for the approach described in this document.

2.1. Assumptions

The following assumptions are assumed to be already addressed and are out of the scope of this document.

- Multicast communication topology: this document considers both 1-to-N (one multicaster and multiple listeners) and M-to-N (multiple multicasters and multiple listeners) communication topologies. The 1-to-N communication topology is the simplest group communication scenario that would serve the needs of a typical low-power and lossy network (LLN). For instance, in a typical lighting control use case, a single switch is the only entity responsible for sending commands to a group of lighting devices. In more advanced lighting control use cases, a M-to-N communication topology would be required, for instance in case multiple sensors (presence or day-light) are responsible to trigger events to a group of lighting devices.

- Multicast group size: security solutions for group communication should be able to adequately support different, possibly large, group sizes. Group size is the combination of the number of multicasters and listeners in a multicast group, with possible overlap (i.e. a multicaster may also be a listener at the same time). In the use cases mentioned in this document, the number of multicasters (normally the controlling devices) is expected to be much smaller than the number of listeners (i.e. the controlled devices). A security solution for group communication that supports 1 to 50 multicasters would be able to properly cover the group sizes required for most use cases that are relevant for this document. The total number of group members is expected to be in the range of 2 to 100 devices. Groups larger than that should be divided into smaller independent multicast groups, e.g. by grouping lights in a building on a per floor basis.

- Establishment and management of Security Contexts: a Security Context must be established among the group members by the Group Manager which manages the multicast group. A secure mechanism must be used to generate, revoke and (re-)distribute keying material, multicast security policies and security parameters in the multicast group. The actual establishment and management of the Security Context is out of the scope of this document, and it is anticipated that an activity in IETF dedicated to the design of a generic key management scheme will include this feature, preferably based on [RFC3740][RFC4046][RFC4535].

- Multicast data security ciphersuite: all group members MUST agree on a ciphersuite to provide authenticity, integrity and confidentiality of messages in the multicast group. The ciphersuite is specified as part of the Security Context.

- Backward security: a new device joining the multicast group should not have access to any old Security Contexts used before its joining. This ensures that a new group member is not able to decrypt confidential data sent before it has joined the group. The adopted key management scheme should ensure that the Security Context is updated to ensure backward confidentiality. The actual mechanism to update the Security Context and renew the group keying material upon a group member's joining has to be defined as part of the group key management scheme.

- Forward security: entities that leave the multicast group should not have access to any future Security Contexts or message exchanged within the group after their leaving. This ensures that a former group member is not able to decrypt confidential data sent within the group anymore. Also, it ensures that a former member is not able to send encrypted and/or integrity protected

messages to the group anymore. The actual mechanism to update the Security Context and renew the group keying material upon a group member's leaving has to be defined as part of the group key management scheme.

2.2. Security Objectives

The approach described in this document aims at fulfilling the following security objectives:

- Data replay protection: replayed group request messages or response messages MUST be detected.

- Group-level data confidentiality: messages sent within the multicast group SHALL be encrypted if privacy sensitive data is exchanged within the group. In fact, some control commands and/or associated responses could pose unforeseen security and privacy risks to the system users, when sent as plaintext. This document considers group-level data confidentiality since messages are encrypted at a group level, i.e. in such a way that they can be decrypted by any member of the multicast group, but not by an external adversary or other external entities.

- Source authentication: messages sent within the multicast group SHALL be authenticated. That is, it is essential to ensure that a message is originated by a member of the group in the first place (group authentication), and in particular by a specific member of the group (source authentication).

- Message integrity: messages sent within the multicast group SHALL be integrity protected. That is, it is essential to ensure that a message has not been tampered with by an external adversary or other external entities which are not group members.

- Message ordering: it MUST be possible to determine the ordering of messages coming from a single sender endpoint. In accordance with OSCORE [I-D.ietf-core-object-security], this results in providing relative freshness of group requests and absolute freshness of responses. It is not required to determine ordering of messages from different sender endpoints.

3. OSCORE Security Context

To support multicast communication secured with OSCORE, each endpoint registered as member of a multicast group maintains a Security Context as defined in Section 3 of [I-D.ietf-core-object-security]. In particular, each endpoint in a group stores:

1. one Common Context, received from the Group Manager upon joining the multicast group and shared by all the endpoints in the group. All the endpoints in the group agree on the same COSE AEAD algorithm. In addition to what is defined in Section 3 of [I-D.ietf-core-object-security], the Common Context includes the following information.

   * Group Identifier (Gid). Variable length byte string identifying the Security Context and used as Master Salt parameter in the derivation of keying material. The Gid is used together with the multicast IP address of the group to retrieve the Security Context, upon receiving a secure multicast request message (see Section 5.2). The Gid associated to a multicast group is determined by the responsible Group Manager. The choice of the Gid for a given group's Security Context is application specific. However, a Gid MUST be random as well as long enough, in order to achieve a negligible probability of collisions between Group Identifiers from different Group Managers. It is the role of the application to specify how to handle possible collisions. An example of specific formatting of the Group Identifier that would follow this specification is given in Appendix B.

