Key Update for OSCORE (KUDOS)

draft-ietf-core-oscore-key-update-12

Rikard Höglund, RISE
Marco Tiloca, RISE

IETF CoRE WG meeting – IETF 124 – November 7th, 2025

Recap of KUDOS (1/2)

- **Key Update for OSCORE (KUDOS)**
  - Renew the Master Secret and Master Salt; derive new Sender/Recipient keys
  - No change to the ID Context or Sender/Recipient ID; can achieve Forward Secrecy
  - Agnostic of the key establishment method originally used
  - Loosely inspired by Appendix B.2 of OSCORE
  - The peers update their current context CTX_OLD, deriving a new context CTX_NEW
  - Redesigned in v-10 to use a more flexible and simpler approach
    - We now explicitly define a KUDOS state machine

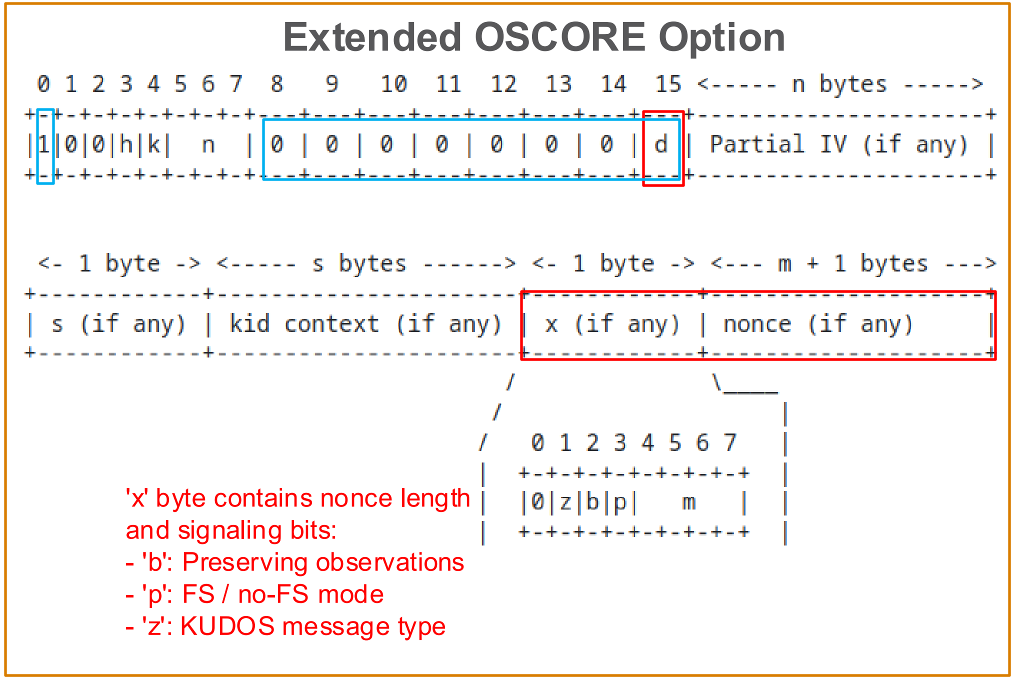
Recap of KUDOS (2/2)

- **Properties**
  - Can be initiated by either peer
  - It is robust against a peer rebooting and loss of state, avoiding the reuse of AEAD (nonce, key)
  - It typically completes in one round trip by exchanging two OSCORE-protected CoAP messages
    - The two peers achieve mutual key confirmation in a following exchange, which is protected with the newly established OSCORE Security Context
  - Flexible in terms of message flow; any CoAP message can be a KUDOS message

- **KUDOS message types**
  - Divergent message: The sender peer is offering its own nonce in the message and waiting to receive the other peer's nonce.
  - Convergent message: The sender peer is offering its own nonce in the message, has received the other peer's nonce, and is going to wait for key confirmation

Rekeying Procedure

- **Key Update for OSCORE (KUDOS)**
  - Message exchange to share two nonces N1 and N2
    - Decoupled from request/response and client/server concepts
  - Nonces are placed in new fields in OSCORE CoAP option
  - *UpdateCtx()* function for deriving new OSCORE Security Context using the two nonces, two ‘x’ bytes and CTX_OLD
  - Two modes
    - FS mode providing forward secrecy
    - No-FS mode for very constrained devices
  - No change of OSCORE identifiers
  - Expected to complete in 1 round trip

