CoRE Working Group
Internet-Draft
Intended status: Standards Track
Expires: 23 April 2026

R. H旦glund
M. Tiloca
RISE AB

20 October 2025

Identifier Update for OSCORE
draft-ietf-core-oscore-id-update-05

Abstract

Two peers that communicate with the CoAP protocol can use the Object Security for Constrained RESTful Environments (OSCORE) protocol to protect their message exchanges end-to-end. To this end, the two peers share an OSCORE Security Context and a number of related identifiers. In particular, each of the two peers stores a Sender ID that identifies its own Sender Context within the Security Context, and a Recipient ID that identifies the Recipient Context associated with the other peer within the same Security Context. These identifiers are sent in plaintext within OSCORE-protected messages. Hence, they can be used to correlate messages exchanged between peers and track those peers, with consequent privacy implications. This document defines an OSCORE ID update procedure that two peers can use to update their OSCORE identifiers. This procedure can be run stand-alone or seamlessly integrated in an execution of the Key Update for OSCORE (KUDOS) procedure.

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-core-oscore-id-update/.

Discussion of this document takes place on the Constrained RESTful Environments (core) Working Group mailing list (mailto:core@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/core/. Subscribe at https://www.ietf.org/mailman/listinfo/core/.

Source for this draft and an issue tracker can be found at https://github.com/core-wg/oscore-id-update.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Copyright Notice

Table of Contents

1. Introduction

When using the CoAP protocol [RFC7252], two peers can use the Object Security for Constrained RESTful Environments (OSCORE) protocol to protect their message exchanges end-to-end. To this end, the two peers share an OSCORE Security Context and a number of related identifiers.

As part of the shared Security Context, each peer stores one Sender Context identified by a Sender ID and used to protect its outgoing messages. Also, it stores a Recipient Context identified by a Recipient ID and used to unprotect the incoming messages from the other peer. That is, one's peer Sender ID (Recipient ID) is equal to the other peer's Recipient ID (Sender ID).

When receiving an OSCORE-protected message, the recipient peer uses its Recipient ID conveyed within the message or otherwise implied, in order to retrieve the correct Security Context and unprotect the message.

These identifiers are sent in plaintext within OSCORE-protected messages and are immutable throughout the lifetime of a Security Context, even in case the two peers migrate to a different network or simply change their addressing information. Therefore, the identifiers can be used to correlate messages that the two peers exchange at different points in time or through different paths, hence allowing to track them with the consequent privacy implications.

In order to address this issue, this document defines an OSCORE ID update procedure that two peers can use to update their OSCORE Sender and Recipient IDs. For instance, two peers may want to use this procedure before switching to a different network, in order to make it more difficult to understand that their communication is continuing in the new network.

The OSCORE ID update procedure can be run stand-alone or seamlessly integrated in an execution of the Key Update for OSCORE (KUDOS) procedure [I-D.ietf-core-oscore-key-update].

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 related to CoAP [RFC7252], Observe [RFC7641], CBOR [RFC8949], OSCORE [RFC8613], and KUDOS [I-D.ietf-core-oscore-key-update].

2. Update of OSCORE Sender/Recipient IDs

This section defines the procedure that two peers can perform, in order to update the OSCORE Sender/Recipient IDs that they use in their shared OSCORE Security Context.

When performing an update of OSCORE Sender/Recipient IDs, a peer provides its new intended OSCORE Recipient ID to the other peer, by means of the Recipient-ID Option defined in Section 2.3. Hereafter, this document refers to a message including the Recipient-ID Option as an "ID update (request/response) message". Furthermore, a peer uses the Recipient-ID-Ack Option to confirm the chosen Recipient ID of the other peer.

This procedure can be initiated by either peer, i.e., the CoAP client or the CoAP server may start it by sending the first OSCORE ID update message.

Furthermore, this procedure can be executed stand-alone, or instead seamlessly integrated in an execution of the KUDOS procedure for updating OSCORE keying material used in its FS mode (see Section 4 of [I-D.ietf-core-oscore-key-update]) or no-FS mode (see Section 4.5 of [I-D.ietf-core-oscore-key-update]).

* In the former stand-alone case, updating the OSCORE Sender/ Recipient IDs effectively results in updating part of the current OSCORE Security Context.

  That is, both peers derive a new Sender Key, Recipient Key, and Common IV, as defined in Section 3.2 of [RFC8613]. Also, both peers re-initialize the Sender Sequence Number and the Replay Window accordingly, as defined in Section 3.2.2 of [RFC8613]. Since the same Master Secret is preserved, forward secrecy is not achieved.

  As defined in Section 2.4.3, the two peers must take additional actions to ensure a safe execution of the OSCORE ID update procedure.

  The new OSCORE Sender/Recipient IDs MUST NOT be used with the OSCORE Security Context CTX_OLD, and MUST NOT be used with the temporary OSCORE Security Context CTX_TEMP used to protect the first KUDOS message of a KUDOS execution.

A peer MUST NOT initiate an OSCORE ID update procedure with another peer, if it has another such procedure ongoing with that other peer.

Upon receiving a valid, first ID update message, a peer MUST continue the procedure and send a following ID update message, except in the case any of the conditions for failing or aborting the procedure apply (see Section 2.2}).

2.1. Workflow of the ID Update Procedure

This section describes the workflow of the OSCORE ID Update procedure in detail.

The procedure begins when either peer:

* Sends a message including the Recipient-ID Option, or

* Receives such a message from the other peer.

During the procedure a peer decides on a value of Recipient ID to offer to the other peer and use as value of the Recipient-ID Option, and continues offering that value until the procedure is completed.

Once the procedure has started a peer shall follow the instructions below:

Sending the Next Message

* The first messages sent using CTX_A, the current shared OSCORE Security Context, after the procedure has started must include the Recipient-ID Option, if this peer hasn’t offered its Recipient ID already.

  - Note that this also informs the other peer of support for the ID update procedure.

Acknowledgment

* If a peer has received a valid message from the other peer including the Recipient-ID Option, it must include the Recipient-ID-Ack Option in subsequent messages.

* The value of Recipient-ID-Ack Option, if used, should be the Recipient ID received from the other peer.

Sending Subsequent Messages

A peer must send one message with the Recipient ID Option according to the following:

* A local timer, REPEAT_TIMER, should be maintained during the procedure. It first starts when the procedure starts. It is RECOMMENDED that the initial time of REPEAT_TIMER is equal to MAX_TRANSMIT_WAIT (see Section 4.8.2 of [RFC7252]).

  - If the timeout expires, the next sent message must include the Recipient ID option and, if applicable, the Recipient-ID-Ack Option with the last received Recipient ID. When that message is sent the timer REPEAT_TIMER restarts.

2.1.1. Procedure Completion

The procedure concludes under one of the following conditions:

Successful Confirmation

The procedure succeeds if a peer has received and successfully verified at least three message from the other peer containing the Recipient-ID-Ack Option, and sent at least two messages containing the Recipient-ID-Ack Option. At this point:

* It is safe to delete CTX_A. This does not mean that CTX_A has to be deleted at this point.

* CTX_B is now considered valid and can be used (e.g., following network migration).

