CoRE Working Group
Internet-Draft
Updates: 7252, 7390, 7641 (if approved)
Intended status: Standards Track
Expires: January 7, 2020

M. Tiloca
R. Hoeglund
RISE AB
C. Amsuess

F. Palombini
Ericsson AB

July 06, 2019

Observe Notifications as CoAP Multicast Responses
draft-tiloca-core-observe-multicast-notifications-00

Abstract

The Constrained Application Protocol (CoAP) allows clients to "observe" resources at a server, and receive notifications as unicast responses upon changes of the resource state. In some use cases, such as based on publish-subscribe, it would be convenient for the server to send a single notification to all the clients observing a same target resource. This document defines how a CoAP server sends observe notifications as response messages over multicast, by synchronizing all the observers of a same resource on a same shared Token value. Besides, this document defines how Group OSCORE can be used to protect multicast notifications end-to-end from the CoAP server to the multiple CoAP clients registered as observers.

Table of Contents

1. Introduction

The Constrained Application Protocol (CoAP) [RFC7252] has been extended with a number of mechanisms, including resource Observation [RFC7641]. This enables CoAP clients to register at a CoAP server as "observers" of a resource, and hence being automatically notified with an unsolicited response upon changes of the resource state.

CoAP supports group communication over IP multicast [RFC7390], and [I-D.dijk-core-groupcomm-bis] has been enabling Observe registration requests over multicast, in order for clients to efficiently register as observers of a resource hosted at multiple servers.

However, in a number of use cases, the opposite usage of multicast messages would be also desirable. That is, it would be useful that a server sends observe notifications for a same target resource to multiple observer clients, as responses over IP multicast.

For instance, in CoAP publish-subscribe [I-D.ietf-core-coap-pubsub], multiple clients can subscribe to a topic, by observing the related resource hosted at the responsible broker. When a new value is published on that topic, it would be convenient for the broker to send a single multicast notification at once, to all the subscriber clients observing that topic.

A different use case concerns clients observing a registration resource at the CoRE Resource Directory [I-D.ietf-core-resource-directory]. For example, multiple clients can benefit of observation for discovering (to-be-created) OSCORE groups [I-D.ietf-core-oscore-groupcomm] and retrieving updated information to join them through their respective Group Manager [I-D.tiloca-core-oscore-discovery].

More in general, multicast notifications would be beneficial whenever several CoAP clients observe a same target resource at a CoAP server, and can be all notified at once by means of a single response message. However, CoAP does not currently define response messages over IP multicast. This specification fills this gap and provides the following twofold contribution.

First, it defines a method to deliver Observe notifications as CoAP responses over IP multicast. The proposed method relies on the server managing the Token space for multicast notifications, by providing all the observers of a target resource with the same Token value to bind to their own observation. That Token value is used in every multicast notification for that target resource. This is achieved by introducing a new CoAP option used in the notification response to the original registration request from each client.

Second, this specification defines how to use Group OSCORE [I-D.ietf-core-oscore-groupcomm] to protect multicast notifications end-to-end between the server and the observer clients. This is achieved by introducing a second new CoAP option used in the notification response to the original registration request from each client. The option specifies parameter values that the server uses to secure every multicast notification for the target resource by

using Group OSCORE. This provides a secure binding between each of such notifications and the observation of each of the clients.

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 terms and concepts described in CoAP [RFC7252], group communication for CoAP [RFC7390][I-D.dijk-core-groupcomm-bis], Observe [RFC7641], CBOR [RFC7049], OSCORE [I-D.ietf-core-object-security], and Group OSCORE [I-D.ietf-core-oscore-groupcomm].

2. The Override-Token Option

The Override-Token option defined in this section has the properties summarized in Figure 1, which extends Table 4 of [RFC7252].

Since the option is not Safe-to-Forward, the column "N" is filled with a dash. The Override-Token option contains a Token value.