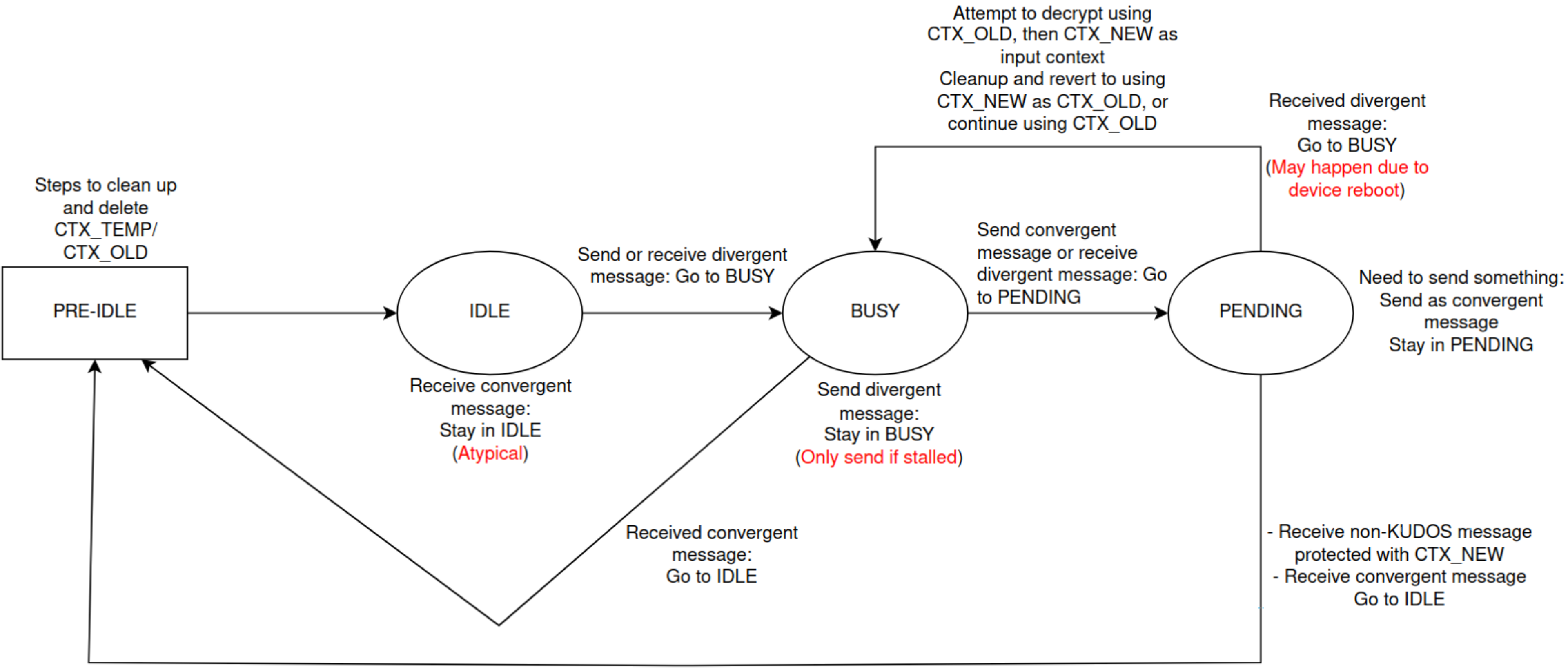


Extended OSCORE Option

```
 0 1 2 3 4 5 6 7  8   9   10  11  12  13  14  15 <----- n bytes ----->
+-+-+-+-+-+-+-+-+---+---+---+---+---+---+---+---+---------------------+
|1|0|0|h|k|  n  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | d | Partial IV (if any) |
+-+-+-+-+-+-+-+-+---+---+---+---+---+---+---+---+---------------------+

 <- 1 byte -> <------ s bytes ------> <- 1 byte -> <--- m + 1 bytes --->
+------------+----------------------+------------+---------------------+
| s (if any) | kid context (if any) | x (if any) | nonce (if any)      |
+------------+----------------------+------------+---------------------+
                                    /             \____
                                   /                   |
                                  /   0 1 2 3 4 5 6 7  |
                                  |  +-+-+-+-+-+-+-+-+ |
                                  |  |0|z|b|p|   m   | |
                                  |  +-+-+-+-+-+-+-+-+ |
```

'x' byte contains nonce length and signaling bits:
- 'b': Preserving observations
- 'p': FS / no-FS mode
- 'z': KUDOS message type

KUDOS State Machine



PRE-IDLE

Steps to clean up and delete CTX_TEMP/ CTX_OLD

IDLE

Receive convergent message: Stay in IDLE (Atypical)

Send or receive divergent message: Go to BUSY

BUSY

Send divergent message: Stay in BUSY (Only send if stalled)

Received convergent message: Go to IDLE

Send convergent message or receive divergent message: Go to PENDING

PENDING

Need to send something: Send as convergent message Stay in PENDING

Received divergent message: Go to BUSY (May happen due to device reboot)

Attempt to decrypt using CTX_OLD, then CTX_NEW as input context

Cleanup and revert to using CTX_NEW as CTX_OLD, or continue using CTX_OLD

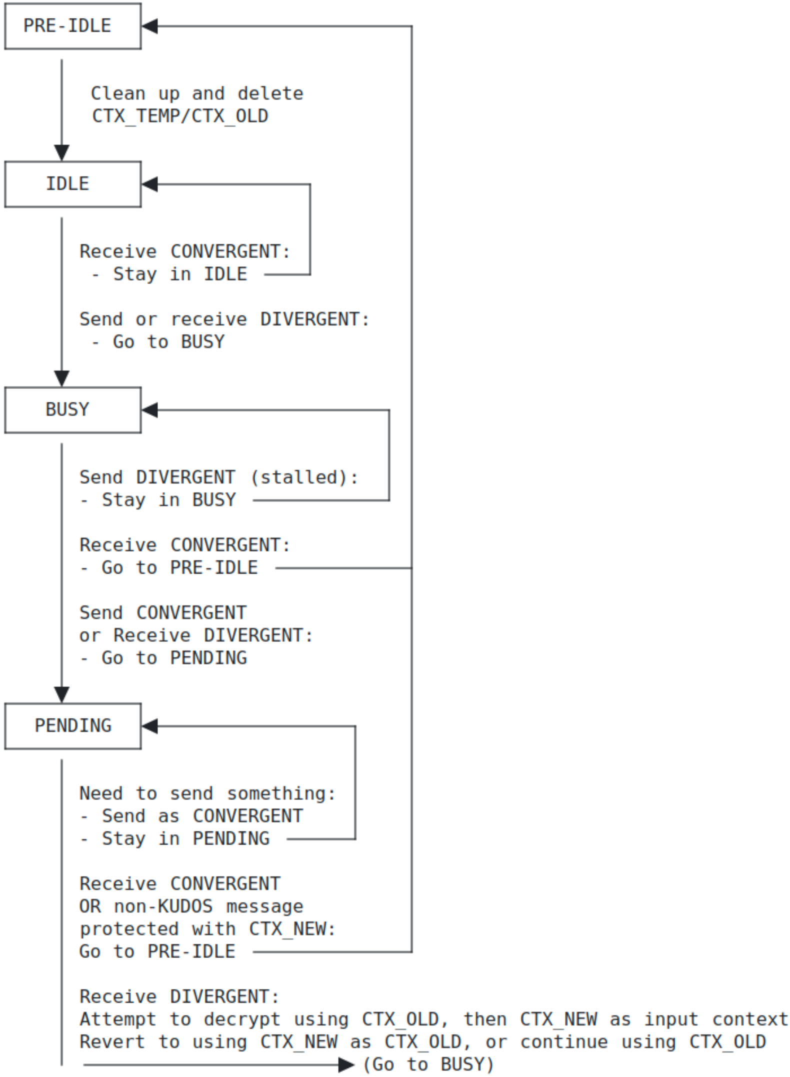
- Receive non-KUDOS message protected with CTX_NEW
- Receive convergent message