Failure

During the procedure a timer, ENDING_TIMER, is maintained and started when the procedure starts. The initial time of ENDING_TIMER should be at least 3 times bigger than the initial time of REPEAT_TIMER. If the ENDING_TIMER expires, and the procedure times out without confirmation:

* The offered Recipient ID must be discarded and added to the list of IDs to prevent reuse.

2.2. Failure of the ID Update Procedure

The following section describes cases where the OSCORE ID update procedure fails, or must to be aborted by one of the peers.

Upon receiving a valid first ID update message, a peer MUST abort the ID update procedure, in the following case:

* The received ID update message is not a KUDOS message (i.e., the OSCORE ID update procedure is being performed stand-alone) and the peer has no eligible Recipient ID to offer (see Section 2.4.3).

Upon receiving a valid ID update message, a peer MUST abort the ID update procedure, in the following case:

* The received ID update message contains a Recipient-ID option with a length that exceeds the maximum length of OSCORE Sender/ Recipient IDs for the AEAD algorithm in use for the OSCORE Security Context shared between the peers. This is the case when the length of the Recipient-ID option exceeds the length of the AEAD nonce minus 6 (see Section 3.3 of [RFC8613]).

If, after receiving an ID update message as CoAP request, a peer aborts the ID update procedure, the peer MUST also reply to the received ID update request message with a protected 5.03 (Service Unavailable) error response. The error response MUST NOT include the Recipient-ID Option, and its diagnostic payload MAY provide additional information. When receiving the error response, the peer terminates the OSCORE IDs procedure as failed.

When the OSCORE ID update procedure is integrated into the execution of the KUDOS procedure, it is possible that the KUDOS procedure succeeds while the OSCORE ID update procedure fails. In such case, the peers continue their communications using the newly derived OSCORE Security Context CTX_NEW obtained from the KUDOS procedure, and still use the old Sender and Recipient IDs. That is, any Recipient IDs conveyed in the exchanged Recipient-ID Options is not considered.

Conversely, the OSCORE ID update procedure may succeed while the KUDOS procedure fails. As long as the peers have exchanged a pair of OSCORE-protected request and response that conveyed their desired new Recipient IDs in the Recipient-ID Option, the peers start using those IDs in their communications.

2.3. The Recipient-ID Option

The Recipient ID-Option defined in this section has the properties summarized in Table 1, which extends Table 4 of [RFC7252]. That is, the option is elective, safe to forward, part of the cache key, and not repeatable.

No.	C	U	N	R	Name	Format	Length	Default
TBD24					Recipient-ID	opaque	any	(none)

Table 1: The Recipient-ID Option. C=Critical, U=Unsafe, N=NoCacheKey, R=Repeatable

Note to RFC Editor: Following the registration of the CoAP Option Number 24, please replace "TBD24" with "24" in the figure above. Then, please delete this paragraph.

The option value can have an arbitrary length, including zero length to indicate intent to use the empty string as Recipient ID. Implementations can limit its length to that of the longest supported Recipient ID.

This document particularly defines how this option is used in messages protected with OSCORE. That is, when the option is included in an outgoing message, the option value specifies the new OSCORE Recipient ID that the sender endpoint intends to use with the other endpoint sharing the OSCORE Security Context.

Therefore, the maximum length of the option value is equal to the maximum length of OSCORE Sender/Recipient IDs. As defined in Section 3.3 of [RFC8613], this is determined by the size of the AEAD nonce of the used AEAD Algorithm in the OSCORE Security Context.

If the length of the Recipient ID included in the Recipient-ID option is zero, the option value SHALL be empty (Option Length = 0).

The Recipient-ID Option is of class E in terms of OSCORE processing (see Section 4.1 of [RFC8613]).

2.4. The Recipient-ID-Ack Option

The Recipient ID-Ack-Option defined in this section has the properties summarized in Table 2, which extends Table 4 of [RFC7252]. That is, the option is elective, safe to forward, part of the cache key, and not repeatable.

No.	C	U	N	R	Name	Format	Length	Default
TBD32					Recipient-ID-Ack	opaque	any	(none)

Table 2: The Recipient-ID-Ack Option. C=Critical, U=Unsafe, N=NoCacheKey, R=Repeatable

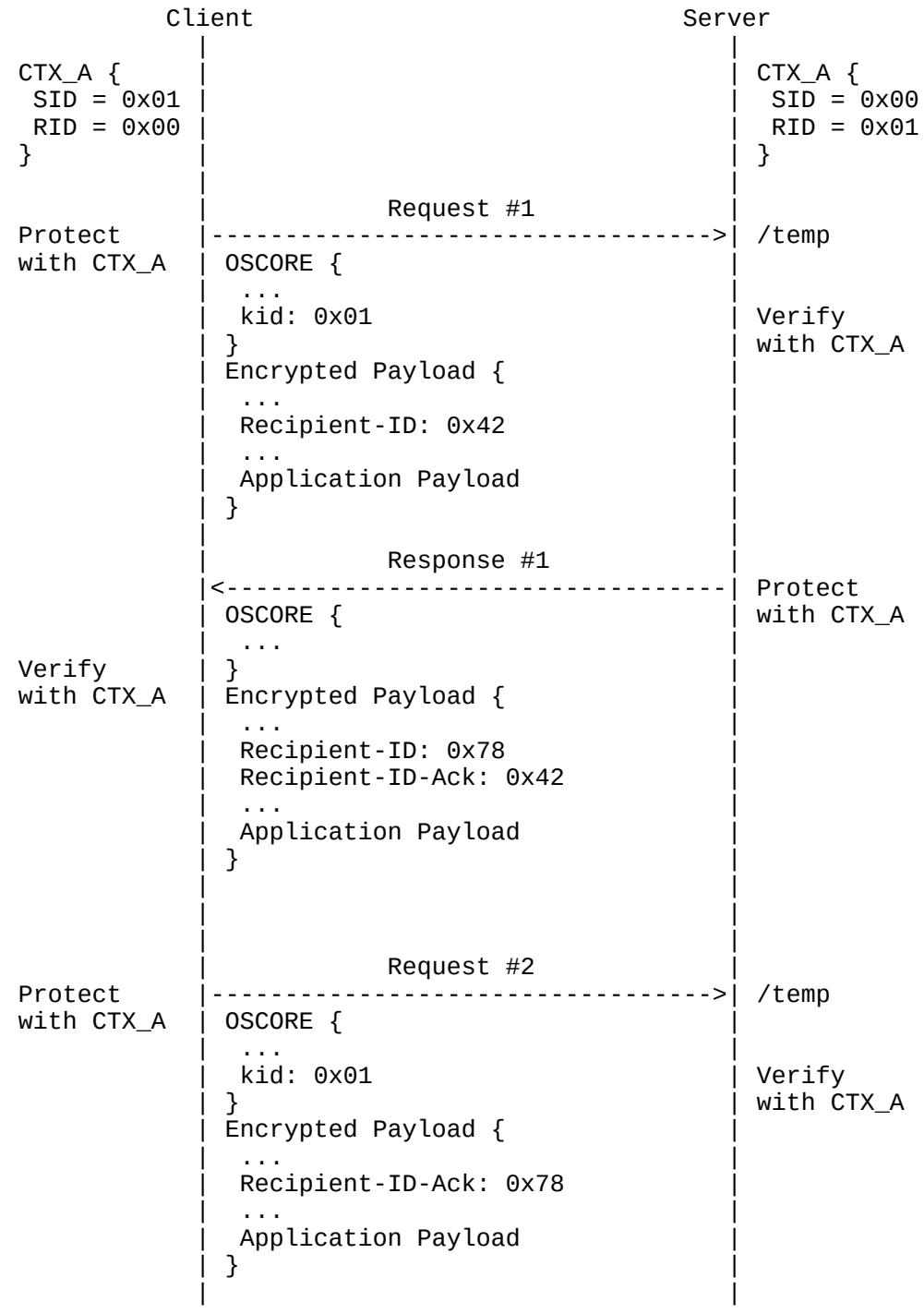
Note to RFC Editor: Following the registration of the CoAP Option Number 32, please replace "TBD32" with "32" in the figure above. Then, please delete this paragraph.