No.	C	U	N	R	Name	Format	Length	Default
TBD1	X	x	-		Override-Token	opaque	1-8	(none)

C=Critical, U=Unsafe, N=NoCacheKey, R=Repeatable

Figure 1: The Override-Token Option.

This document specifically defines how this option is used to support Observe notifications over IP multicast (see Section 4).

If a server provides multicast notifications for a target resource, the server includes the Override-Token option in the unicast notification response sent as the first reply to a registration request from each client to that resource, i.e. a GET request with the Observe option set to 0. The server indicates a same immutable Token value T in the Override-Token option of each of these first replies. In any other circumstance, this option is not included.

The server will use that same Token value T when sending multicast notifications to registered clients observing that resource. This

ensures that every multicast notification for that resource is expected on the same Token value T by each observing client.

The Override-Token option is of class U for OSCORE [I-D.ietf-core-object-security][I-D.ietf-core-oscore-groupcomm], since intermediaries may be present, and may replace it with a new instance (see Section 6).

3. The Override-AAD Option

The Override-AAD option defined in this section has the properties summarized in Figure 2, which extends Table 4 of [RFC7252].

No.	C	U	N	R	Name	Format	Length	Default
TBD2	X				Override-AAD	(*)	3-255	(none)

C=Critical, U=Unsafe, N=NoCacheKey, R=Repeatable
(*) See below.

Figure 2: The Override-AAD Option

If a server that provides multicast notifications for a target resource protects them with Group OSCORE [I-D.ietf-core-oscore-groupcomm], the server includes the Override-AAD option in the unicast notification response sent as the first reply to a registration request from each client to that resource, i.e. a GET request with the Observe option set to 0. In any other circumstance, this option is not included.

The Override-AAD option contains a CBOR array [RFC7049] composed of the two following elements.

- The first element is a CBOR byte string, which encodes the Sender ID of the server in the OSCORE group.

- The second element is a CBOR integer, which encodes a particular value SN of the Sender Sequence Number of the server in the OSCORE group.

The server uses this same immutable pair to build the two OSCORE 'external_aad' (see Section 5.4 of [I-D.ietf-core-object-security] and Sections 3.1 and 3.2 of [I-D.ietf-core-oscore-groupcomm]), when encrypting and countersigning every multicast notification for the observed resource using Group OSCORE [I-D.ietf-core-oscore-groupcomm].

This ensures that every multicast notification for a same observed resource is securely bound to the first unicast notification sent to each client observing that resource.

The Override-AAD option is of class E for OSCORE [I-D.ietf-core-object-security][I-D.ietf-core-oscore-groupcomm].

4. Resource Observation

Clients interested in receiving multicast notifications from a server have to first register their interest, as described in Section 4.1. This registration is performed over unicast, i.e. comprising both the observation request and the first notification response.

Upon a change of the state of the target resource, the server sends a multicast notification, i.e. a single CoAP response over Multicast IP intended to all the clients in the list of observers of that resource, as described in Section 4.2. Multicast notifications MUST be non-confirmable.

Interested clients need to know the IP multicast address and UDP port number where the server sends multicast notifications for the target resource(s). To this end, a possible approach may rely on the CoRE Resource Directory (RD) and the RD-Groups usage pattern (see Appendix A of [I-D.ietf-core-resource-directory]). In particular, application groups may be registered to the RD, as composed of the resource(s) for which a server provides multicast notifications, and specifying the used IP multicast address and UDP port number. Further details on how clients retrieve this information are out of the scope of this specification.

The server MUST NOT send multicast notifications to unmanaged IP multicast addresses, such as All CoAP Nodes (see Section 12.8 of [RFC7252]).

4.1. Client Registration

The registration process occurs according to the following steps.

1. A client sends an observation request to the server as described in [RFC7641], i.e. a GET request with an Observe option set to 0 (register).