Go to IDLE

Changes for v-12

- **Added state machine as ASCII figure**
  - Now in Appendix B
  - Aligned with was shown in the previous slide
  - The state machine is described in Section 4.3.3

- **Editorial improvements**



PRE-IDLE

Clean up and delete
CTX_TEMP/CTX_OLD

IDLE

Receive CONVERGENT:
- Stay in IDLE

Send or receive DIVERGENT:
- Go to BUSY

BUSY

Send DIVERGENT (stalled):
- Stay in BUSY

Receive CONVERGENT:
- Go to PRE-IDLE

Send CONVERGENT
or Receive DIVERGENT:
- Go to PENDING

PENDING

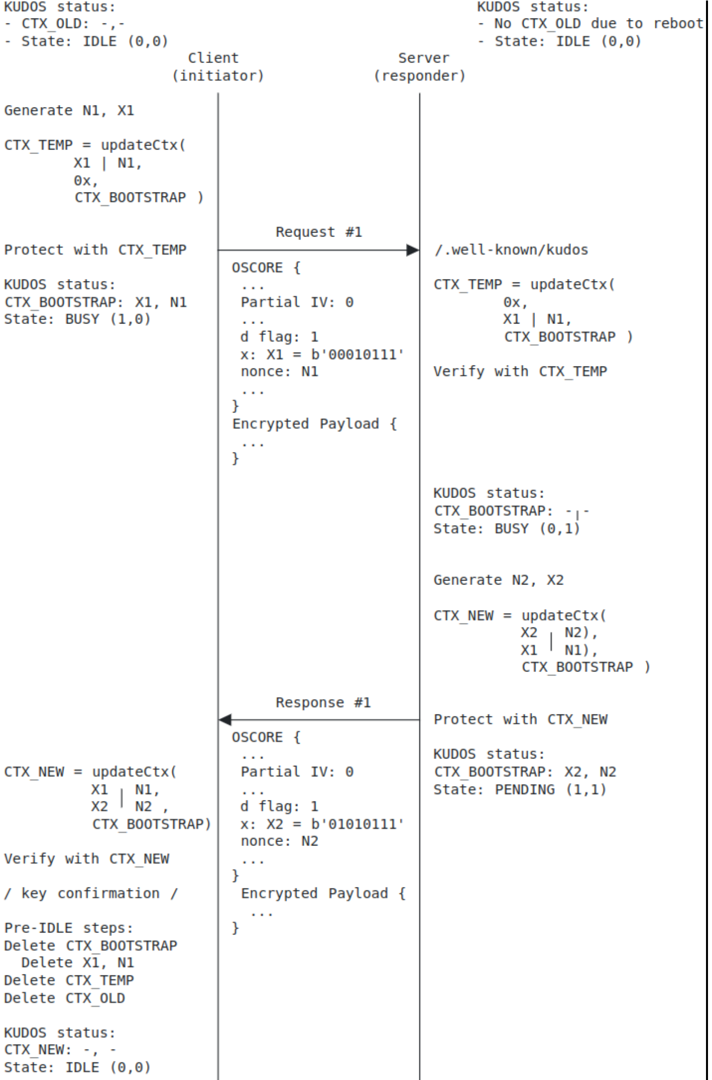
Need to send something:
- Send as CONVERGENT
- Stay in PENDING

Receive CONVERGENT
OR non-KUDOS message
protected with CTX_NEW:
Go to PRE-IDLE

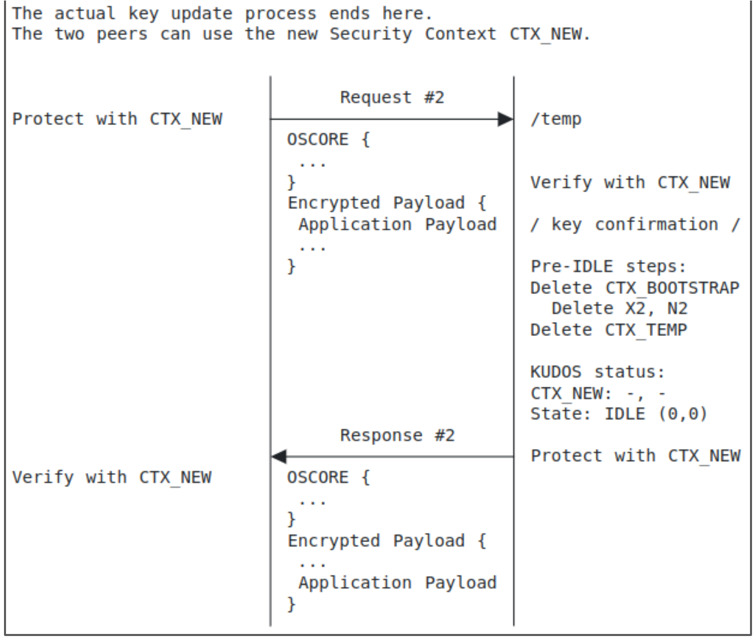
Receive DIVERGENT:
Attempt to decrypt using CTX_OLD, then CTX_NEW as input context
Revert to using CTX_NEW as CTX_OLD, or continue using CTX_OLD
(Go to BUSY)

Changes for v-12

- **Multiple additional message flow examples as appendixes**
  - KUDOS Execution Initiated with a Request Message, with Non-capable Server that has Rebooted
    - *The peers have run KUDOS prior to this KUDOS execution and have learned that they must from now on run KUDOS only in no-FS mode*
  - KUDOS Execution Initiated with a Request Message, where the Client Executes KUDOS again after the first Execution
    - *A second KUDOS execution is started by the client immediately after a successful KUDOS key update*
  - KUDOS Execution Initiated with a Request Message, where KUDOS Response #1 is Lost
    - *The server's first response is dropped by the network; thus the client retries and both sides end up deriving the same a CTX_NEW*
  - KUDOS Execution Completed using two Request Messages
    - *Both peers independently initiate KUDOS and exchange two request messages that ultimately result in the same CTX_NEW*