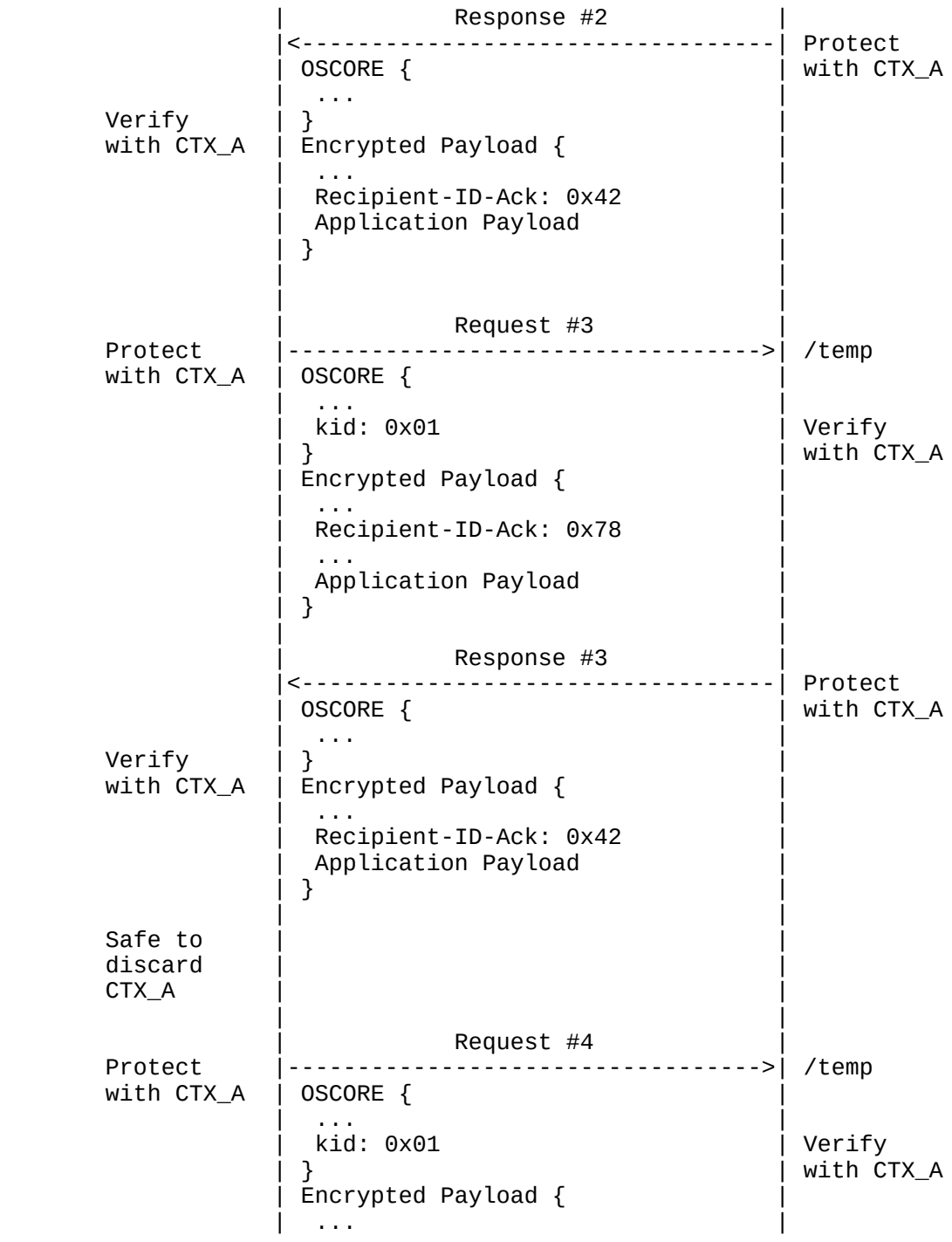
This document particularly defines how this option is used in messages protected with OSCORE. That is, when the option is included in an outgoing message, the option value confirms the new OSCORE Recipient ID that the recipient endpoint has chosen for this sender endpoint.

The Recipient-ID-Ack Option is of class E in terms of OSCORE processing (see Section 4.1 of [RFC8613]).

2.4.1. OSCORE ID Update Procedure Initiated with a Request Message

Figure 1 shows an example of the OSCORE ID update procedure, run stand-alone and initiated by the client sending a request message. On each peer, SID and RID denote the OSCORE Sender ID and Recipient ID of that peer, respectively. An example where the server initiates the procedure is shown in Appendix A.1.


Client
Server
CTX_A {
SID = 0x01
RID = 0x00
}
CTX_A {
SID = 0x00
RID = 0x01
}
Request #1
Protect
with CTX_A
/temp
OSCORE {
...
kid: 0x01
}
Verify
with CTX_A
Encrypted Payload {
...
Recipient-ID: 0x42
...
Application Payload
}
Response #1
Protect
with CTX_A
OSCORE {
...
}
Verify
with CTX_A
Encrypted Payload {
...
Recipient-ID: 0x78
Recipient-ID-Ack: 0x42
...
Application Payload
}
Request #2
Protect
with CTX_A
/temp
OSCORE {
...
kid: 0x01
}
Verify
with CTX_A
Encrypted Payload {
...
Recipient-ID-Ack: 0x78
...
Application Payload
}


Response #2
OSCORE {
...
}
Encrypted Payload {
...
Recipient-ID-Ack: 0x42
Application Payload
}
Verify with CTX_A
Request #3
Protect with CTX_A
/temp
OSCORE {
...
kid: 0x01
}
Encrypted Payload {
...
Recipient-ID-Ack: 0x78
...
Application Payload
}
Verify with CTX_A
Response #3
Protect with CTX_A
OSCORE {
...
}
Encrypted Payload {
...
Recipient-ID-Ack: 0x42
Application Payload
}
Verify with CTX_A
Safe to discard CTX_A
Request #4
Protect with CTX_A
/temp
OSCORE {
...
kid: 0x01
}
Encrypted Payload {
...
Verify with CTX_A

