CoRE Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 2, 2018

M. Tiloca
RISE SICS AB
G. Selander
F. Palombini
Ericsson AB

July 01, 2017

Secure group communication for CoAP
draft-tiloca-core-multicast-oscoap-02

Abstract

This document describes a method for protecting group communication over the Constrained Application Protocol (CoAP). The proposed approach relies on Object Security of CoAP (OSCOAP) and the CBOR Object Signing and Encryption (COSE) format. All security requirements fulfilled by OSCOAP are maintained for multicast OSCOAP request messages and related unicast OSCOAP 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.

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 B).

Furthermore, [RFC7390] recognizes the importance to introduce a secure mode for CoAP group communication. This specification defines such a mode.

Object Security of CoAP (OSCOAP)[I-D.ietf-core-object-security] describes a security protocol based on the exchange of protected CoAP messages. OSCOAP builds on CBOR Object Signing and Encryption (COSE) [I-D.ietf-cose-msg] 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 OSCOAP, providing end-to-end security of CoAP messages exchanged between members of a multicast group. In particular, the described approach defines how OSCOAP 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 unicast CoAP responses sent as reply by multiple listener nodes. Multicast OSCOAP 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 OSCOAP, 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", "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 CoAP [RFC7252]; group communication for CoAP [RFC7390]; COSE and counter signatures [I-D.ietf-cose-msg].

Readers are also expected to be familiar with the terms and concepts for protection and processing of CoAP messages through OSCOAP, 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.

- 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 unicast 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.

- 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. Prerequisites and Requirements

The following security prerequisites are assumed to be already fulfilled and are out of the scope of this document.

- Establishment and management of a security context: 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.

The following security requirements need to be fulfilled by the approach described in 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.

- Data replay protection: it must be possible to detect a replayed group request message or response message.

- 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 OSCOAP [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. Set-up Phase

An endpoint joins a multicast group by explicitly interacting with the responsible Group Manager. The actual join process can be based on the ACE framework [I-D.ietf-ace-oauth-authz] and the OSCOAP profile of ACE [I-D.seitz-ace-oscoap-profile], as discussed in Appendix A.

An endpoint registered as member of a group can behave as a multicaster and/or as a listener. As a multicaster, it can transmit multicast request messages to the group. As a listener, it receives multicast request messages from any multicaster in the group, and possibly replies by transmitting unicast response messages. A pure listener never replies to multicast request messages. Upon joining the group, endpoints are not required to know how many and what endpoints are active in the same group. A number of use cases that benefit from secure group communication are discussed in Appendix B.

An endpoint is identified by an endpoint ID provided by the Group Manager upon joining the group, unless configured exclusively as pure listener. That is, pure listener endpoints are not associated to and are not provided with an endpoint ID. 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.

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. Those include keying material used to protect and verify group messages, as well as the public keys of other endpoints in the groups, in order to verify digital signatures of secure messages and ensure their source authenticity. The Group Manager provides these pieces of information to an endpoint upon its joining, through out-of-band means or other pre-established secure channels. Further details about establishment, revocation and renewal of the security context and keying material are out of the scope of this document.

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 unicast response messages associated to a given multicast request message.

4. Security Context

To support multicast communication secured with OSCOAP, 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. Besides, in addition to what is defined in [I-D.ietf-core-object-security], the Common Context stores the following parameters:

   * Context Identifier (Cid). Variable length byte string that identifies the Security Context. The Cid used in a multicast group is determined by the responsible Group Manager and does not change over time. A Cid MUST be unique in the set of all the multicast groups associated to the same Group Manager. The choice of the Cid for a given group's Security Context is application specific. However, Cids MUST be random as well as long enough so that the probability of collisions is negligible and Context Identifiers are globally unique. It is the role of the application to specify how to handle possible collisions.