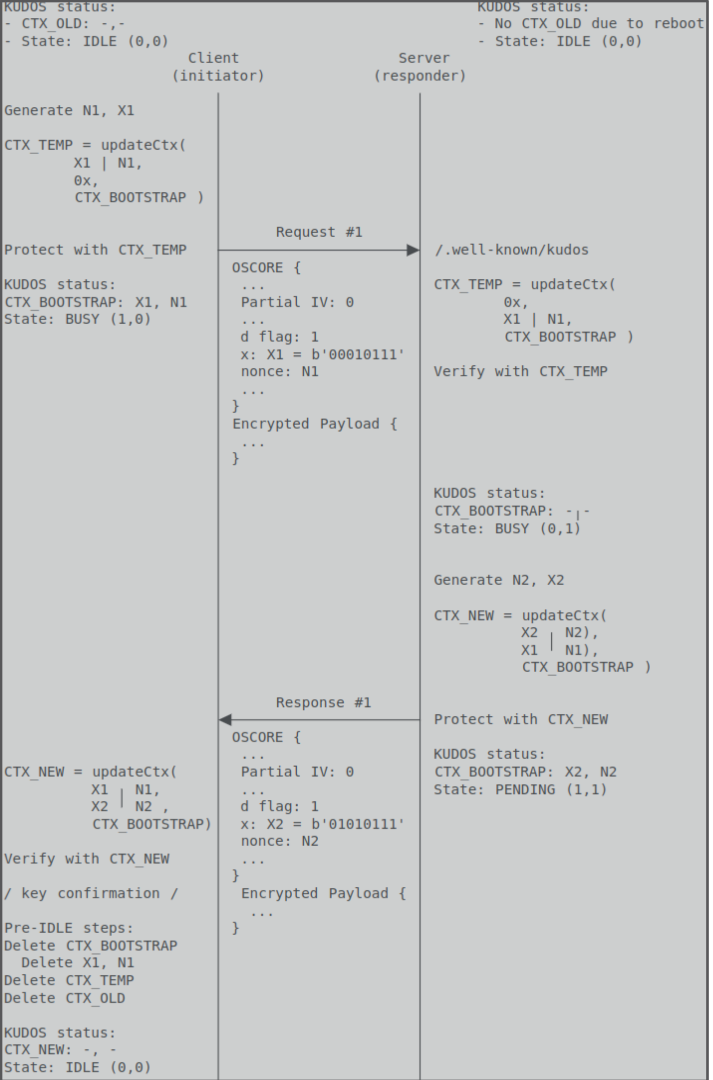


```
KUDOS status:                                KUDOS status:
- CTX_OLD: -,-                               - No CTX_OLD due to reboot
- State: IDLE (0,0)                          - State: IDLE (0,0)
                   Client                    Server
                (initiator)               (responder)
                       |                        |
Generate N1, X1        |                        |
                       |                        |
CTX_TEMP = updateCtx(  |                        |
        X1 | N1,       |                        |
        0x,            |                        |
        CTX_BOOTSTRAP )|                        |
                       |                        |
                       |       Request #1       |
Protect with CTX_TEMP  |----------------------->| /.well-known/kudos
                       | OSCORE {               |
KUDOS status:          |  ...                   | CTX_TEMP = updateCtx(
CTX_BOOTSTRAP: X1, N1  |  Partial IV: 0         |         0x,
State: BUSY (1,0)      |  ...                   |         X1 | N1,
                       |  d flag: 1             |         CTX_BOOTSTRAP )
                       |  x: X1 = b'00010111'   |
                       |  nonce: N1             | Verify with CTX_TEMP
                       |  ...                   |
                       | }                      |
                       | Encrypted Payload {    |
                       |  ...                   |
                       | }                      |
                       |                        |
                       |                        | KUDOS status:
                       |                        | CTX_BOOTSTRAP: -,-
                       |                        | State: BUSY (0,1)
                       |                        |
                       |                        | Generate N2, X2
                       |                        |
                       |                        | CTX_NEW = updateCtx(
                       |                        |           X2 | N2),
                       |                        |           X1 | N1),
                       |                        |           CTX_BOOTSTRAP )
                       |                        |
                       |       Response #1      |
                       |<-----------------------| Protect with CTX_NEW
                       | OSCORE {               |
                       |  ...                   | KUDOS status:
CTX_NEW = updateCtx(   |  Partial IV: 0         | CTX_BOOTSTRAP: X2, N2
          X1 | N1,     |  ...                   | State: PENDING (1,1)
          X2 | N2 ,    |  d flag: 1             |
          CTX_BOOTSTRAP)  x: X2 = b'01010111'   |
                       |  nonce: N2             |
Verify with CTX_NEW    |  ...                   |
                       | }                      |
/ key confirmation /   |  Encrypted Payload {   |
                       |   ...                  |
Pre-IDLE steps:        | }                      |
Delete CTX_BOOTSTRAP   |                        |
  Delete X1, N1        |                        |
Delete CTX_TEMP        |                        |
Delete CTX_OLD         |                        |
                       |                        |
KUDOS status:          |                        |
CTX_NEW: -, -          |                        |
State: IDLE (0,0)      |                        |
```

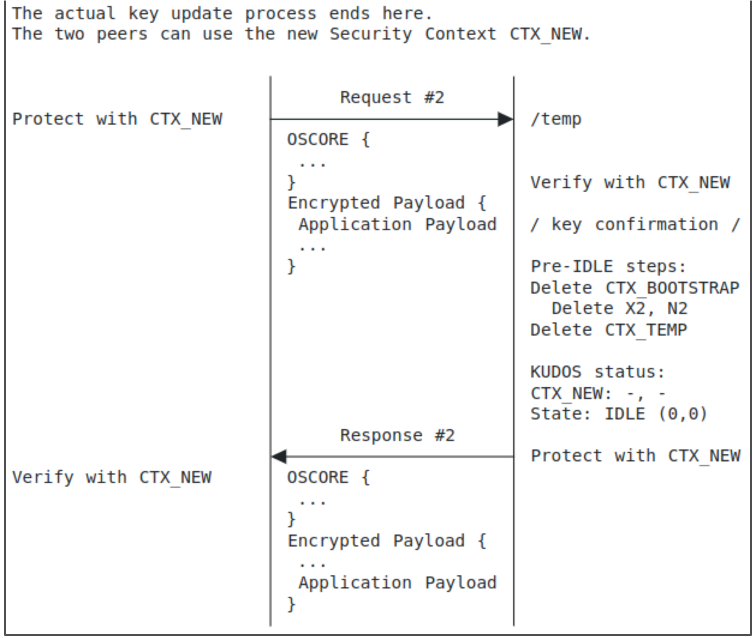


```
The actual key update process ends here.
The two peers can use the new Security Context CTX_NEW.

                        |       Request #2       |
Protect with CTX_NEW    |----------------------->| /temp
                        | OSCORE {               |
                        |  ...                   |
                        | }                      | Verify with CTX_NEW
                        | Encrypted Payload {    |
                        |  Application Payload   | / key confirmation /
                        |  ...                   |
                        | }                      | Pre-IDLE steps:
                        |                        | Delete CTX_BOOTSTRAP
                        |                        |   Delete X2, N2
                        |                        | Delete CTX_TEMP
                        |                        |
                        |                        | KUDOS status:
                        |                        | CTX_NEW: -, -
                        |                        | State: IDLE (0,0)
                        |       Response #2      |
                        |<-----------------------| Protect with CTX_NEW
Verify with CTX_NEW     | OSCORE {               |
                        |  ...                   |
                        | }                      |
                        | Encrypted Payload {    |
                        |  ...                   |
                        |  Application Payload   |
                        | }                      |
```