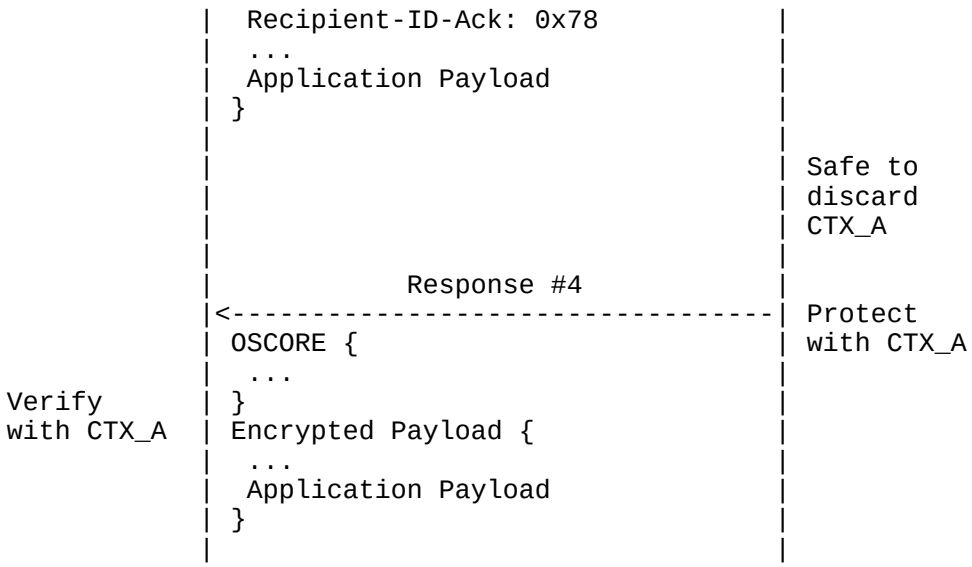


Figure 1: Example of the OSCORE ID update procedure initiated with a request message

Before the OSCORE ID update procedure starts, the client (the server) shares with the server (the client) an OSCORE Security Context CTX_A with Sender ID 0x01 (0x00) and Recipient ID 0x00 (0x01).

When starting the OSCORE ID update procedure, the client determines its new intended OSCORE Recipient ID 0x42. Then, the client prepares a CoAP request targeting an application resource at the server. The request includes the Recipient-ID Option, with value the client's new Recipient ID 0x42.

The client protects the request with CTX_A, i.e., by using the keying material derived from the client's current Sender ID 0x01. The protected request specifies the client's current Sender ID 0x01 in the 'kid' field of the OSCORE Option. After that, the client sends the request to the server as Request #1.

Upon receiving, decrypting, and successfully verifying the OSCORE message Request #1, the server retrieves the value 0x42 from the Recipient-ID Option, and determines its new intended OSCORE Recipient ID 0x78. Then, the server prepares a CoAP response including the Recipient-ID Option, with value the server's new Recipient ID 0x78, and the Recipient-ID-Ack Option, with value the client's offered Recipient ID 0x42.

The server protects the response with CTX_A, i.e., by using the keying material derived from the server's current Sender ID 0x00. After that, the server sends the response to the client.

Next, the client sends the OSCORE message Request #2, which is protected with CTX_A and includes the Recipient-ID-Ack Option, with value the server's offered Recipient ID 0x78.

From this point, following messages exchanges between the peers will include the Recipient-ID-Ack Option. A peer will only stop including that option when it has verified 3 messages from the other peer containing the Recipient-ID-Ack Option.

Upon receiving, decrypting, and successfully verifying the OSCORE message Response #3, the client considers 0x78 and 0x42 as the new Sender ID and Recipient ID to use when deriving CTX_B. Practically, the client can install a new OSCORE Security Context CTX_B where: i) its Sender ID and Recipient ID are 0x78 and 0x42, respectively; ii) the Sender Sequence Number and the Replay Window are re-initialized (see Section 3.2.2 of [RFC8613]); iii) anything else is like in the OSCORE Security Context CTX_A.

Upon receiving, decrypting, and successfully verifying the OSCORE message Request #4, the server considers 0x42 and 0x78 as its new Sender ID and Recipient ID to use for CTX_B. Practically, the server installs a new OSCORE Security Context CTX_A where: i) its Sender ID and Recipient ID are 0x42 and 0x78, respectively; ii) the Sender Sequence Number and the Replay Window are re-initialized (see Section 3.2.2 of [RFC8613]); iii) anything else is like in the OSCORE Security Context CTX_A.

At this point both client and server are in a position to derive CTX_B already, or wait to do it. Regardless they are both able to start using CTX_B, e.g., after network migration.

2.4.2. Establishing New OSCORE Identifiers Ahead of Network Migration

Peers may use the OSCORE ID update procedure to establish new OSCORE IDs in advance of a network change. However, peers SHOULD NOT begin using these new identifiers on the current network prior to network migration. Using a new identifier on the old network, or using the old identifiers on the new network, would allow observers to correlate activity across networks, defeating the unlinkability that updating the OSCORE IDs is intended to provide. To be effective, new identifiers SHOULD only be used for sending OSCORE protected messages once the network migration is completed. Establishing new OSCORE IDs ahead of time ensures that migration can proceed without delay, but care must be taken to ensure that premature use of the identifiers

does not enable linkability.

To accomplish this, the peers execute the ID update procedure as normal, with the following difference: the peers must not begin using the OSCORE Security Context CTX_B until after the network migration has taken place. Thus, both peers will be in the position to derive CTX_B, but will not transition to use it until the first request protected with CTX_B is transmitted in the new network, that is after network migration. Note that peers may want to retain CTX_A to have available for migration back to the old network.

2.4.3. Additional Actions for Execution

After having experienced a loss of state, a peer MUST NOT participate in a stand-alone OSCORE ID update procedure with another peer, until having performed a full-fledged establishment/renewal of an OSCORE Security Context with the other peer (e.g., by running KUDOS [I-D.ietf-core-oscore-key-update] or the authenticated key establishment protocol EDHOC [RFC9528]).

More precisely, a peer has experienced a loss of state if it cannot access the latest snapshot of the latest OSCORE Security Context CTX_OLD or the whole set of OSCORE Sender/Recipient IDs that have been used with the triplet (Master Secret, Master Salt, ID Context) of CTX_OLD. This can happen, for instance, after a device reboots.

Furthermore, when participating in a stand-alone OSCORE ID update procedure, a peer performs the following additional steps.

* When a peer sends an ID update message, the value of the Recipient-ID Option that the peer specifies as its new intended OSCORE Recipient ID MUST fulfill both the following conditions: it is currently available as Recipient ID to use for the peer (see Section 3.3 of [RFC8613]); and the peer has never used it as Recipient ID with the current triplet (Master Secret, Master Salt, ID Context).

* When receiving an ID update message, the peer MUST abort the procedure if it has already used the identifier specified in the Recipient-ID Option as its own Sender ID with the current triplet (Master Secret, Master Salt, ID Context).

In order to fulfill the conditions above, a peer has to keep track of the OSCORE Sender/Recipient IDs that it has used with the current triplet (Master Secret, Master Salt, ID Context) since the latest update of the OSCORE Master Secret (e.g., performed by running KUDOS).

2.5. Preserving Observations Across ID Updates

When having run the OSCORE ID update procedure stand-alone and starting to use CTX_B, or having run the OSCORE ID update procedure integrated in an execution of KUDOS, the following holds if Observe [RFC7641] is supported, in order to preserve ongoing observations beyond a change of OSCORE identifiers.