   * Counter signature algorithm. Value identifying the algorithm used for source authenticating messages sent within the group, by means of a counter signature (see Section 4.5 of [RFC8152]). Its value is immutable once the Security Context is established. All the endpoints in the group agree on the same counter signature algorithm. The Group Manager MUST define a list of supported signature algorithms as part of the group communication policy. Such a list MUST include the EdDSA signature algorithm ed25519 [RFC8032].

2. one Sender Context, unless the endpoint is configured exclusively as pure listener. The Sender Context is used to secure outgoing messages and is initialized according to Section 3 of [I-D.ietf-core-object-security], once the endpoint has joined the multicast group. In practice, the sender endpoint shares the same symmetric keying material stored in the Sender Context with all the recipient endpoints receiving its outgoing OSCORE messages. The Sender ID in the Sender Context coincides with the Endpoint ID received upon joining the group. It is responsibility of the Group Manager to assign Endpoint IDs to new joining endpoints in such a way that uniquess is ensured within the multicast group. Besides, in addition to what is defined in [I-D.ietf-core-object-security], the Sender Context stores also the endpoint's public-private key pair.

3. one Recipient Context for each distinct endpoint from which messages are received, used to process such incoming secure messages. The endpoint creates a new Recipient Context upon receiving an incoming message from another endpoint in the group for the first time. In practice, the recipient endpoint shares the symmetric keying material stored in the Recipient Context with the associated other endpoint from which secure messages are received. Besides, in addition to what is defined in [I-D.ietf-core-object-security], each Recipient Context stores also the public key of the associated other endpoint from which secure messages are received.

Upon receiving a secure CoAP message, a recipient endpoint relies on the sender endpoint's public key, in order to verify the counter signature conveyed in the COSE Object.

If not already stored in the Recipient Context associated to the sender endpoint, the recipient endpoint retrieves the public key from a trusted key repository. In such a case, the correct binding between the sender endpoint and the retrieved public key MUST be assured, for instance by means of public key certificates.

It is RECOMMENDED that the Group Manager acts as trusted key repository, and hence is configured to store public keys of group members and provide them to other members of the same group upon request. Possible approaches to provision public keys upon joining the group and to retrieve public keys of group members are discussed in Appendix C.2.

The Sender Key/IV stored in the Sender Context and the Recipient Keys/IVs stored in the Recipient Contexts are derived according to the same scheme defined in Section 3.2 of [I-D.ietf-core-object-security].

3.1. Management of Group Keying Material

The approach described in this specification should take into account the risk of compromise of group members. Such a risk is reduced when multicast groups are deployed in physically secured locations, like lighting inside office buildings. Nevertheless, the adoption of key management schemes for secure revocation and renewal of Security Contexts and group keying material should be considered.

Consistently with the security assumptions in Section 2, it is RECOMMENDED to adopt a group key management scheme, and securely distribute a new value for the Master Secret parameter of the group's Security Context, before a new joining endpoint is added to the group or after a currently present endpoint leaves the group. This is

necessary in order to preserve backward security and forward security in the multicast group. The Group Manager responsible for the group is entrusted with such a task.

In particular, the Group Manager MUST distribute also a new Group Identifier (Gid) for that group, together with a new value for the Master Secret parameter. An example of how this can be done is provided in Appendix B. Then, each group member re-derives the keying material stored in its own Sender Context and Recipient Contexts as described in Section 3, using the updated Group Identifier.

Especially in dynamic, large-scale, multicast groups where endpoints can join and leave at any time, it is important that the considered group key management scheme is efficient and highly scalable with the group size, in order to limit the impact on performance due to the Security Context and keying material update.

4. The COSE Object

When creating a protected CoAP message, an endpoint in the group computes the COSE object using the untagged COSE_Encrypt0 structure [RFC8152] as defined in Section 5 of [I-D.ietf-core-object-security], with the following modifications.

- The value of the "kid" parameter in the "unprotected" field of responses SHALL be set to the Sender ID of the endpoint transmitting the group message.

- The "unprotected" field of the "Headers" field SHALL additionally include the following parameters:

  * gid : its value is set to the Group Identifier (Gid) of the group's Security Context. This parameter MAY be omitted if the message is a CoAP response.

  * countersign : its value is set to the counter signature of the COSE object (Appendix C.3.3 of [RFC8152]), computed by the endpoint by means of its own private key as described in Section 4.5 of [RFC8152].