2. If the list of observers for the target resource has just changed from empty to including one observer, the server selects a currently available value T from its Token space and exclusively assigns it as Token value to the list of observers of the target resource (see also related considerations in Section 5). That

is, from then on, the server MUST use T as its own local Token value associated to that observation, with respect to the (next hop towards the) client.

3. The server adds the client to the list of observers of the target resource.

4. The server sends a unicast response notification to the client as described in [RFC7641], i.e. a 2.05 (Content) or 2.03 (Valid) response with an Observe option including a sequence number. Additionally, the server includes an Override-Token option defined in Section 2, which MUST contain the value T. Note that every client further added to the same non-empty list of observers of that target resource receives a notification response to its registration request with the same value T included in the Override-Token option.

5. Upon receiving the unicast notification response to the observation request, the client retrieves the Override-Token option and the conveyed value T. The client interprets this response as if the server: i) has successfully added an entry with the client endpoint and Token value T to the list of observers of the target resource; and ii) will notify changes to the state of the target resource, by means of multicast notifications with Token value T. The client MAY adopt a policy for re-registering its interest for observation, if the Override-Token option includes a Token value already in use with that server. Clients MUST treat as non valid and silently discard responses that include an Override-Token option but do not include also an Observe option.

6. From then on, the client MUST be able to receive, accept and process multicast notifications about the state of the target resource from the server. To this end, the client is required to know in advance the IP multicast address and port number where the server will send multicast notifications to. Also, from then on, the client MUST use T as its own local Token value associated to that observation, with respect to the (next hop towards the) server. The particular way to achieve this is implementation specific.

4.2. Multicast Notifications

Upon a change of the status of the target resource, the server sends a multicast notification intended to all the clients in the list of observers of that resource. In particular, a multicast notification MUST include an Observe option, as specified in [RFC7641].

Compared to notifications as described in [RFC7641], the following two differences apply for a multicast notification.

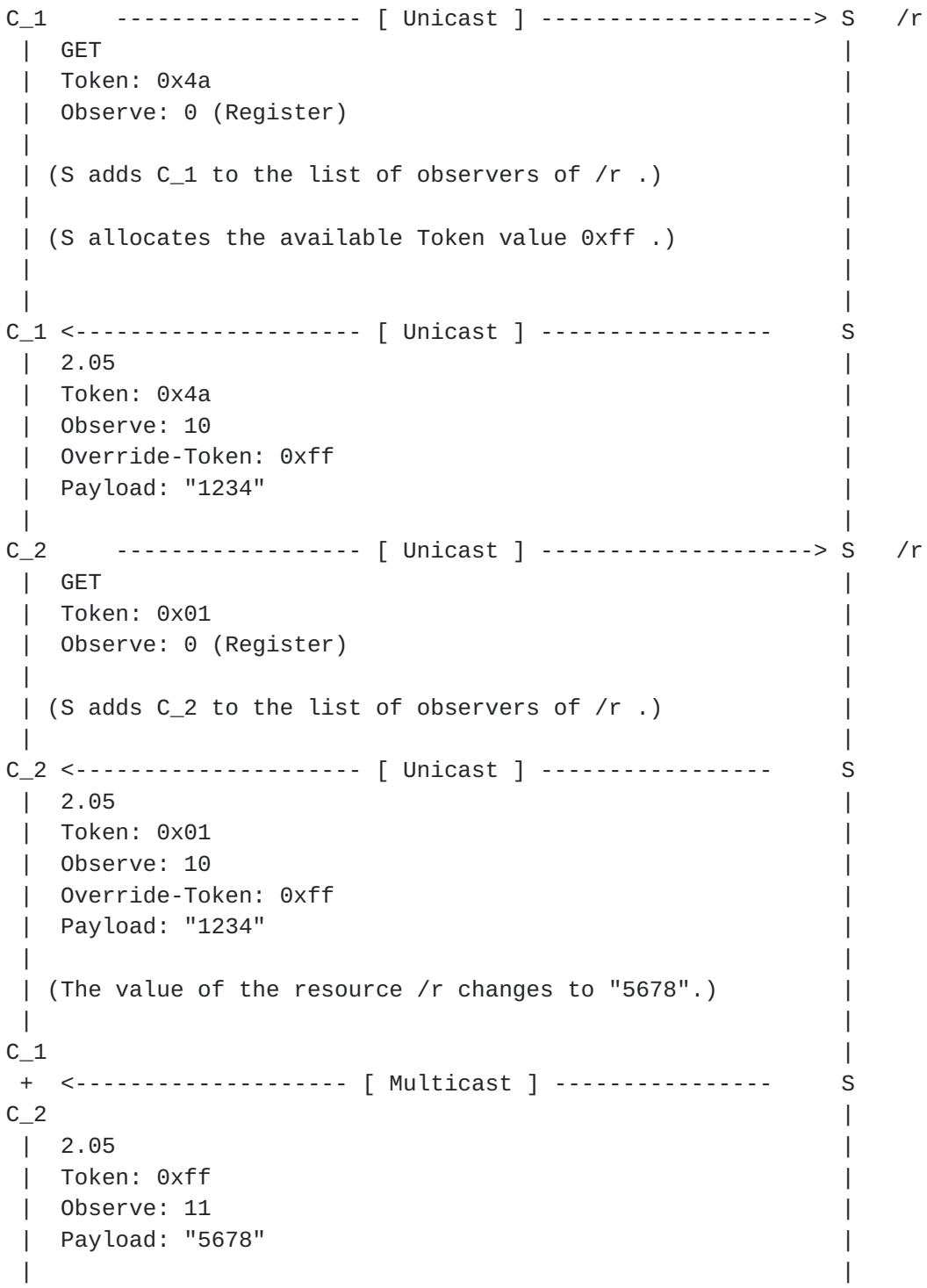
- It is sent as a single CoAP response over Multicast IP.
- It has Token value T, as indicated to every interested client in the Override-Token option of the notification response to its observation request to the target resource (see Section 4.1).