   * Counter signature algorithm. Value that identifies the algorithm used for source authenticating messages sent within the group, by means of a counter signature (see Section 4.5 of [I-D.ietf-cose-msg]). Its value is immutable once the security context is established. All the endpoints in the group agree on the same counter signature algorithm. In the absence of an application profile standard specifying otherwise, a compliant application MUST implement 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 OSCOAP messages. The Sender ID in the Sender Context coincides with the endpoint ID received upon joining the group. As stated in

Section 3, 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. Possible approaches to provision and retrieve public keys of group members are discussed in Section 7.4.

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].

The 3-tuple (Cid, Sender ID, Partial IV) is called Transaction Identifier (Tid), and SHALL be unique for each Master Secret. The Tid is used as a unique challenge in the COSE object of the protected CoAP request. The Tid is part of the Additional Authenticated Data (AAD, see Section 5.2 of [I-D.ietf-core-object-security]) of the protected CoAP response message, which is how unicast responses are bound to multicast requests.

5. Message Processing

Each multicast request message and unicast response message is protected and processed as specified in [I-D.ietf-core-object-security], with the modifications described in the following sections. Furthermore, error handling and processing of invalid messages are performed according to the same principles adopted in [I-D.ietf-core-object-security]. In particular, a receiver endpoint MUST stop processing and reject any message which is malformed and does not follow the format specified in Section 6.

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 - Cid. That is, it SHALL be able to find the correct Security Context used to protect the multicast request and verify the unicast response(s) by using the CoAP Token used in the message exchange.

2. The multicaster endpoint computes the COSE object as defined in Section 6 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 Context Identifier from the "gid" parameter of the received COSE object, and uses it to identify the correct group's Security Context.

2. The listener endpoint retrieves the Sender ID from the header of the 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 unicast 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 retrieves the Transaction Identifier (Tid) as defined in Section 4 of this specification.

2. The listener endpoint computes the COSE object as defined in Section 6 of this specification.

5.4. Verifying the Response

Upon receiving a secure unicast 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 identified by the Token of the received response message.

2. The multicaster endpoint retrieves the Sender ID from the header of the 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 unicast response messages from listener endpoints and the associated multicast request message from a multicaster endpoint relies on the Transaction Identifier (Tid) associated to the secure multicast request message. The Tid is used by listener endpoints as part of the Additional Authenticated Data when protecting their own response message, as described in Section 4.

6. The COSE Object

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

1. The value of the "Partial IV" parameter in the "unprotected" field is set to the Sequence Number used to protect the message, and SHALL always be present in both multicast requests and unicast responses. Specifically, a multicaster endpoint sets the value of "Partial IV" to the Sequence Number from its own Sender Context, upon sending a multicast request message. Furthermore, unlike described in Section 5 of [I-D.ietf-core-object-security],

a listener endpoint explicitly sets the value of "Partial IV" to the Sequence Number from its own Sender Context, upon sending a unicast response message.

2. The value of the "kid" parameter in the "unprotected" field is set to the Sender ID of the endpoint and SHALL always be present in both multicast requests and unicast responses.

3. The "unprotected" field of the "Headers" field SHALL include also the following parameters:

   * gid : its value is set to the Context Identifier (Cid) 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 [I-D.ietf-cose-msg]), computed by the endpoint by means of its own private key as described in Section 4.5 of [I-D.ietf-cose-msg].

4. The Additional Authenticated Data (AAD) considered to compute the COSE object is extended. In particular, the "external_aad" considered for secure response messages SHALL include also the following parameter:

   * gid : bstr, contains the Context Idenfier (Cid) of the Security Context considered to protect the request message (which is same as the Cid considered to protect the response message).

5. The compressed version of COSE 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 three least significant bit of the first byte SHALL NOT have value 0, since the "Partial IV" parameter is always present for both multicast requests and unicast responses.

   * The fourth least significant bit of the first byte SHALL be set to 1, to indicate the presence of the "kid" parameter in the compressed message for both multicast requests and unicast responses.

   * The fifth least significant bit of the first byte is set to 1 if the "gid" parameter is present, or to 0 otherwise. In order to enable secure group communication as described in this specification, this bit SHALL be set to 1 for multicast requests.

* 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 following n bytes (n being the value of the Partial IV size in the first byte) encode the value of the "Partial IV", which is always present in the compressed message.