* If a peer intends to keep active beyond an update of its Sender ID the observations where it is acting as CoAP client, then the peer:

  - MUST store the value of the 'kid' parameter from the original Observe requests, and retain it for the whole duration of the observations, throughout which the client MUST NOT update the stored value associated with the corresponding Observe registration request; and

  - MUST use the stored value of the 'kid' parameter from the original Observe registration request as value for the 'request_kid' parameter in the external_aad structure (see Section 5.4 of [RFC8613]), when verifying notifications for that observation as per Section 8.4.2 of [RFC8613].

* If a peer is acting as CoAP server in an ongoing observation, then the peer:

  - MUST store the value of the 'kid' parameter from the original Observe registration request, and retain it for the whole duration of the observation, throughout which the peer MUST NOT update the stored value associated with the corresponding Observe registration request; and

  - MUST use the stored value of the 'kid' parameter from the original Observe registration request as value for the 'request_kid' parameter in the external_aad structure (see Section 5.4 of [RFC8613]), when protecting notifications for that observation as per Section 8.3.1 of [RFC8613].

3. Security Considerations

The same security considerations as in [RFC8613] and [I-D.ietf-core-oscore-key-update] hold for this document.

The OSCORE ID update procedure is a mechanism to mitigate the risk of tracking by on-path adversaries. By enabling endpoints to update their identifiers, either in response to specific events or on a regular basis, this approach helps prevent correlations that could otherwise be drawn between OSCORE messages on different networks.

While the ID update procedure helps reduce linkability across networks, the change of IDs alone might not completely prevent adversaries from recognizing traffic patterns that reveal message ordering or frequency. That is, the procedure becomes more effective if combined with the protection and/or change of other information that can help identify endpoints across different networks.

In that spirit, when a peer installs a new OSCORE Security Context as a result of the OSCORE ID update procedure, it re-initializes the Sender Sequence Number to 0. That prevents an adversary from obviously tracking the peer by leveraging the Partial IV of observed messages, since the Partial IV value does not predictably continue from the last known value that was used in the previous network. Building on that, the peer can in fact re-initialize the Sender Sequence Number to a value greater than 0, thus making tracking further difficult.

Likewise, other information such as addressing information, may still be used to track the peers. Thus, it is recommended to combine the usage of the OSCORE ID update procedure also with the following, upon network migration:

* Changing the network address (e.g., intentionally, or due to mobility, or NAT rebinding).

* Changing the link-layer address (e.g., MAC address randomization).

Furthermore, it is recommended that a peer does not start using its newly established OSCORE Sender ID until after network migration, in order to mitigate tracking attempts.

4. IANA Considerations

This document has the following actions for IANA.

Note to RFC Editor: Please replace all occurrences of "[RFC-XXXX]" with the RFC number of this specification and delete this paragraph.

4.1. CoAP Option Numbers Registry

IANA is asked to enter the following entries to the "CoAP Option Numbers" registry within the "Constrained RESTful Environments (CoRE) Parameters" registry group.

Number	Name	Reference
TBD24	Recipient-ID	[RFC-XXXX]
TBD32	Recipient-ID-Ack	[RFC-XXXX]

Table 3: New CoAP Option Numbers

Note to RFC Editor: Following the registration of the CoAP Option Number 24, please replace "TBD24" with "24" in the table above. Then, please delete this paragraph. Note to RFC Editor: Following the registration of the CoAP Option Number 32, please replace "TBD32" with "32" in the table above. Then, please delete this paragraph.

5. References

5.1. Normative References

[I-D.ietf-core-oscore-key-update]
H旦glund, R. and M. Tiloca, "Key Update for OSCORE (KUDOS)", Work in Progress, Internet-Draft, draft-ietf-core-oscore-key-update-11, 7 July 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-core-oscore-key-update-11>.

[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/rfc/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/rfc/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/rfc/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/rfc/rfc8174>.

[RFC8613] Selander, G., Mattsson, J., Palombini, F., and L. Seitz, "Object Security for Constrained RESTful Environments (OSCORE)", RFC 8613, DOI 10.17487/RFC8613, July 2019, <https://www.rfc-editor.org/rfc/rfc8613>.

[RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <https://www.rfc-editor.org/rfc/rfc8949>.

5.2. Informative References

[RFC9528] Selander, G., Preu Mattsson, J., and F. Palombini, "Ephemeral Diffie-Hellman Over COSE (EDHOC)", RFC 9528, DOI 10.17487/RFC9528, March 2024, <https://www.rfc-editor.org/rfc/rfc9528>.

Appendix A. Examples

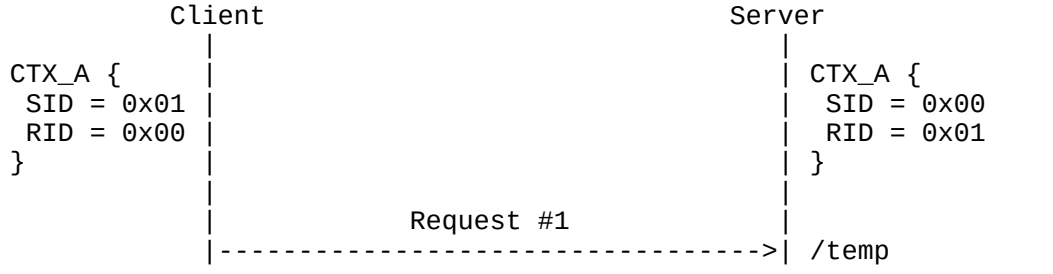
This appendix provides examples where the OSCORE ID update procedure is used. In particular:

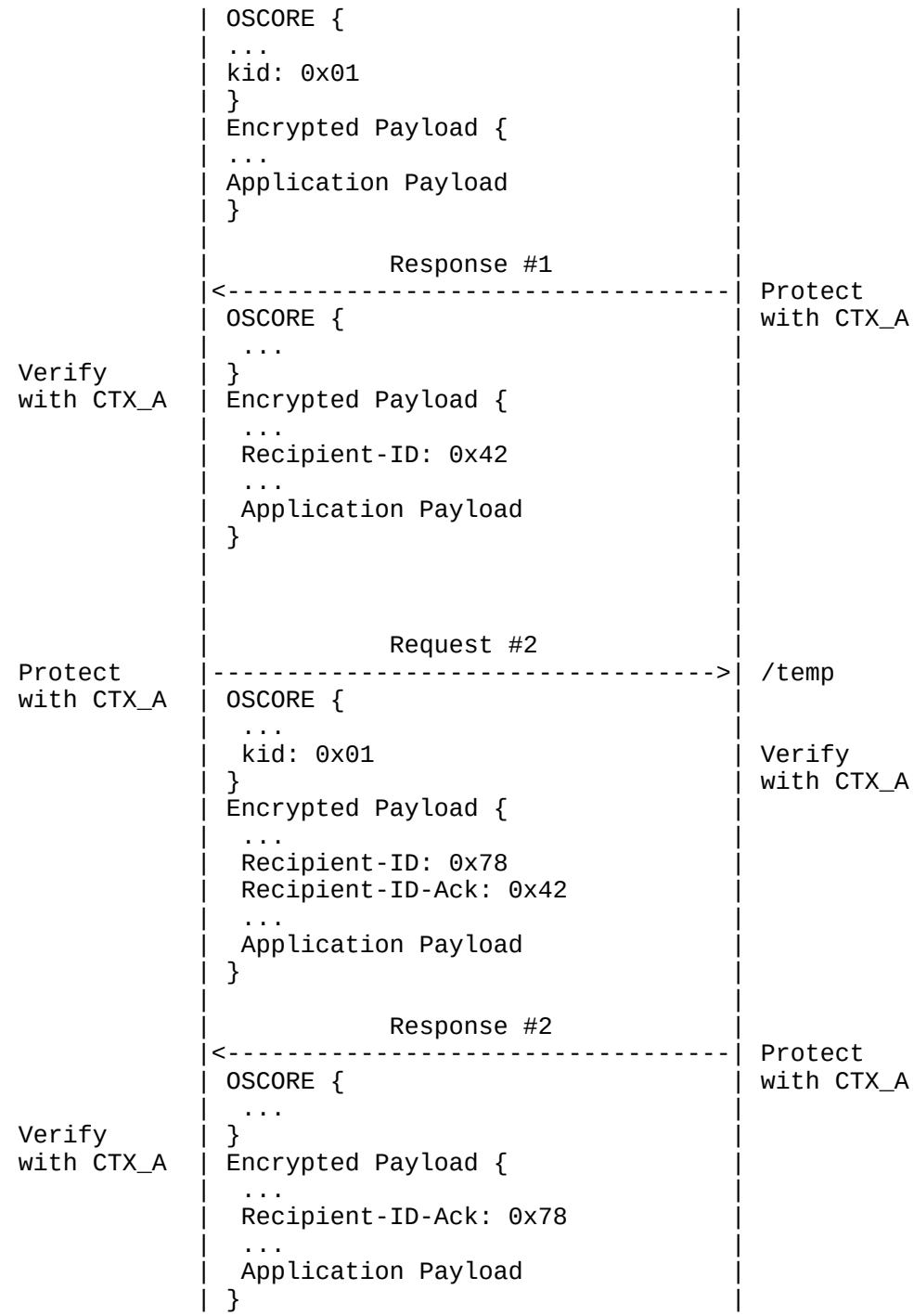
* Appendix A.1 shows an example of the OSCORE ID update procedure initiated by the server sending a response message.

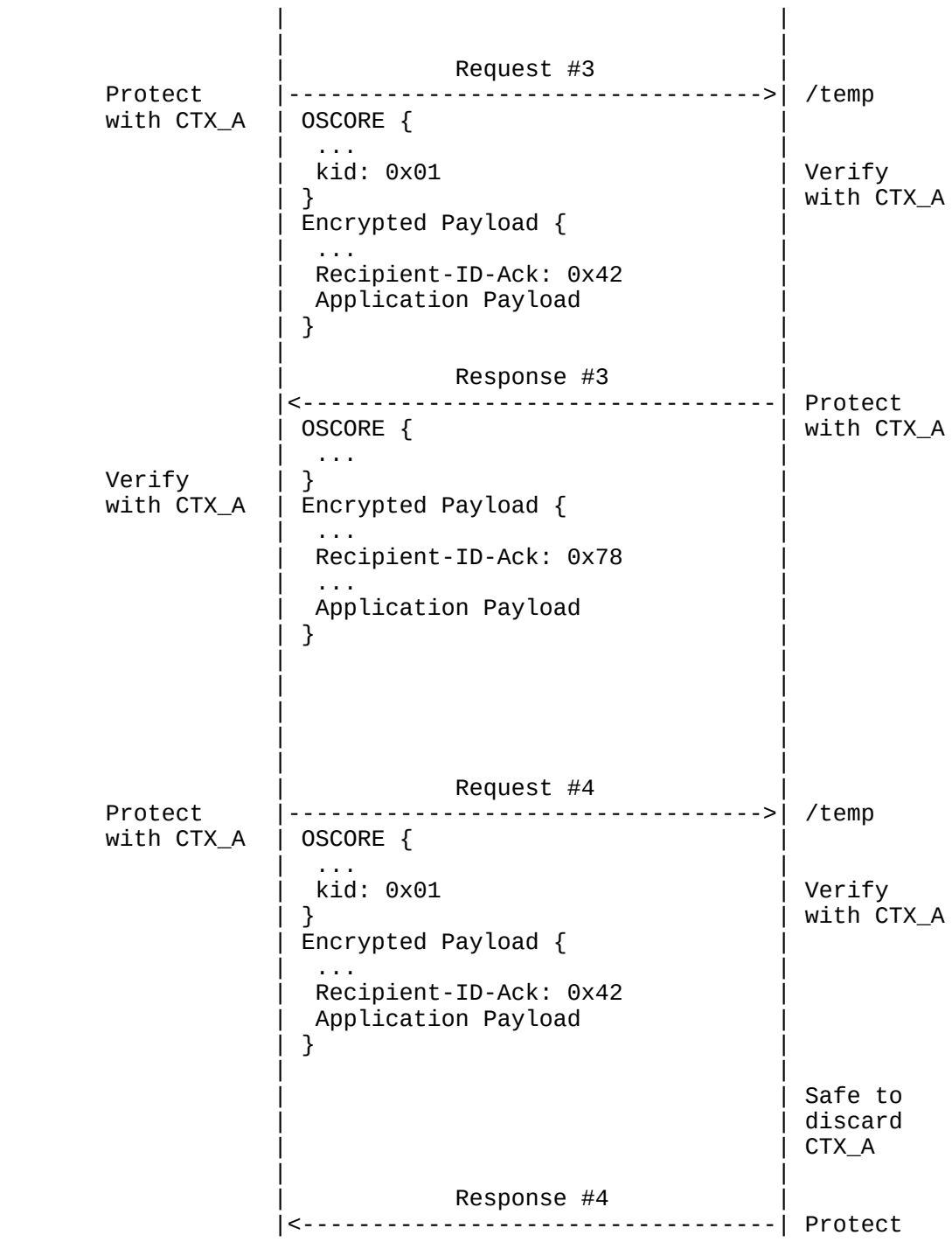
* Appendix A.2 shows an example of the OSCORE ID update procedure initiated by the client sending a request message where the procedure fails to complete.

A.1. OSCORE ID Update Procedure Initiated with a Response Message

Figure 2 shows an example of the OSCORE ID update procedure, run stand-alone and initiated by the server sending a response message. On each peer, SID and RID denote the OSCORE Sender ID and Recipient ID of that peer, respectively. The prerequisites and the actions taken by the peers involved are aligned with what is described in Section 2.4.1, except that it is the server that takes the initiative to perform the OSCORE ID update procedure.