That is, every multicast notification for a target resource is not bound to the different original observation requests, but rather to the whole set of clients currently in the list of observers of that resource.

4.3. Example

The following example refers to two clients C_1 and C_2 that register to observe a resource /r at a Server S.

Before the following exchanges occur, no clients are observing the resource /r , which has value "1234".



5. Token Values for Multicast Notifications

The Token space for multicast notifications is shared by all the clients that have registered to observe resources at a server. As described in Section 4, the server aligns all the clients observing a

same resource to consider a same Token value, which is then used in every multicast notification sent for that resource.

This specification updates [RFC7252] by defining the Token values in Figure 3 as intended for multicast notifications.

A server supporting the Override-Token option and Observe multicast notifications MUST use the Token values in Figure 3 for its multicast notifications and as content of the Override-Token option. A server MUST NOT use the same Token value in multicast notifications for multiple resources currently under observation.

A client supporting the Override-Token option and Observe multicast notifications MUST NOT use the Token values in Figure 3 for its outgoing messages, except when explicitly cancelling the observation, i.e. a GET request to the server with an Observe option set to 1 (see Section 3.6 of [RFC7641]). That GET request has the same Token value used by the server in the multicast notifications for that observation.

This ensures clients to correctly distinguish a multicast notification from a regular (notification) response, as well as to correctly bind the former with the corresponding observation, while the latter with the corresponding original request.

Token size (Bytes)	Token value range	Range size
1	[0xf0 , 0xff]	16
2	[0xffc0 , 0xffff]	64
3	[0xffff00 , 0xffffff]	256
4	[0xffffbff , 0xffffffff]	1024
5	[0xfffffff7ff , 0xffffffffff]	2048
6	[0xffffffffefff , 0xffffffffffff]	4096
7	[0xfffffffffdfff , 0xffffffffffffff]	8192
8	[0xfffffffffffbfff , 0xffffffffffffffff]	16384

Figure 3: Range of Token values for multicast notifications.