In particular, "gid" is included as COSE header parameter as defined in Figure 1.

name	label	value type	value registry	description
gid	TBD	bstr		Identifies the OSCORE group Security Context

Figure 1: Additional common header parameter for the COSE object

o The Additional Authenticated Data (AAD) considered to compute the COSE object is extended, in order to include also the Group Identifier (Gid) of the Security Context used to protect the request message. In particular, the "external_aad" in Section 5.3 of [I-D.ietf-core-object-security] SHALL include also gid as follows:

```
external_aad = [
   version : uint,
   alg : int,
   request_kid : bstr,
   request_piv : bstr,
   gid : bstr,
   options : bstr
]
```

o The OSCORE compression defined in Section 8 of [I-D.ietf-core-object-security] is used, with the following additions for the encoding of the object-security option.

  * The fourth least significant bit of the first byte of the object-security option value SHALL be set to 1, to indicate the presence of the "kid" parameter for both multicast requests and responses.

  * The fifth least significant bit of the first byte MUST be set to 1 for multicast requests, to indicate the presence of the Context Hint in the OSCORE payload. The Context Hint flag MAY be set to 1 for responses.

  * The sixth least significant bit of the first byte is set to 1 if the "countersign" parameter is present, or to 0 otherwise. In order to ensure source authentication of group messages as described in this specification, this bit SHALL be set to 1.

  * The Context Hint value encodes the Group Identifier value (Gid) of the group's Security Context.

* The following q bytes (q given by the counter signature algorithm specified in the Security Context) encode the value of the "countersign" parameter including the counter signature of the COSE object.

* The remaining bytes in the Object-Security value encode the value of the "kid" parameter, which is always present both in multicast requests and in responses.

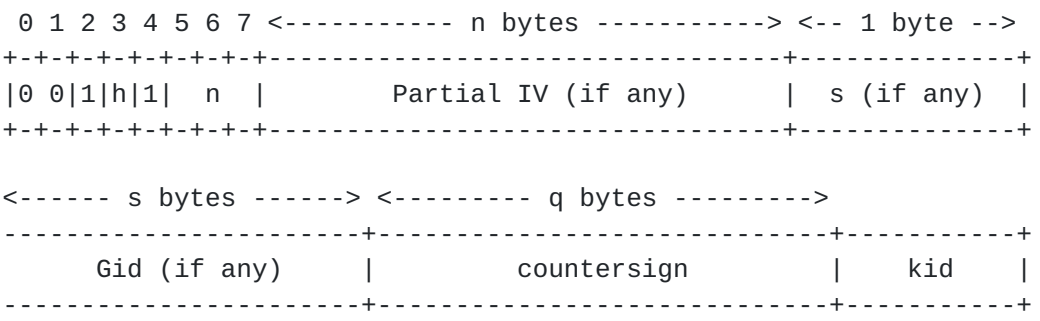

Figure 2: Object-Security Value

5. Message Processing

Each multicast request message and response message is protected and processed as specified in [I-D.ietf-core-object-security], with the modifications described in the following sections.

Furthermore, endpoints in the multicast group locally perform error handling and processing of invalid messages according to the same principles adopted in [I-D.ietf-core-object-security]. However, a receiver endpoint MUST stop processing and silently reject any message which is malformed and does not follow the format specified in Section 4, without sending back any error message. This prevents listener endpoints from sending multiple error messages to a multicaster endpoint, so avoiding the risk of flooding the multicast group.

5.1. Protecting the Request

A multicaster endpoint transmits a secure multicast request message as described in Section 7.1 of [I-D.ietf-core-object-security], with the following modifications.

1. The multicaster endpoint stores the association Token - Group Identifier. That is, it SHALL be able to find the correct Security Context used to protect the multicast request and verify the response(s) by using the CoAP Token used in the message exchange.

2. The multicaster computes the COSE object as defined in Section 4 of this specification.

5.2. Verifying the Request

Upon receiving a secure multicast request message, a listener endpoint proceeds as described in Section 7.2 of [I-D.ietf-core-object-security], with the following modifications.

1. The listener endpoint retrieves the Group Identifier from the "gid" parameter of the received COSE object. Then, it uses the Group Identifier together with the destination IP address of the multicast request message to identify the correct group's Security Context.

2. The listener endpoint retrieves the Sender ID from the "kid" parameter of the received COSE object. Then, the Sender ID is used to retrieve the correct Recipient Context associated to the multicaster endpoint and used to process the request message. When receiving a secure multicast CoAP request message from that multicaster endpoint for the first time, the listener endpoint creates a new Recipient Context, initializes it according to Section 3 of [I-D.ietf-core-object-security], and includes the multicaster endpoint's public key.

3. The listener endpoint retrieves the corresponding public key of the multicaster endpoint from the associated Recipient Context. Then, it verifies the counter signature and decrypts the request message.

5.3. Protecting the Response

A listener endpoint that has received a multicast request message may reply with a secure response message, which is protected as described in Section 7.3 of [I-D.ietf-core-object-security], with the following modifications.