* The following byte encodes the size of the "kid" parameter and SHALL NOT have value 0.

* The following m bytes (m given by the previous byte) encode the value of the "kid" parameter.

* The following byte encodes the size of the "gid" parameter and SHALL NOT have value 0.

* The following p bytes (p given by the previous byte) encode the value of the "gid" parameter.

* 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 remainining bytes encode the ciphertext.

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

name	label	value type	value registry	description
gid	TBD	bstr		Identifies the OSCOAP group security context

Table 1: Additional common header parameter for the COSE object

Appendix C discusses a possible alternative configuration of the Object-Security option, to avoid the usage of digital signatures and provide only group authentication of secure CoAP messages. This can be required by application scenarios that have particularly strict requirements such as low message latency (see Section 3 of

[I-D.somaraju-ace-multicast]), and thus cannot afford digital signatures. However, such a purely symmetric approach does not provide source authentication of group messages, and thus is NOT RECOMMENDED by this specification.

Appendix D discusses a possible alternative configuration of the Object-Security option, to include digital signatures in OSCOAP messages exchanged between two endpoints engaging pure unicast communication.

7. Security Considerations

The same security considerations from OSCOAP (Section 10 of [I-D.ietf-core-object-security]) apply to this specification. Furthermore, 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.

7.2. Management of Group Keying Material

The presented approach should take into consideration 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. The adoption of key management schemes for secure revocation and renewal of security contexts and group keying material should be considered.

As stated in Section 2, it is RECOMMENDED to adopt a group key management scheme that updates the security context and keying material in the group, before a new endpoint joins 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.

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.

7.3. 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. In order to address this issue, a listener can perform 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 Context Identifier of the OSCOAP group's Security Context is still present in the Object-Security Option as part of the COSE object (see Section 6).

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 Sequence Number to the value included in 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.

7.4. Provisioning of Public Keys

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. Further details about how this requirement can be fulfilled are out of the scope of this document.

Alternatively, the Group Manager can be configured to store public keys of group members and provide them upon request. In such a case, upon joining a multicast group, an endpoint provides its own public key to the Group Manager, by means of the same secure channel used to carry out the join procedure. 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. From then on, that public key will be available for secure and trusted delivery to other endpoints in the multicast group.

Note that a joining endpoint is not required to provide its own public key to the Group Manager in the following two cases. First, the endpoint is joining the multicast group exclusively as pure listener. Second, the endpoint has already provided its own public key, upon previously joining a multicast group under the same Group Manager.

Furthermore, in simple, less dynamic, multicast groups, it can be convenient for the Group Manager to provide an endpoint upon its joining with the public keys associated to the endpoints currently present in the group.

8. IANA Considerations

TBD. Header parameter 'gid'.

9. Acknowledgments

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

10. References

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

[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>.

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

10.2. Informative References

[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.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.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.

[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.

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

[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, <http://www.rfc-editor.org/info/rfc4046>.

[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, <http://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, <http://www.rfc-editor.org/info/rfc4944>.

[RFC4949] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007, <http://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, <http://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, <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>.

[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>.

Appendix A. 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 [I-D.ietf-ace-oauth-authz] and the OSCOAP profile of ACE [I-D.seitz-ace-oscoap-profile]. With reference to the terminology defined in OAuth 2.0 [RFC6749]:

- o The joining endpoint acts as Client;
- o The Group Manager acts as Resource Server, exporting one join-resource for each multicast group it is responsible for;
- o An Authorization Server enables and enforces authorized access of the joining endpoint to the Group Manager and its join-resources.

Then, in accordance with [I-D.seitz-ace-oscoap-profile], the joining endpoint and the Group Manager rely on OSCOAP [I-D.ietf-core-object-security] for secure communication and can use Ephemeral Diffie-Hellman Over COSE (EDHOC) [I-D.selander-ace-cose-ecdhe] as a possible method to establish key material.