6. Intermediaries

In case intermediaries such as CoAP proxies are involved, the same approach described in Section 4 is used independently on both the client- and server-side of each proxy. Care must be taken to only use IP multicast addresses that have all the same meaning on all interfaces of the involved hops.

Upon receiving the unicast response to an original observation request, a proxy on the path between the server and a client performs the following actions.

- The proxy retrieves the Token value T from the Override-Token option. From then on, the proxy MUST use T as its own local Token value associated to that observation, with respect to the next hop towards the server.

- The proxy MAY remove the original Override-Token option. In such a case, the proxy MUST include a new Override-Token option. The newly included Override-Token option specifies a Token value T' (which may be equal to T), consistently with the rules defined in Section 5. From then on, the proxy uses T' as its own local Token value associated to that observation, with respect to the next hop towards the clients. Otherwise, if the proxy does not remove the Override-Token option, the proxy uses T as its own local Token value associated to that observation, with respect to the next hop towards the clients.

The process described above starts at the server and continues until the clients are eventually reached. Even in the presence of intermediaries, this ensures general conflict-free synchronization of Token values at each hop on the path from the server to the clients.

7. Protection of Multicast Notifications with Group OSCORE

A server can protect multicast notifications by using Group OSCORE [I-D.ietf-core-oscore-groupcomm]. In such a case, both the server and the clients interested in receiving multicast notifications from that server have to be members of the same OSCORE group.

Building on the approach suggested in Section 4.1 to discover IP multicast addresses and UDP port numbers, clients may discover the OSCORE group to refer to by using the method in [I-D.tiloca-core-oscore-discovery], also based on the CoRE Resource Directory (RD) [I-D.ietf-core-resource-directory].

Furthermore, both the clients and server may join the OSCORE group by using the approach described in [I-D.ietf-ace-key-groupcomm-oscore]

and based on the ACE framework for Authentication and Authorization in constrained environments [I-D.ietf-ace-oauth-authz].

Further details on how to discover the OSCORE group and join it are out of the scope of this specification.

Alternative security protocols than Group OSCORE, such as OSCORE [I-D.ietf-core-object-security] and/or DTLS [RFC6347], can be used to protect other unicast exchanges between the server and each client, including the original client registration described in Section 4.1.

7.1. Secure Binding of Multicast Notifications

When using Group OSCORE to protect multicast notifications, the registration process occurs as described in Section 4.1, with the following additions.

- If the list of observers for the target resource has just changed from empty to including one observer, the server consumes the current value of its own Sender Sequence Number SN in the OSCORE group, and hence updates it to SN* = (SN + 1).

  Note for implementation: a possible way to achieve this is for the server to produce a dummy request addressed to the OSCORE group, and protect it using its own Sender Context of the Group OSCORE Security Context. This dummy request is not actually transmitted, i.e. it does not hit the wire.

- Upon sending the unicast first notification response to a just registered client, the server includes in that response an Override-AAD option defined in Section 3. The option MUST contain the pair ('kid' ; 'piv') encoded as defined in Section 3, where 'kid' is the Sender ID of the server in the OSCORE group, while 'piv' is the previously consumed Sender Sequence Number value SN of the server in the OSCORE group, i.e. (SN* - 1). Note that every client further added to the same non-empty list of observers of that target resource receives a notification response to its registration request including the exact same pair ('kid' ; 'piv') in the Override-AAD option.

- Upon receiving the unicast notification response to the observation request, the client retrieves the Override-AAD option and the conveyed pair ('kid' ; 'piv'). From then on, when verifying multicast notifications as described in Section 6.4 of [I-D.ietf-core-oscore-groupcomm], the client MUST use 'kid' as 'request_kid' and 'piv' as 'request_piv' in the two 'external_aad' for decrypting and verifying every multicast notification from the server for the target resource (see Sections 3.1 and 3.2 of

[I-D.ietf-core-oscore-groupcomm]). The particular way to achieve this is implementation specific. Clients MUST treat as non valid and silently discard responses that include an Override-AAD option but that do not include also both an Override-Token option and an Observe option. A client has to be a current member of the OSCORE group comprising also the server and associated to the target resource, and MUST otherwise silently discard responses that include an Override-AAD option.