OSCORE {
...
kid: 0x01
}
Encrypted Payload {
...
Application Payload
}
Response #1
Protect
with CTX_A
OSCORE {
...
}
Verify
with CTX_A
Encrypted Payload {
...
Recipient-ID: 0x42
...
Application Payload
}
Request #2
Protect
with CTX_A
/temp
OSCORE {
...
kid: 0x01
Verify
with CTX_A
}
Encrypted Payload {
...
Recipient-ID: 0x78
Recipient-ID-Ack: 0x42
...
Application Payload
}
Response #2
Protect
with CTX_A
OSCORE {
...
Verify
with CTX_A
}
Encrypted Payload {
...
Recipient-ID-Ack: 0x78
...
Application Payload
}


Request #3
Protect with CTX_A
OSCORE {
...
kid: 0x01
}
Encrypted Payload {
...
Recipient-ID-Ack: 0x42
Application Payload
}
/temp
Verify with CTX_A
Response #3
Protect with CTX_A
OSCORE {
...
}
Encrypted Payload {
...
Recipient-ID-Ack: 0x78
...
Application Payload
}
Verify with CTX_A
Request #4
Protect with CTX_A
OSCORE {
...
kid: 0x01
}
Encrypted Payload {
...
Recipient-ID-Ack: 0x42
Application Payload
}
/temp
Verify with CTX_A
Safe to discard CTX_A
Response #4
Protect

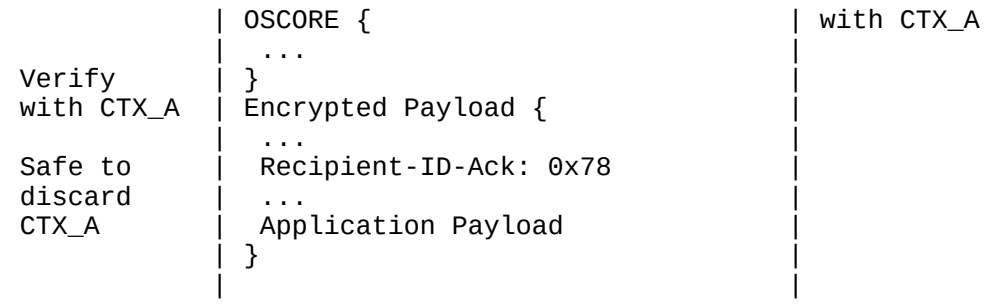
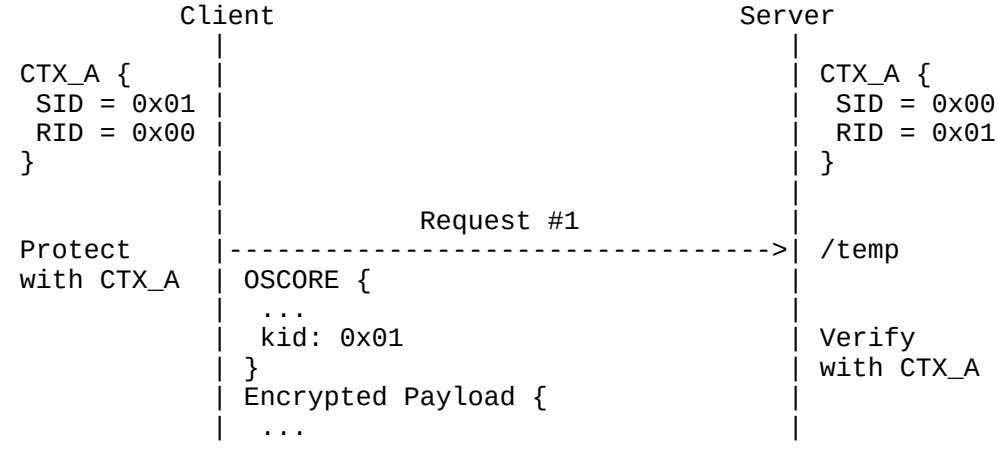


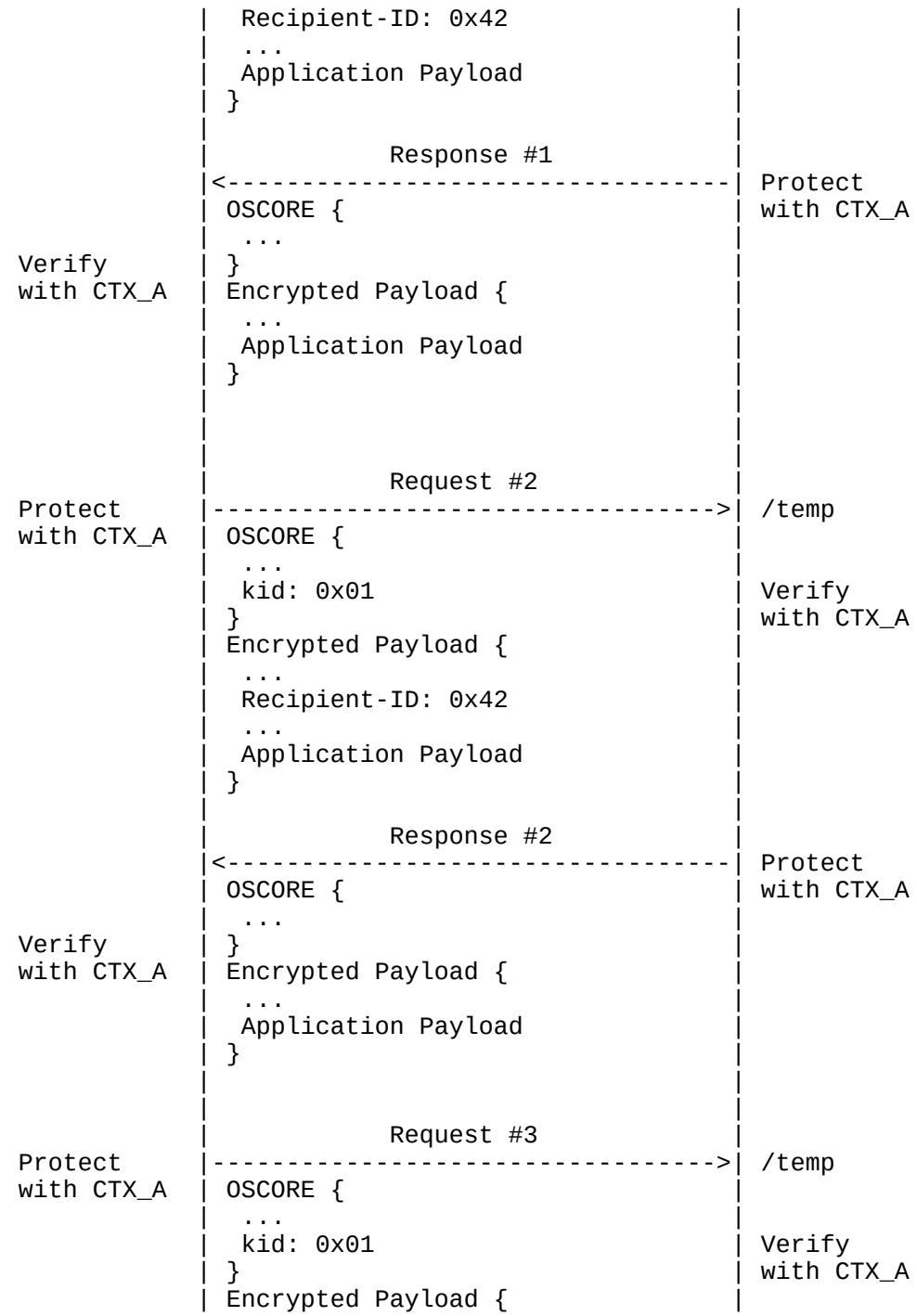
Figure 2: Example of the OSCORE ID update procedure initiated with a response message

A.2. Failure of the OSCORE ID Update Procedure Initiated with a Request Message

Figure 3 shows an example of the OSCORE ID update procedure, run stand-alone and initiated by the client sending a request message where the procedure fails to complete due to the server not including the Recipient-ID-Ack option or the Recipient-ID in its response messages. On each peer, SID and RID denote the OSCORE Sender ID and Recipient ID of that peer, respectively. This example assumes that the value of the REPEAT_TIMER on the client is such that it expires between each request the client sends.