1. The listener endpoint computes the COSE object as defined in Section 4 of this specification.

5.4. Verifying the Response

Upon receiving a secure response message, a multicaster endpoint proceeds as described in Section 7.4 of [I-D.ietf-core-object-security], with the following modifications.

1. The multicaster endpoint retrieves the Security Context by using the Token of the received response message.

2. The multicaster endpoint retrieves the Sender ID from the "kid" parameter of the received COSE object. Then, the Sender ID is used to retrieve the correct Recipient Context associated to the listener endpoint and used to process the response message. When receiving a secure CoAP response message from that listener endpoint for the first time, the multicaster endpoint creates a new Recipient Context, initializes it according to Section 3 of [I-D.ietf-core-object-security], and includes the listener endpoint's public key.

3. The multicaster endpoint retrieves the corresponding public key of the listener endpoint from the associated Recipient Context. Then, it verifies the counter signature and decrypts the response message.

The mapping between response messages from listener endpoints and the associated multicast request message from a multicaster endpoint relies on the 3-tuple (Group ID, Sender ID, Partial IV) associated to the secure multicast request message. This is used by listener endpoints as part of the Additional Authenticated Data when protecting their own response message, as described in Section 4.

6. Synchronization of Sequence Numbers

Upon joining the multicast group, new listeners are not aware of the sequence number values currently used by different multicasters to transmit multicast request messages. This means that, when such listeners receive a secure multicast request from a given multicaster for the first time, they are not able to verify if that request is fresh and has not been replayed. The same applies when a listener endpoint loses synchronization with sequence numbers of multicasters, for instance after a device reboot.

The exact way to address this issue depends on the specific use case and its synchronization requirements. The Group Manager should define also how to handle synchronization of sequence numbers, as part of the policies enforced in the multicast group. In particular, the Group Manager can suggest to single specific listener endpoints how they can exceptionally behave in order to synchronize with sequence numbers of multicasters. Appendix D describes three possible approaches that can be considered.

7. Security Considerations

The same security considerations from OSCORE (Section 11 of [I-D.ietf-core-object-security]) apply to this specification. Additional security aspects to be taken into account are discussed below.

7.1. Group-level Security

The approach described in this document relies on commonly shared group keying material to protect communication within a multicast group. This means that messages are encrypted at a group level (group-level data confidentiality), i.e. they can be decrypted by any member of the multicast group, but not by an external adversary or other external entities.

In addition, it is required that all group members are trusted, i.e. they do not forward the content of group messages to unauthorized entities. However, in many use cases, the devices in the multicast group belong to a common authority and are configured by a commissioner. For instance, in a professional lighting scenario, the roles of multicaster and listener are configured by the lighting commissioner, and devices strictly follow those roles.

8. IANA Considerations

TBD. Header parameter 'gid'.

9. Acknowledgments

The authors sincerely thank Stefan Beck, Rolf Blom, Carsten Bormann, Klaus Hartke, Richard Kelsey, John Mattsson, Jim Schaad and Ludwig Seitz for their feedback and comments.

10. References

10.1. Normative References

[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.

[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>.

[RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, DOI 10.17487/RFC8032, January 2017, <https://www.rfc-editor.org/info/rfc8032>.

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

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

10.2. Informative References

[I-D.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.aragon-ace-ipsec-profile]
Aragon, S., Tiloca, M., and S. Raza, "IPsec profile of ACE", draft-aragon-ace-ipsec-profile-01 (work in progress), October 2017.

[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-09 (work in progress), November 2017.

[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.somaraju-ace-multicast]
Somaraju, A., Kumar, S., Tschofenig, H., and W. Werner, "Security for Low-Latency Group Communication", draft-somaraju-ace-multicast-02 (work in progress), October 2016.

[I-D.tiloca-ace-oscoap-joining]
Tiloca, M. and J. Park, "Joining of OSCORE multicast groups in ACE", draft-tiloca-ace-oscoap-joining-02 (work in progress), October 2017.

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

[RFC2094] Harney, H. and C. Muckenhirn, "Group Key Management Protocol (GKMP) Architecture", RFC 2094, DOI 10.17487/RFC2094, July 1997, <https://www.rfc-editor.org/info/rfc2094>.

[RFC2627] Wallner, D., Harder, E., and R. Agee, "Key Management for Multicast: Issues and Architectures", RFC 2627, DOI 10.17487/RFC2627, June 1999, <https://www.rfc-editor.org/info/rfc2627>.

[RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, DOI 10.17487/RFC3376, October 2002, <https://www.rfc-editor.org/info/rfc3376>.

[RFC3740] Hardjono, T. and B. Weis, "The Multicast Group Security Architecture", RFC 3740, DOI 10.17487/RFC3740, March 2004, <https://www.rfc-editor.org/info/rfc3740>.