Upon sending every multicast notification for the target resource as described in Section 4.2, the server protects it with Group OSCORE. In particular, the process described in Section 6.3 of [I-D.ietf-core-oscore-groupcomm] applies, with the following differences when building the two OSCORE 'external_aad' to encrypt and countersign the multicast notification (see Sections 3.1 and 3.2 of [I-D.ietf-core-oscore-groupcomm]).

- The 'request_kid' contains the 'kid' value that the server specifies to clients as first element in the Override-AAD option, when replying to the their registration request.

- The 'request_piv' contains the 'piv' value that the server specifies to clients as second element in the Override-AAD option, when replying to their registration request.

7.2. Example

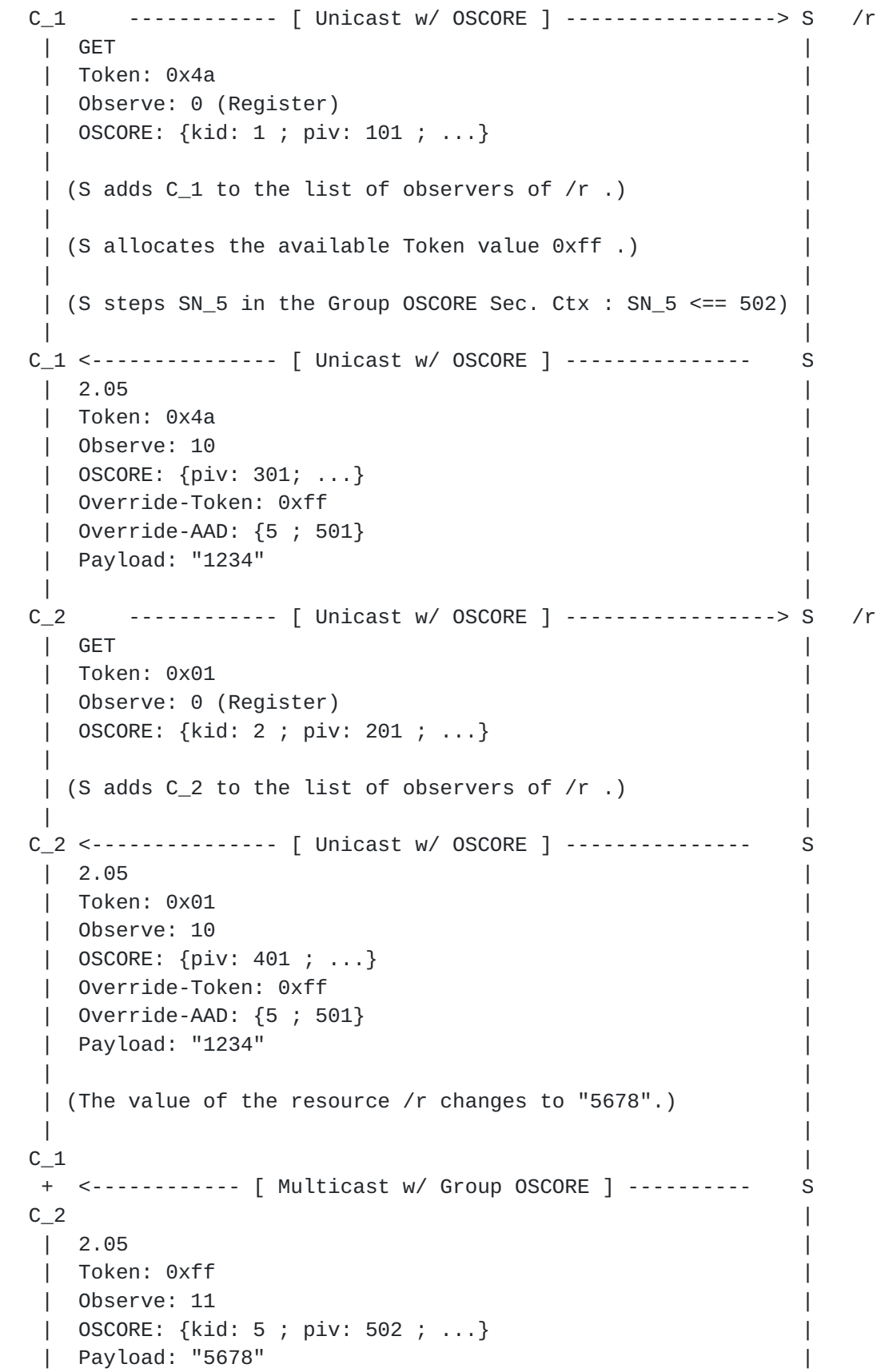
The following example refers to two clients C_1 and C_2 that register to observe a resource /r at a Server S. Pairwise communication over unicast are protected with OSCORE, while S protects multicast notifications with Group OSCORE.