The client repeatedly tries sending requests to the client including the Recipient-ID option, but does not receive acknowledgments in the form of responses containing the Response-ID-Ack option from the server. Thus the client eventually reaches the expiration of its ENDING_TIMER, aborts the OSCORE ID update procedure, and proceeds to continue communication with normal OSCORE messages.




Recipient-ID: 0x42
...
Application Payload
}
Response #1
Protect
with CTX_A
OSCORE {
...
}
Verify
with CTX_A
Encrypted Payload {
...
Application Payload
}
Request #2
Protect
with CTX_A
/temp
OSCORE {
...
kid: 0x01
}
Verify
with CTX_A
Encrypted Payload {
...
Recipient-ID: 0x42
...
Application Payload
}
Response #2
Protect
with CTX_A
OSCORE {
...
}
Verify
with CTX_A
Encrypted Payload {
...
Application Payload
}
Request #3
Protect
with CTX_A
/temp
OSCORE {
...
kid: 0x01
}
Verify
with CTX_A
Encrypted Payload {

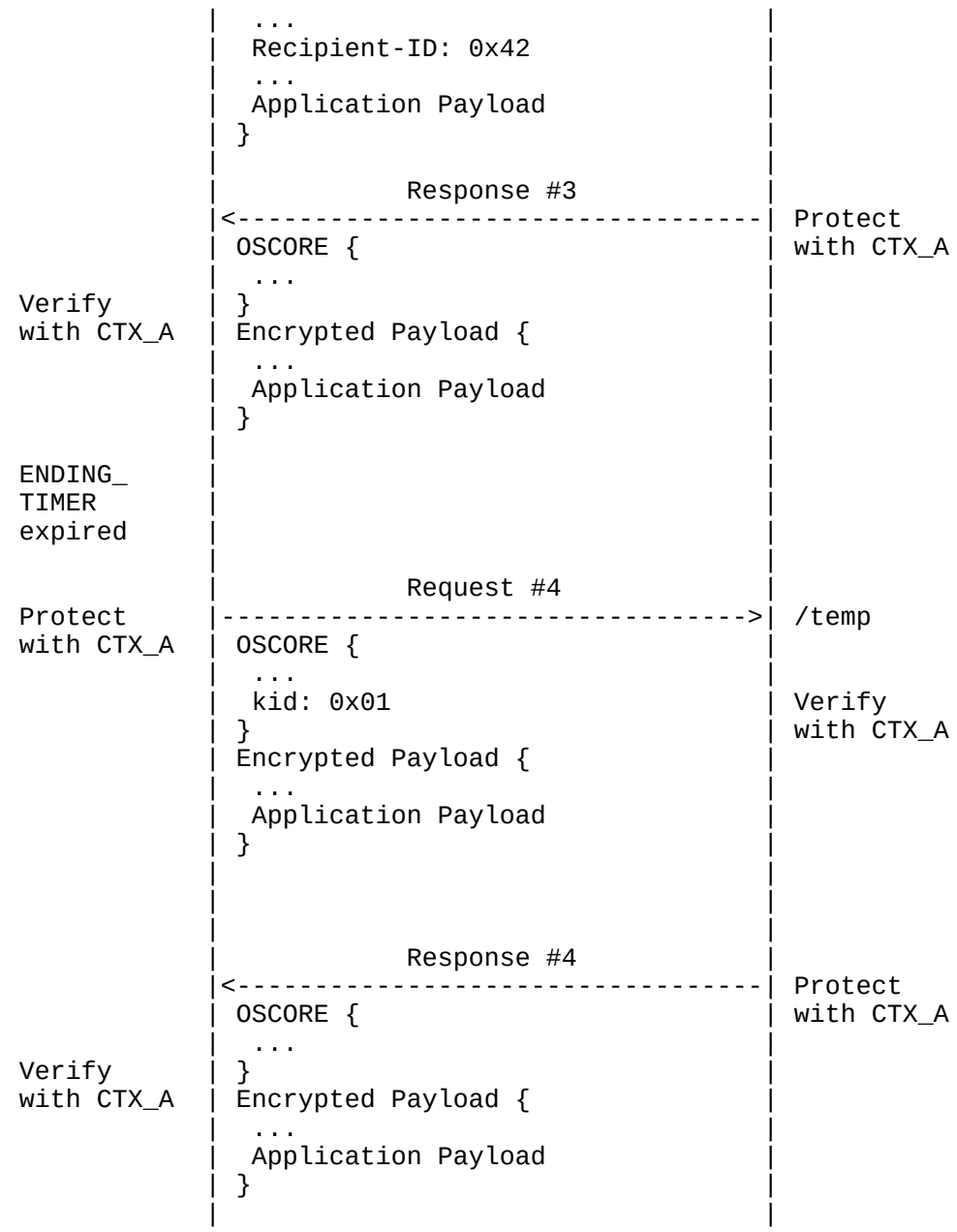


Figure 3: Example of the OSCORE ID update procedure failing when initiated with a request message

Appendix B. Document Updates

This section is to be removed before publishing as an RFC.

B.1. Version -04 to -05

* Editorial updates.
* Add additional message flow examples, including a failure case.

B.2. Version -03 to -04

* Fixes in presenting the new approach.
* Early recommendations for initial values of timers.

B.3. Version -02 to -03

* Editorial improvements.
* Improved security considerations.
* Using the ID update procedure ahead of network migration and switching to new IDs after migration.
* Update design more in line with KUDOS.

B.4. Version -01 to -02

* Split long section into subsections.
* Updated references.

B.5. Version -00 to -01

* Revised and extended error handling.
* Specify that the Recipient-ID option may need to be empty.
* Failure cases when running the ID update procedure integrated with KUDOS.
* Clarifications and editorial improvements.

B.6. Version -00

* Split out material from Key Update for OSCORE draft into this new document.

* Extended terminology.

* Editorial improvements.

Acknowledgments

The authors sincerely thank Christian Amsss, Carsten Bormann, John Preu Mattsson, and Gran Selander for their feedback and comments.

The work on this document has been partly supported by VINNOVA and the Celtic-Next projects CRITISEC and CYPRESS; and by the H2020 projects SIFIS-Home (Grant agreement 952652) and ARCADIAN-IoT (Grant agreement 101020259).

Authors' Addresses

Rikard Hglund
RISE AB
Isafjordsgatan 22
SE-16440 Stockholm Kista
Sweden
Email: rikard.hoglund@ri.se

Marco Tiloca
RISE AB
Isafjordsgatan 22
SE-16440 Stockholm Kista
Sweden
Email: marco.tiloca@ri.se