[RFC3810] Vida, R., Ed. and L. Costa, Ed., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, DOI 10.17487/RFC3810, June 2004, <https://www.rfc-editor.org/info/rfc3810>.

[RFC4046] Baugher, M., Canetti, R., Dondeti, L., and F. Lindholm, "Multicast Security (MSEC) Group Key Management Architecture", RFC 4046, DOI 10.17487/RFC4046, April 2005, <https://www.rfc-editor.org/info/rfc4046>.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://www.rfc-editor.org/info/rfc4301>.

[RFC4535] Harney, H., Meth, U., Colegrove, A., and G. Gross, "GSAKMP: Group Secure Association Key Management Protocol", RFC 4535, DOI 10.17487/RFC4535, June 2006, <https://www.rfc-editor.org/info/rfc4535>.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007, <https://www.rfc-editor.org/info/rfc4944>.

[RFC4949] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007, <https://www.rfc-editor.org/info/rfc4949>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <https://www.rfc-editor.org/info/rfc6282>.

[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>.

[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>.

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

Appendix A. List of Use Cases

Group Communication for CoAP [RFC7390] provides the necessary background for multicast-based CoAP communication, with particular reference to low-power and lossy networks (LLNs) and resource constrained environments. The interested reader is encouraged to first read [RFC7390] to understand the non-security related details. This section discusses a number of use cases that benefit from secure group communication. Specific security requirements for these use cases are discussed in Section 2.

- Lighting control: consider a building equipped with IP-connected lighting devices, switches, and border routers. The devices are organized into groups according to their physical location in the building. For instance, lighting devices and switches in a room or corridor can be configured as members of a single multicast group. Switches are then used to control the lighting devices by sending on/off/dimming commands to all lighting devices in a group, while border routers connected to an IP network backbone (which is also multicast-enabled) can be used to interconnect routers in the building. Consequently, this would also enable logical multicast groups to be formed even if devices in the lighting group may be physically in different subnets (e.g. on wired and wireless networks). Connectivity between ligthing devices may be realized, for instance, by means of IPv6 and (border) routers supporting 6LoWPAN [RFC4944][RFC6282]. Group communication enables synchronous operation of a group of connected lights, ensuring that the light preset (e.g. dimming level or color) of a large group of luminaires are changed at the same perceived time. This is especially useful for providing a visual synchronicity of light effects to the user. Devices may reply back to the switches that issue on/off/dimming commands, in order to report about the execution of the requested operation (e.g. OK, failure, error) and their current operational status.

- Integrated building control: enabling Building Automation and Control Systems (BACSs) to control multiple heating, ventilation and air-conditioning units to pre-defined presets. Controlled units can be organized into multicast groups in order to reflect their physical position in the building, e.g. devices in the same room can be configured as members of a single multicast group. Furthermore, controlled units are expected to possibly reply back to the BACS issuing control commands, in order to report about the execution of the requested operation (e.g. OK, failure, error) and their current operational status.

- Software and firmware updates: software and firmware updates often comprise quite a large amount of data. This can overload a LLN that is otherwise typically used to deal with only small amounts of data, on an infrequent base. Rather than sending software and firmware updates as unicast messages to each individual device, multicasting such updated data to a larger group of devices at once displays a number of benefits. For instance, it can significantly reduce the network load and decrease the overall time latency for propagating this data to all devices. Even if the complete whole update process itself is secured, securing the individual messages is important, in case updates consist of relatively large amounts of data. In fact, checking individual received data piecemeal for tampering avoids that devices store

large amounts of partially corrupted data and that they detect tampering hereof only after all data has been received. Devices receiving software and firmware updates are expected to possibly reply back, in order to provide a feedback about the execution of the update operation (e.g. OK, failure, error) and their current operational status.

- Parameter and configuration update: by means of multicast communication, it is possible to update the settings of a group of similar devices, both simultaneously and efficiently. Possible parameters are related, for instance, to network load management or network access controls. Devices receiving parameter and configuration updates are expected to possibly reply back, to provide a feedback about the execution of the update operation (e.g. OK, failure, error) and their current operational status.

- Commissioning of LLNs systems: a commissioning device is responsible for querying all devices in the local network or a selected subset of them, in order to discover their presence, and be aware of their capabilities, default configuration, and operating conditions. Queried devices displaying similarities in their capabilities and features, or sharing a common physical location can be configured as members of a single multicast group. Queried devices are expected to reply back to the commissioning device, in order to notify their presence, and provide the requested information and their current operational status.

- Emergency multicast: a particular emergency related information (e.g. natural disaster) is generated and multicast by an emergency notifier, and relayed to multiple devices. The latters may reply back to the emergency notifier, in order to provide their feedback and local information related to the ongoing emergency.

Appendix B. Example of Group Identifier Format

This section provides an example of how the Group Identifier (Gid) can be specifically formatted. That is, the Gid can be composed of two parts, namely a Group Prefix and a Group Epoch.