Before the following exchanges occur, no clients are observing the resource /r , which has value "1234". In addition:

- C_1 and S have a pairwise OSCORE Security Context. In particular, C_1 has 'kid' = 1 as Sender ID, and SN_1 = 101 as Sequence Number. Also, S has 'kid' = 3 as Sender ID, and SN_3 = 301 as Sequence Number.

- C_2 and S have a pairwise OSCORE Security Context. In particular, C_2 has 'kid' = 2 as Sender ID, and SN_2 = 201 as Sequence Number. Also, S has 'kid' = 4 as Sender ID, and SN_4 = 401 as Sequence Number.

- C_1, C_2 and S are members of an OSCORE group with 'kid_context' = "feedca57ab2e" as Group ID. In the OSCORE group, S has 'kid' = 5 as Sender ID, and SN_5 = 501 as Sequence Number.


C_1 ---------- [Unicast w/ OSCORE] ----------> S /r
GET
Token: 0x4a
Observe: 0 (Register)
OSCORE: {kid: 1 ; piv: 101 ; ...}
(S adds C_1 to the list of observers of /r .)
(S allocates the available Token value 0xff .)
(S steps SN_5 in the Group OSCORE Sec. Ctx : SN_5 <== 502)
C_1 <---------- [Unicast w/ OSCORE] ---------- S
2.05
Token: 0x4a
Observe: 10
OSCORE: {piv: 301; ...}
Override-Token: 0xff
Override-AAD: {5 ; 501}
Payload: "1234"
C_2 ---------- [Unicast w/ OSCORE] ----------> S /r
GET
Token: 0x01
Observe: 0 (Register)
OSCORE: {kid: 2 ; piv: 201 ; ...}
(S adds C_2 to the list of observers of /r .)
C_2 <---------- [Unicast w/ OSCORE] ---------- S
2.05
Token: 0x01
Observe: 10
OSCORE: {piv: 401 ; ...}
Override-Token: 0xff
Override-AAD: {5 ; 501}
Payload: "1234"
(The value of the resource /r changes to "5678".)
C_1
+ <---------- [Multicast w/ Group OSCORE] ---------- S
C_2
2.05
Token: 0xff
Observe: 11
OSCORE: {kid: 5 ; piv: 502 ; ...}
Payload: "5678"

The two external_aad used to encrypt and countersign the multicast notification above have 'req_kid' = 5 and 'req_iv' = 501, as indicated in the Override-AAD option to the two clients. Thus, the two clients can build the two same external_aad for decrypting and verifying this multicast notification and the following ones.

8. Security Considerations

The same security considerations from [RFC7252][RFC7390][RFC7641][I-D.dijk-core-groupcomm-bis][I-D.ietf-core-object-security][I-D.ietf-core-oscore-groupcomm] hold for this document.