The joining endpoint sends to the Group Manager an OSCOAP request to access the join-resource associated to the multicast group to join. The Group Manager replies with an OSCOAP response including the Common Context associated to that group (see Section 4). In case the Group Manager is configured to store the public keys of group members, the joining endpoint additionally provides the Group Manager with its own public key, and MAY obtain from the Group Manager the public keys of the endpoints currently present in the group (see Section 7.4).

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, including OSCOAP and DTLS [RFC6347].

Appendix B. 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.

o 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.

o 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.

o 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.

o 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.

o 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.

o 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 C. No Source Authentication

Some application scenarios based on group communication can display particularly strict requirements, for instance low message latency in non-emergency lighting applications [I-D.somaraju-ace-multicast]. For such and similar applications, it can be inconvenient or even infeasible to ensure source authentication of group messages through approaches based on digital signatures.

Due to such performance contraints and given the more relaxed security requirements of such non-critical applications, it can be acceptable to provide only group authentication of messages exchanged within the group. This can be achieved by authenticating group messages through a key which either is commonly shared among group members or can be derived by any of them. As a result, there is evidence that a given message has been originated by a group member, although not specifically identifiable.

Although this is NOT RECOMMENDED by this specification, it is possible to avoid digital signing of group messages and provide only their group authentication as follows.

o In every Security Context (Section 4): the Common Context has the "Counter signature algorithm" field set to NULL; the Sender Context does not include the key pair associated to the endpoint; each Recipient Context does not include the public key associated to the respective endpoint.

o When encoding the Object-Security option of a group message (Section 6), the sixth least significant bit of the first byte is set to 0, to indicate that the "countersign" parameter including the counter signature of the COSE object is not present.

o No counter signature is computed when securing a multicast request (Section 5.1) or a unicast response (Section 5.3), while no counter signature is verified upon receiving a multicast request (Section 5.2) or a unicast response (Section 5.4).

As a consequence, each message is group-authenticated by means of the AEAD algorithm and the Sender Key/IV used by the sender endpoint. Note that such Sender Key/IV can be derived by all the group members from the Sender ID and the commonly shared Master Secret and Master Salt.

Appendix D. Unicast OSCOAP Messages with Digital Signature

Two endpoints engaging pure unicast communication secured with OSCOAP can benefit from exchanging digitally signed messages. This especially applies to scenarios where end-to-end confidentiality is not a security requirement to fulfill, and thus proxies are able to fully inspect, process and aggregate messages, while still not able to alter them.

With reference to two endpoints using OSCOAP [I-D.ietf-core-object-security] for pure unicast communication, digital signing of exchanged messages can be enabled as follows.

- Each of the two endpoint additionally stores in the Security Context: i) the respective public-private key pair; ii) the other endpoint's public key; iii) a "Counter signature algorithm" field as defined in Section 4 of this specification.

- The Object-Security option of OSCOAP messages is encoded as described in Section 8.1 of [I-D.ietf-core-object-security], with the following differences.

  * The fifth least significant bit of the first byte is set to 0, to indicate that the "gid" parameter introduced in this specification and including the Context Identifier of an OSCOAP multicast group is not present.

  * The sixth least significant bit of the first byte is set to 1, to indicate the presence of the "countersign" parameter introduced in this specification and including the counter signature of the COSE object.

  * The q bytes before the "ciphertext" field (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.

- Before transmitting an OSCOAP message, a sender endpoint uses its own private key to create a counter signature of the COSE object (Appendix C.4 of [I-D.ietf-cose-msg]). Then, the counter signature is included in the Header of the COSE object, in the "countersign" paramenter of the "unprotected" field.

- Upon receiving an OSCOAP message, the receiver endpoint retrieves the corresponding public key of the sender endpoint from the Security Context. Then, it verifies the counter signature and unsecures the message according to the cryptographic algorithm specified in the Security Context.

Authors' Addresses

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

Email: marco.tiloca@ri.se

Goeran Selander
Ericsson AB
Farogatan 6
Kista SE-16480 Stockholm
Sweden

Email: goran.selander@ericsson.com

Francesca Palombini
Ericsson AB
Farogatan 6
Kista SE-16480 Stockholm
Sweden

Email: francesca.palombini@ericsson.com