The Group Prefix is uniquely defined in the set of all the multicast groups associated to the same Group Manager. The choice of the Group Prefix for a given group's Security Context is application specific. Group Prefixes are random as well as long enough, in order to achieve a negligible probability of collisions between Group Identifiers from different Group Managers.

The Group Epoch is set to 0 upon the group's initialization, and is incremented by 1 upon completing each renewal of the Security Context

and keying material in the group (see Section 3.1). In particular, once a new Master Secret has been distributed to the group, all the group members increment by 1 the Group Epoch in the Group Identifier of that group (see Section 3).

Appendix C. Set-up of New Endpoints

An endpoint joins a multicast group by explicitly interacting with the responsible Group Manager. All communications between a joining endpoint and the Group Manager rely on the CoAP protocol and MUST be secured. Specific details on how to secure communications between joining endpoints and a Group Manager are out of the scope of this specification.

In order to receive multicast messages sent to the group, a joining endpoint has to register with a network router device [RFC3376][RFC3810], signaling its intent to receive packets sent to the multicast IP address of that group. As a particular case, the Group Manager can also act as such a network router device. Upon joining the group, endpoints are not required to know how many and what endpoints are active in the same group.

Furthermore, in order to participate in the secure group communication, an endpoint needs to maintain a number of information elements stored in its own Security Context (see Section 3). The following Appendix C.1 describes which of this information is provided to an endpoint upon joining a multicast group through the responsible Group Manager.

C.1. Join Process

An endpoint requests to join a multicast group by sending a confirmable CoAP POST request to the Group Manager responsible for that group. The join request is addressed to a CoAP resource associated to that group and carries the following information.

- Role: the exact role of the joining endpoint in the multicast group. Possible values are: "multicaster", "listener", "pure listener", "multicaster and listener", or "multicaster and pure listener".

- Identity credentials: information elements to enforce source authentication of group messages from the joining endpoint, such as its public key. The exact content depends on whether the Group Manager is configured to store the public keys of group members. If this is the case, this information is omitted if it has been provided to the same Group Manager upon previously joining the same or a different multicast group under its control. This

information is also omitted if the joining endpoint is configured exclusively as pure listener for the joined group. Appendix C.2 discusses additional details on provisioning of public keys and other information to enforce source authentication of joining node's messages.

- Retrieval flag: indication of interest to receive the public keys of the endpoints currently in the multicast group, as included in the following join response. This flag MUST be set to false if the Group Manager is not configured to store the public keys of group members, or if the joining endpoint is configured exclusively as pure listener for the joined group.

The Group Manager MUST be able to verify that the joining enpoint is authorized to become a member of the multicast group. To this end, the Group Manager can directly authorize the joining endpoint, or expect it to provide authorization evidence previously obtained from a trusted entity. Appendix C.3 describes how this can be achieved by leveraging the ACE framework for Authentication and Authorization in constrained environments [I-D.ietf-ace-oauth-authz].

In case of successful authorization check, the Group Manager generates an Endpoint ID assigned to the joining node, before proceeding with the rest of the join process. Instead, in case the authorization check fails, the Group Manager MUST abort the join process. Further details about the authorization of joining endpoint are out of the scope of this specification.

As discussed in Section 3.1, it is then RECOMMENDED that the Security Context is renewed before the joining endpoint becomes a new active member of the multicast group. This is achieved by securely distributing a new Master Secret and a new Group Identifier to the endpoints currently present in the same group.

Once renewed the Security Context in the multicast group, the Group Manager replies to the joining endpoint with a CoAP response carrying the following information.

- Security Common Context: the OSCORE Security Common Context associated to the joined multicast group (see Section 3).

- Endpoint ID: the Endpoint ID associated to the joining node. This information is not included in case "Role" in the join request is equal to "pure listener".

- Management keying material: the set of administrative keying material used to participate in the group rekeying process run by the Group Manager (see Section 3.1). The specific elements of

this management keying material depend on the group rekeying protocol used in the group. For instance, this can simply consist in a group key encryption key and a pairwise symmetric key shared between the joining node and the Group Manager, in case GKMP [RFC2093][RFC2094] is used. Instead, if key-tree based rekeying protocols like LKH [RFC2627] are used, it can consist in the set of symmetric keys associated to the key-tree leaf representing the group member up to the key-tree root representing the group key encryption key.

- Member public keys: the public keys of the endpoints currently present in the multicast group. This includes: the public keys of the non-pure listeners currently in the group, if the joining endpoint is configured (also) as multicaster; and the public keys of the multicasters currently in the group, if the joining endpoint is configured (also) as listener or pure listener. This information is omitted in case the Group Manager is not configured to store the public keys of group members or if the "Retrieval flag" was set to false in the join request. Appendix C.2 discusses additional details on provisioning public keys upon joining the group and on retrieving public keys of group members.