Successful KUDOS Execution Initiated with a Request Message, with Non-capable Server that has Rebooted



```
KUDOS status:                                          KUDOS status:
- CTX_OLD: -,-                                         - No CTX_OLD due to reboot
- State: IDLE (0,0)                                    - State: IDLE (0,0)
                     Client                  Server
                   (initiator)             (responder)
                         |                      |
Generate N1, X1          |                      |
                         |                      |
CTX_TEMP = updateCtx(    |                      |
        X1 | N1,         |                      |
        0x,              |                      |
        CTX_BOOTSTRAP )  |                      |
                         |                      |
                         |      Request #1      |
Protect with CTX_TEMP    +--------------------->| /.well-known/kudos
                         | OSCORE {             |
KUDOS status:            |  ...                 | CTX_TEMP = updateCtx(
CTX_BOOTSTRAP: X1, N1    |  Partial IV: 0       |         0x,
State: BUSY (1,0)        |  ...                 |         X1 | N1,
                         |  d flag: 1           |         CTX_BOOTSTRAP )
                         |  x: X1 = b'00010111' |
                         |  nonce: N1           | Verify with CTX_TEMP
                         |  ...                 |
                         | }                    |
                         | Encrypted Payload {  |
                         |  ...                 |
                         | }                    |
                         |                      |
                         |                      | KUDOS status:
                         |                      | CTX_BOOTSTRAP: -,-
                         |                      | State: BUSY (0,1)
                         |                      |
                         |                      | Generate N2, X2
                         |                      |
                         |                      | CTX_NEW = updateCtx(
                         |                      |           X2 | N2),
                         |                      |           X1 | N1),
                         |                      |           CTX_BOOTSTRAP )
                         |                      |
                         |      Response #1     |
                         |<---------------------+ Protect with CTX_NEW
                         | OSCORE {             |
                         |  ...                 | KUDOS status:
CTX_NEW = updateCtx(     |  Partial IV: 0       | CTX_BOOTSTRAP: X2, N2
          X1 | N1,       |  ...                 | State: PENDING (1,1)
          X2 | N2 ,      |  d flag: 1           |
          CTX_BOOTSTRAP) |  x: X2 = b'01010111' |
                         |  nonce: N2           |
Verify with CTX_NEW      |  ...                 |
                         | }                    |
/ key confirmation /     |  Encrypted Payload { |
                         |   ...                |
Pre-IDLE steps:          | }                    |
Delete CTX_BOOTSTRAP     |                      |
  Delete X1, N1          |                      |
Delete CTX_TEMP          |                      |
Delete CTX_OLD           |                      |
                         |                      |
KUDOS status:            |                      |
CTX_NEW: -, -            |                      |
State: IDLE (0,0)        |                      |
```



```
The actual key update process ends here.
The two peers can use the new Security Context CTX_NEW.

                            |                       |
                            |      Request #2       |
Protect with CTX_NEW        +---------------------->| /temp
                            | OSCORE {              |
                            |  ...                  |
                            | }                     | Verify with CTX_NEW
                            | Encrypted Payload {   |
                            |  Application Payload  | / key confirmation /
                            |  ...                  |
                            | }                     | Pre-IDLE steps:
                            |                       | Delete CTX_BOOTSTRAP
                            |                       |   Delete X2, N2
                            |                       | Delete CTX_TEMP
                            |                       |
                            |                       | KUDOS status:
                            |                       | CTX_NEW: -, -
                            |                       | State: IDLE (0,0)
                            |                       |
                            |      Response #2      |
                            |<----------------------+ Protect with CTX_NEW
Verify with CTX_NEW         | OSCORE {              |
                            |  ...                  |
                            | }                     |
                            | Encrypted Payload {   |
                            |  ...                  |
                            |  Application Payload  |
                            | }                     |
```

Successful KUDOS Execution Initiated with a Request Message, with Non-capable Server that has Rebooted

Summary and next steps

- **KUDOS implementations**
  - Update the implementation in Java [1] to be aligned with the latest design
  - Update the implementation in C for Contiki-NG to be aligned with the latest design

- **Procedural: Move content relevant to SCHC from draft-ietf-schc-8824-update**
  - Considering the timeline, that content about the extended OSCORE option fits better in this draft

- **Comments and reviews are welcome!**

[1] https://github.com/rikard-sics/californium/blob/oscore_cmd/cf-oscore/

Thank you!