The Override-Token option is of class U for OSCORE, hence intermediaries and on-path active adversaries are able to modify its value. This yields the same effects of altering the Token value of CoAP messages.

If multicast notifications are protected using Group OSCORE, the original registration requests and related unicast notification responses MUST also be secured. This prevents on-path active adversaries from altering the Override-AAD option, and thus ensures secure binding between every multicast notification for a same observed resource and the first notification response sent to each client observing that resource.

To this end, clients and servers MUST use OSCORE or Group OSCORE, for which the Override-AAD option is of class E and would thus be hidden also from intermediaries such as CoAP proxies. This ensures that the secure binding above is enforced end-to-end between the server and each observing client.

9. IANA Considerations

This document has the following actions for IANA.

9.1. CoAP Option Numbers Registry

IANA is asked to enter the following option numbers to the "CoAP Option Numbers" registry defined in [RFC7252] within the "CoRE Parameters" registry.

Number	Name	Reference
TBD1	Override-Token	[[this document]]
TBD2	Override-AAD	[[this document]]

10. References

10.1. Normative References

[I-D.dijk-core-groupcomm-bis]
Dijk, E., Wang, C., and M. Tiloca, "Group Communication for the Constrained Application Protocol (CoAP)", draft-dijk-core-groupcomm-bis-00 (work in progress), March 2019.

[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-16 (work in progress), March 2019.

[I-D.ietf-core-oscore-groupcomm]
Tiloca, M., Selander, G., Palombini, F., and J. Park, "Group OSCORE - Secure Group Communication for CoAP", draft-ietf-core-oscore-groupcomm-05 (work in progress), July 2019.

[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>.

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

[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>.

[RFC7641] Hartke, K., "Observing Resources in the Constrained Application Protocol (CoAP)", RFC 7641, DOI 10.17487/RFC7641, September 2015, <https://www.rfc-editor.org/info/rfc7641>.

[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.ietf-ace-key-groupcomm-oscore]
Tiloca, M., Park, J., and F. Palombini, "Key Management for OSCORE Groups in ACE", draft-ietf-ace-key-groupcomm-oscore-02 (work in progress), July 2019.

[I-D.ietf-ace-oauth-authz]
Seitz, L., Selander, G., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "Authentication and Authorization for Constrained Environments (ACE) using the OAuth 2.0 Framework (ACE-OAuth)", draft-ietf-ace-oauth-authz-24 (work in progress), March 2019.

[I-D.ietf-core-coap-pubsub]
Koster, M., Keranen, A., and J. Jimenez, "Publish-Subscribe Broker for the Constrained Application Protocol (CoAP)", draft-ietf-core-coap-pubsub-08 (work in progress), March 2019.

[I-D.ietf-core-resource-directory]
Shelby, Z., Koster, M., Bormann, C., Stok, P., and C. Amsuess, "CoRE Resource Directory", draft-ietf-core-resource-directory-22 (work in progress), July 2019.

[I-D.tiloca-core-oscore-discovery]
Tiloca, M., Amsuess, C., and P. Stok, "Discovery of OSCORE Groups with the CoRE Resource Directory", draft-tiloca-core-oscore-discovery-03 (work in progress), July 2019.

[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>.

[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>.

Acknowledgments

The authors sincerely thank John Mattsson, Ludwig Seitz and Goeran Selander for their comments and feedback.

The work on this document has been partly supported by VINNOVA and the Celtic-Next project CRITISEC.

Authors' Addresses

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

Email: marco.tiloca@ri.se

Rikard Hoeglund
RISE AB
Isafjordsgatan 22
Kista SE-16440 Stockholm
Sweden

Email: rikard.hoglund@ri.se

Christian Amsuess
Hollandstr. 12/4
Vienna 1020
Austria

Email: christian@amsuess.com

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

Email: francesca.palombini@ericsson.com