C.2. Provisioning and Retrieval of Public Keys

As mentioned in Section 3, it is RECOMMENDED that the Group Manager acts as trusted key repository, stores public keys of group members and provide them to other members of the same group upon request. In such a case, a joining endpoint provides its own public key to the Group Manager, as "Identity credentials" of the join request, when joining the multicast group (see Appendix C.1).

After that, the Group Manager MUST verify that the joining endpoint actually owns the associated private key, for instance by performing a proof-of-possession challenge-response. In case of success, the Group Manager stores the received public key as associated to the joining endpoint and its Endpoint ID, before sending the join response and continuing with the rest of the join process. From then on, that public key will be available for secure and trusted delivery to other endpoints in the multicast group.

The joining node does not have to provide its own public key if that already occurred upon previously joining the same or a different multicast group under the same Group Manager. However, separately for each multicast group under its control, the Group Manager maintains an updated list of active Endpoint IDs associated to a same endpoint's public key.

Instead, in case the Group Manager does not act as trusted key repository, the following information is exchanged with the Group Manager during the join process.

1. The joining endpoint signs its own certificate by using its own private key. There is no restriction on the Certificate Subject included in the joining node's certificate.

2. The joining endpoint includes the following information as "Identity credentials" in the join request (Appendix C.1): the signed certificate; and the identifier of the Certification Authority that issued the certificate. The joining endpoint can optionally specify also a list of public key repositories storing its own certificate.

3. When processing the join request, the Group Manager first validates the certificate by verifying the signature of the issuer CA, and then verifies the signature of the joining node.

4. The Group Manager stores the association between the Certificate Subject of the joining node's certificate and the pair {Group ID, Endpoint ID of the joining node}. If received from the joining endpoint, the Group Manager also stores the list of public key repositories storing the certificate of the joining endpoint.

When a group member X wants to retrieve the public key of another group member Y in the same multicast group, the endpoint X proceeds as follows.

1. The endpoint X contacts the Group Manager, specifying the pair {Group ID, Endpoint ID of the endpoint Y}.

2. The Group Manager provides the endpoint X with the Certificate Subject CS from the certificate of endpoint Y. If available, the Group Manager provides the endpoint X also with the list of public key repositories storing the certificate of the endpoint Y.

3. The endpoint X retrieves the certificate of the endpoint X from a key repository storing it, by using the Certificate Subject CS.

C.3. Group Joining Based on the ACE Framework

The join process to register an endpoint as a new member of a multicast group can be based on the ACE framework for Authentication and Authorization in constrained environments [I-D.ietf-ace-oauth-authz], built on re-use of OAuth 2.0 [RFC6749].

In particular, the approach described in [I-D.tiloca-ace-oscoap-joining] uses the ACE framework to delegate the authentication and authorization of joining endpoints to an Authorization Server in a trust relation with the Group Manager. At the same time, it allows a joining endpoint to establish a secure channel with the Group Manager, by leveraging protocol-specific profiles of ACE [I-D.ietf-ace-oscore-profile] [I-D.ietf-ace-dtls-authorize] [I-D.aragon-ace-ipsec-profile] to achieve communication security, proof-of-possession and server authentication.

More specifically and with reference to the terminology defined in OAuth 2.0:

- The joining endpoint acts as Client;
- The Group Manager acts as Resource Server, with different CoAP resources for different multicast groups it is responsible for;
- An Authorization Server enables and enforces authorized access of the joining endpoint to the Group Manager and its CoAP resources paired with multicast groups to join.

Both the joining endpoint and the Group Manager MUST adopt secure communication also for any message exchange with the Authorization Server. To this end, different alternatives are possible, such as OSCORE, DTLS [RFC6347] or IPsec [RFC4301].

Appendix D. Examples of Synchronization Approaches

This section describes three possible approaches that can be considered by listener endpoints to synchronize with sequence numbers of multicasters.

D.1. Best-Effort Synchronization

Upon receiving a multicast request from a multicaster, a listener endpoint does not take any action to synchonize with the sequence number of that multicaster. This provides no assurance at all as to message freshness, which can be acceptable in non-critical use cases.

D.2. Baseline Synchronization

Upon receiving a multicast request from a given multicaster for the first time, a listener endpoint initializes its last-seen sequence number in its Recipient Context associated to that multicaster. However, the listener drops the multicast request without delivering it to the application layer. This provides a reference point to

identify if future multicast requests from the same multicaster are fresher than the last one received.

A replay time interval exists, between when a possibly replayed message is originally transmitted by a given multicaster and the first authentic fresh message from that same multicaster is received. This can be acceptable for use cases where listener endpoints admit such a trade-off between performance and assurance of message freshness.

D.3. Challenge-Response Synchronization

A listener endpoint performs a challenge-response exchange with a multicaster, by using the Repeat Option for CoAP described in Section 2 of [I-D.amsuess-core-repeat-request-tag].