Comments/questions?

https://github.com/core-wg/oscore-key-update

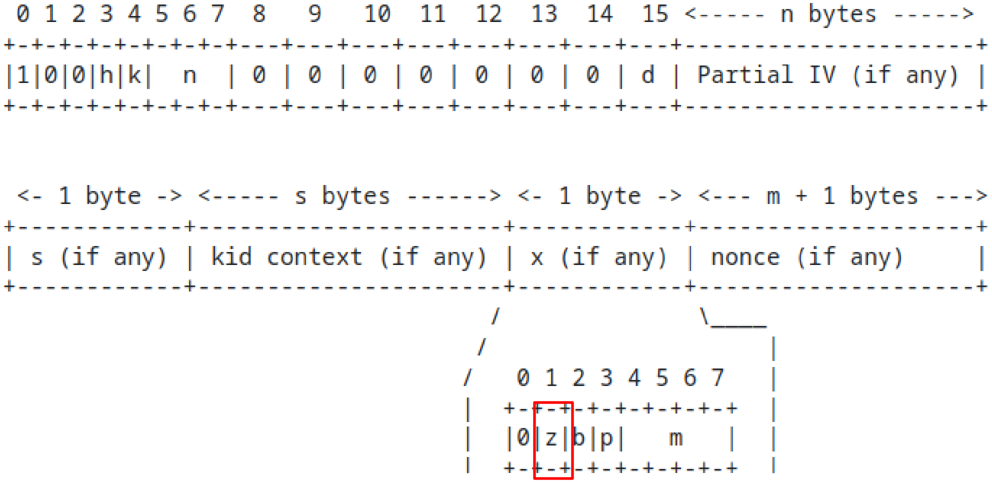
Backup

KUDOS Message Types

- **Two types of KUDOS messages, distinguished by the 7th least significant bit 'z' in the 'x' byte**
  - Indicates if only one or both nonces have been exchanged

- **z = 0: "divergent message"**
- This message is protected with the temporary Security Context CTX_TEMP (was CTX_1)
- The sender peer is offering its own nonce in the message and waiting to receive the other peer's nonce.

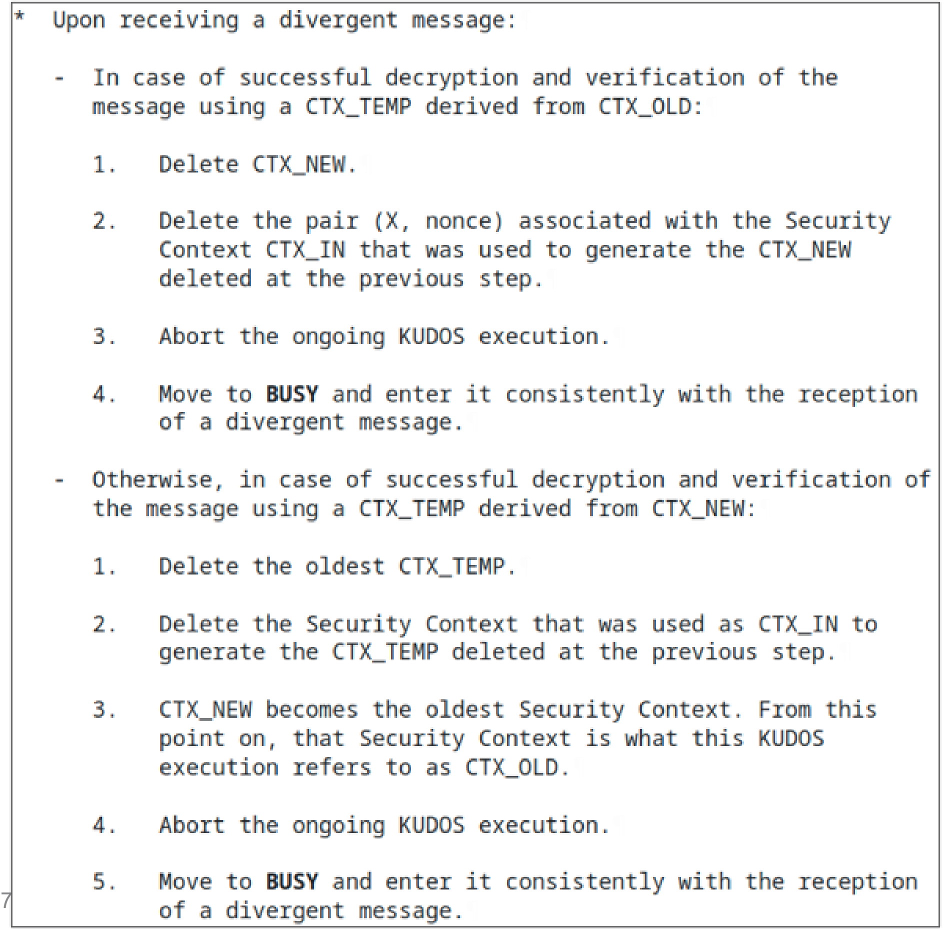
- **z = 1: "convergent message"**
- This message is protected with the final Security Context CTX_NEW.
- The sender peer is offering its own nonce in the message, has received the other peer's nonce, and is going to wait for key confirmation



```
 0 1 2 3 4 5 6 7  8   9   10  11  12  13  14  15 <----- n bytes ----->
+-+-+-+-+-+-+-+-+---+---+---+---+---+---+---+---+---------------------+
|1|0|0|h|k|  n  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | d | Partial IV (if any) |
+-+-+-+-+-+-+-+-+---+---+---+---+---+---+---+---+---------------------+


 <- 1 byte -> <----- s bytes ------> <- 1 byte -> <--- m + 1 bytes --->
+------------+----------------------+------------+---------------------+
| s (if any) | kid context (if any) | x (if any) | nonce (if any)      |
+------------+----------------------+------------+---------------------+
                                   /              \____
                                  /                    |
                                 /   0 1 2 3 4 5 6 7   |
                                 |  +-+-+-+-+-+-+-+-+  |
                                 |  |0|z|b|p|   m   |  |
                                 |  +-+-+-+-+-+-+-+-+  |
```

Note: The z bit used to have another meaning

State Machine when in PENDING



* Upon receiving a divergent message:

  - In case of successful decryption and verification of the message using a CTX_TEMP derived from CTX_OLD:

    1. Delete CTX_NEW.

    2. Delete the pair (X, nonce) associated with the Security Context CTX_IN that was used to generate the CTX_NEW deleted at the previous step.

