Identifier Update for OSCORE

draft-ietf-core-oscore-id-update-04

Rikard Höglund, RISE
Marco Tiloca, RISE

IETF CoRE WG meeting – IETF 123 – July 22nd, 2025

Recap

- **Method for updating peers' OSCORE Sender/Recipient IDs**
  - This procedure can be initiated by a client or by a server
  - Peers start with an original OSCORE Security Context, CTX_A, and use the new IDs for deriving a new OSCORE Security Context CTX_B
- **Properties**
  - The message sender indicates its new wished Recipient ID
  - Both peers have to opt-in and agree in order for the IDs to be updated
  - Must not be done immediately following a reboot if run standalone (e.g., KUDOS must be run first)
  - Offered Recipient ID must not be used yet under the same (Master Secret, Master Salt, ID Context)
  - Received Recipient ID must not be used yet as own Sender ID under the same triple
  - **Goal**: Mitigate privacy issues due to message correlation and tracking of OSCORE peers
- **Document status**
  - Submitted v-03 ahead of the cut-off with a new design more aligned with KUDOS
  - Submitted v-04 on Sunday with minor improvements

Updated design in line with KUDOS

- **The procedure starts when either peer**
  - Sends a message including the Recipient-ID Option, or
  - Receives such a message from the other peer

- **During the procedure**
- **Sending a first message**
  - The first messages sent after the procedure has started must include the Recipient-ID Option, if this peer hasn't offered its Recipient ID already

- **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 (with value the Recipient ID received from the other peer)

- **Sending Subsequent Messages**
  - A peer must send a message with the Recipient ID Option regularly, specifically when the local timer REPEAT_TIMER expires

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

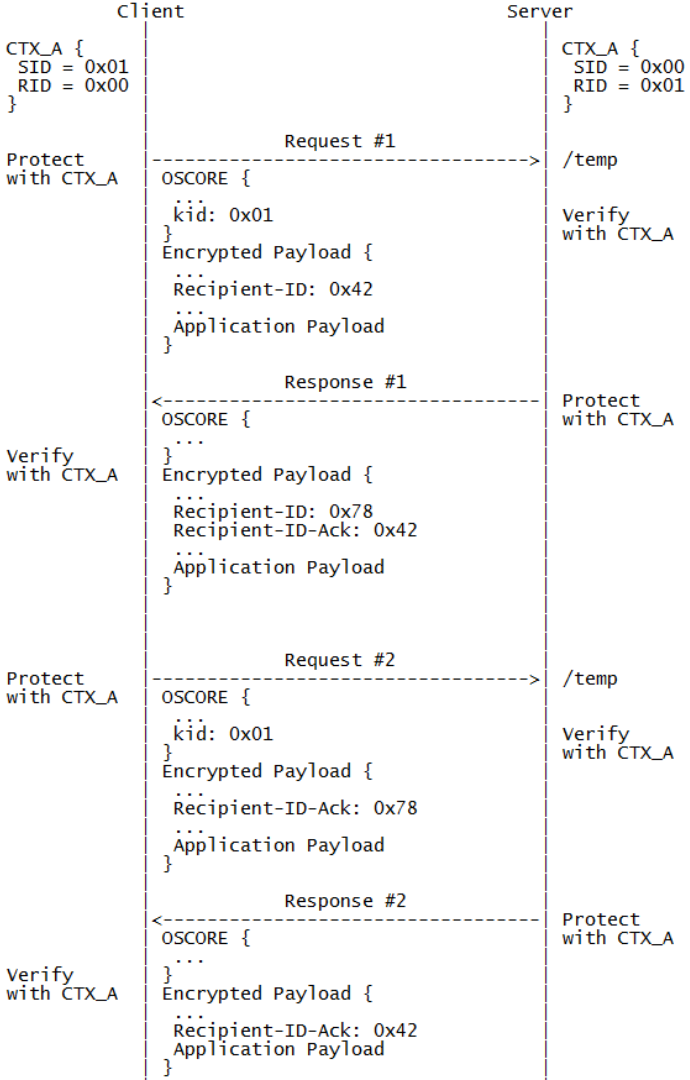
Table 1: The Recipient-ID Option. C=Critical, U=Unsafe, N=NoCacheKey, R=Repeatable