That is, upon receiving a multicast request from a particular multicaster for the first time, the listener processes the message as described in Section 5.2 of this specification, but, even if valid, does not deliver it to the application. Instead, the listener replies to the multicaster with a 4.03 Forbidden response message including a Repeat Option, and stores the option value included therein.

Upon receiving a 4.03 Forbidden response that includes a Repeat Option and originates from a verified group member, a multicaster MUST send a group request as a unicast message addressed to the same listener, echoing the Repeat Option value. In particular, the multicaster does not necessarily resend the same group request, but can instead send a more recent one, if the application permits it. This makes it possible for the multicaster to not retain previously sent group requests for full retransmission, unless the application explicitly requires otherwise. In either case, the multicaster uses the sequence number value currently stored in its own Sender Context. If the multicaster stores group requests for possible retransmission with the Repeat Option, it should not store a given request for longer than a pre-configured time interval. Note that the unicast request echoing the Repeat Option is correctly treated and processed as a group message, since the "gid" field including the Group Identifier of the OSCORE group is still present in the Object-Security Option as part of the COSE object (see Section 4).

Upon receiving the unicast group request including the Repeat Option, the listener verifies that the option value equals the stored and previously sent value; otherwise, the request is silently discarded. Then, the listener verifies that the unicast group request has been received within a pre-configured time interval, as described in [I-D.amsuess-core-repeat-request-tag]. In such a case, the request

is further processed and verified; otherwise, it is silently discarded. Finally, the listener updates the Recipient Context associated to that multicaster, by setting the Replay Window according to the Sequence Number from the unicast group request conveying the Repeat Option. The listener either delivers the request to the application if it is an actual retransmission of the original one, or discard it otherwise. Mechanisms to signal whether the resent request is a full retransmission of the original one are out of the scope of this specification.

In case it does not receive a valid group request including the Repeat Option within the configured time interval, the listener node SHOULD perform the same challenge-response upon receiving the next multicast request from that same multicaster.

A listener SHOULD NOT deliver group request messages from a given multicaster to the application until one valid group request from that same multicaster has been verified as fresh, as conveying an echoed Repeat Option [I-D.amsuess-core-repeat-request-tag]. Also, a listener MAY perform the challenge-response described above at any time, if synchronization with sequence numbers of multicasters is (believed to be) lost, for instance after a device reboot. It is the role of the application to define under what circumstances sequence numbers lose synchronization. This can include a minimum gap between the sequence number of the latest accepted group request from a multicaster and the sequence number of a group request just received from the same multicaster. A multicaster MUST always be ready to perform the challenge-response based on the Repeat Option in case a listener starts it.

Note that endpoints configured as pure listeners are not able to perform the challenge-response described above, as they do not store a Sender Context to secure the 4.03 Forbidden response to the multicaster. Therefore, pure listeners should adopt alternative approaches to achieve and maintain synchronization with sequence numbers of multicasters.

This approach provides an assurance of absolute message freshness. However, it can result in an impact on performance which is undesirable or unbearable, especially in large multicast groups where many nodes at the same time might join as new members or lose synchronization.

Appendix E. No Verification of Signatures

There are some application scenarios using group communications that have particularly strict requirements. One example of this is the requirement of low message latency in non-emergency lighting

applications [I-D.somaraju-ace-multicast]. For those applications which have tight performance constraints and relaxed security requirements, it can be inconvenient for some endpoints to verify digital signatures in order to assert source authenticity of received group messages. In other cases, the signature verification can be deferred or only checked for specific actions. For instance, a command to turn a bulb on where the bulb is already on does not need the signature to be checked. In such situations, the counter signature needs to be included anyway as part of the group message, so that an endpoint that needs to validate the signature for any reason has the ability to do so.

In this specification, it is NOT RECOMMENDED that endpoints do not verify the counter signature of received group messages. However, it is recognized that there may be situations where it is not always required. The consequence of not doing the signature validation is that security in the group is based only on the group-authenticity of the shared keying material used for encryption. That is, endpoints in the multicast group have evidence that a received message has been originated by a group member, although not specifically identifiable in a secure way. This can violate a number of security requirements, as the compromise of any element in the group means that the attacker has the ability to control the entire group. Even worse, the group may not be limited in scope, and hence the same keying material might be used not only for light bulbs but for locks as well. Therefore, extreme care must be taken in situations where the security requirements are relaxed, so that deployment of the system will always be done safely.

Authors' Addresses

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

Email: marco.tiloca@ri.se

Goeran Selander
Ericsson AB
Torshamnsgatan 23
Kista SE-16440 Stockholm
Sweden

Email: goran.selander@ericsson.com

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

Email: francesca.palombini@ericsson.com

Jiye Park
Universitaet Duisburg-Essen
Schuetzenbahn 70
Essen 45127
Germany

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