    3. Abort the ongoing KUDOS execution.

    4. Move to **BUSY** and enter it consistently with the reception of a divergent message.

  - Otherwise, in case of successful decryption and verification of the message using a CTX_TEMP derived from CTX_NEW:

    1. Delete the oldest CTX_TEMP.

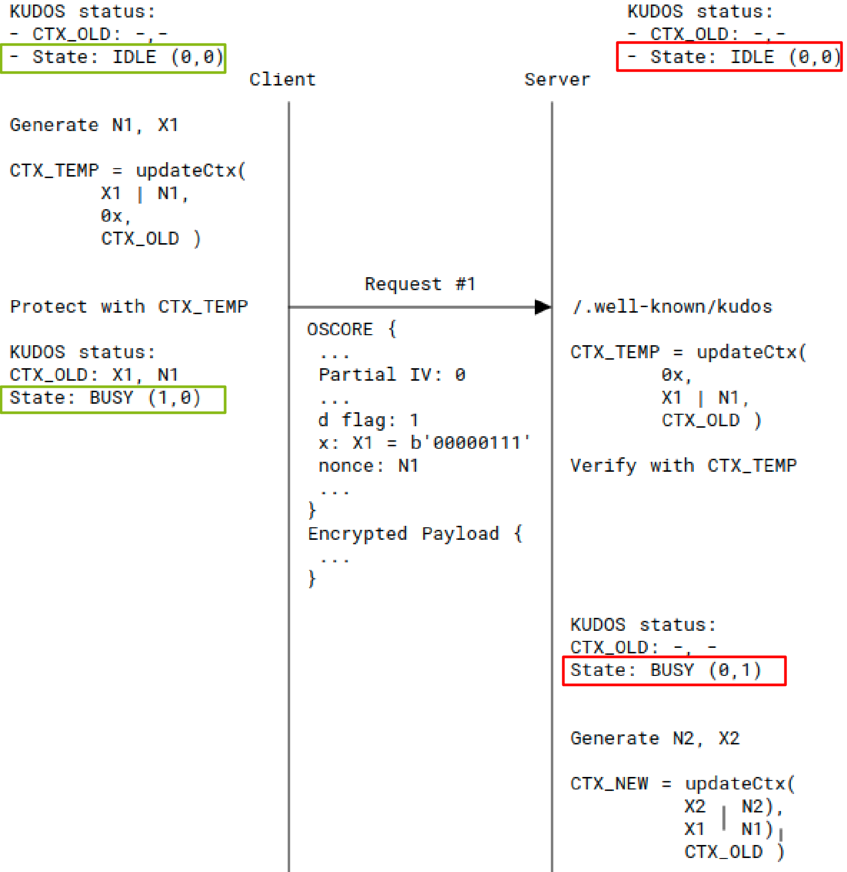
    2. Delete the Security Context that was used as CTX_IN to generate the CTX_TEMP deleted at the previous step.

    3. CTX_NEW becomes the oldest Security Context. From this point on, that Security Context is what this KUDOS execution refers to as CTX_OLD.

    4. Abort the ongoing KUDOS execution.

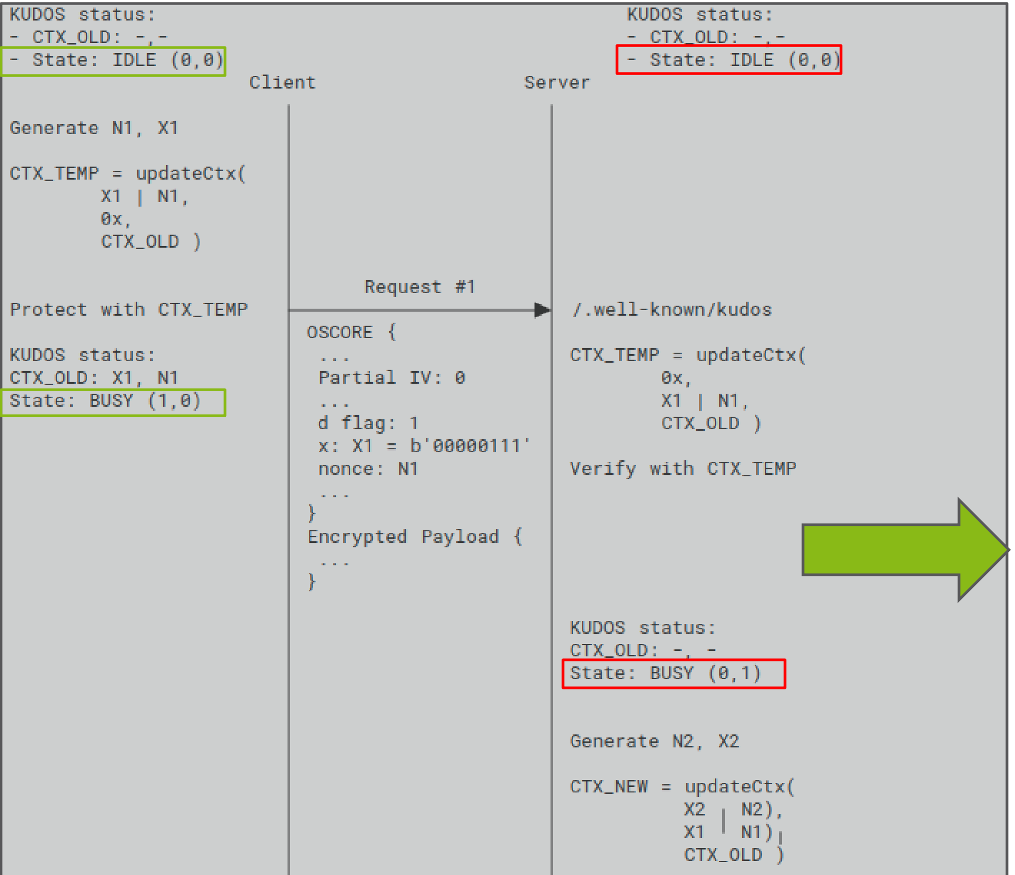
    5. Move to **BUSY** and enter it consistently with the reception of a divergent message.

Example Execution

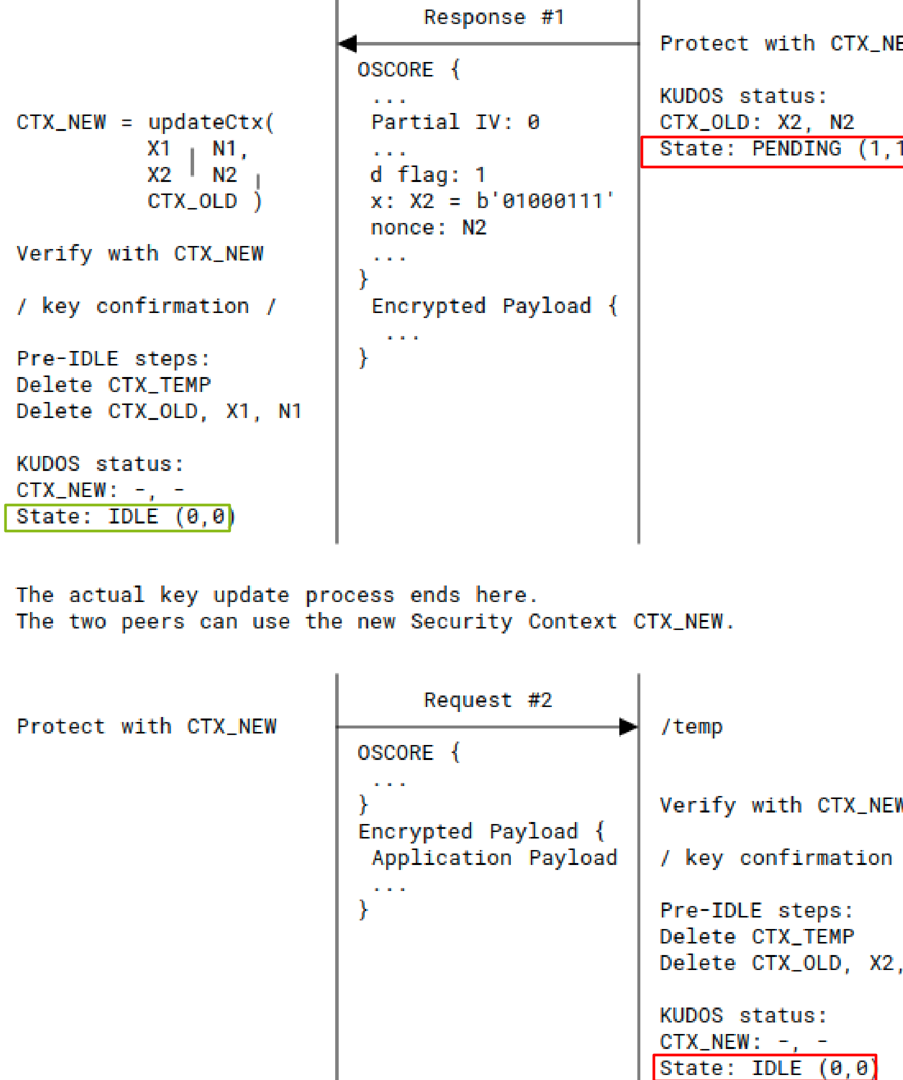


```
KUDOS status:                                               KUDOS status:
- CTX_OLD: -,-                                              - CTX_OLD: -,-
- State: IDLE (0,0)                                         - State: IDLE (0,0)
                        Client                   Server

Generate N1, X1

CTX_TEMP = updateCtx(
        X1 | N1,
        0x,
        CTX_OLD )

                                  Request #1
Protect with CTX_TEMP           ----------------------->  /.well-known/kudos
                                OSCORE {
KUDOS status:                    ...                      CTX_TEMP = updateCtx(
CTX_OLD: X1, N1                  Partial IV: 0                    0x,
State: BUSY (1,0)                ...                              X1 | N1,
                                 d flag: 1                        CTX_OLD )
                                 x: X1 = b'00000111'
                                 nonce: N1                Verify with CTX_TEMP
                                 ...
                                }
                                Encrypted Payload {
                                 ...
                                }

                                                          KUDOS status:
                                                          CTX_OLD: -, -
                                                          State: BUSY (0,1)


                                                          Generate N2, X2

                                                          CTX_NEW = updateCtx(
                                                                    X2 | N2),
                                                                    X1 | N1)|
                                                                    CTX_OLD )
```

Example Execution



```
KUDOS status:                                   KUDOS status:
- CTX_OLD: -,-                                  - CTX_OLD: -,-
- State: IDLE (0,0)                             - State: IDLE (0,0)
                      Client                  Server

Generate N1, X1

CTX_TEMP = updateCtx(
        X1 | N1,
        0x,
        CTX_OLD )

                                Request #1
Protect with CTX_TEMP      ------------------>  /.well-known/kudos
                           OSCORE {
KUDOS status:               ...                 CTX_TEMP = updateCtx(
CTX_OLD: X1, N1             Partial IV: 0               0x,
State: BUSY (1,0)           ...                         X1 | N1,
                            d flag: 1                   CTX_OLD )
                            x: X1 = b'00000111'
                            nonce: N1           Verify with CTX_TEMP
                            ...
                           }
                           Encrypted Payload {
                            ...
                           }

                                                KUDOS status:
                                                CTX_OLD: -, -
                                                State: BUSY (0,1)


                                                Generate N2, X2

                                                CTX_NEW = updateCtx(
                                                          X2 | N2),
                                                          X1 | N1)|
                                                          CTX_OLD )
```



```
                                 Response #1
                             <------------------   Protect with CTX_NE
                             OSCORE {
                              ...                  KUDOS status:
CTX_NEW = updateCtx(          Partial IV: 0        CTX_OLD: X2, N2
           X1 | N1,           ...                  State: PENDING (1,1
           X2 | N2 |          d flag: 1
           CTX_OLD )          x: X2 = b'01000111'
                              nonce: N2
Verify with CTX_NEW           ...
                             }
/ key confirmation /          Encrypted Payload {
                              ...
Pre-IDLE steps:              }
Delete CTX_TEMP
Delete CTX_OLD, X1, N1

KUDOS status:
CTX_NEW: -, -
State: IDLE (0,0)

The actual key update process ends here.
The two peers can use the new Security Context CTX_NEW.

                                 Request #2
Protect with CTX_NEW         ------------------>   /temp
                             OSCORE {
                              ...
                             }                     Verify with CTX_NEW
                             Encrypted Payload {
                              Application Payload  / key confirmation
                              ...
                             }                     Pre-IDLE steps:
                                                   Delete CTX_TEMP
                                                   Delete CTX_OLD, X2,

                                                   KUDOS status:
                                                   CTX_NEW: -, -
                                                   State: IDLE (0,0)
```