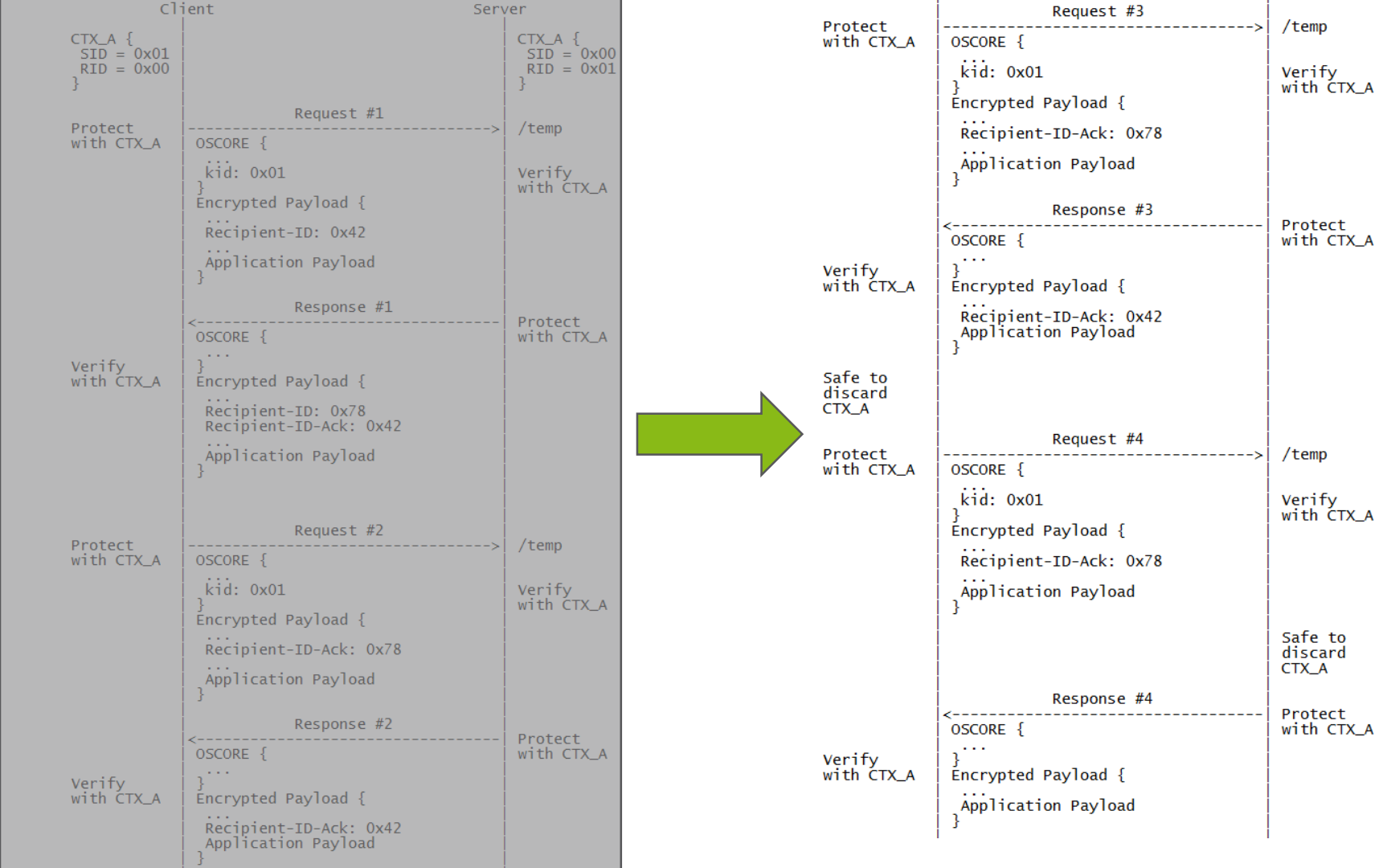
No.	C	U	N	R	Name	Format	Length	Default
TBD32					Recipient-ID-Ack	opaque	any	(none)

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

Updated design in line with KUDOS

- **Procedure Completion**
  - **Success**
    - If a peer has received and successfully verified at least three message from the other peer containing the Recipient-ID-Ack Option
    - Now safe to delete CTX_A (does not mean that CTX_A has to be deleted at this point)
    - CTX_B is considered valid and can be used
  - **Failure**
    - An ENDING_TIMER, is maintained and started when the procedure starts
    - If the ENDING_TIMER expires, the procedure times out without confirmation, and fails
    - The offered Recipient ID must be discarded and added to the list of IDs to prevent reuse


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
Protect
with CTX_A
OSCORE {
...
}
Verify
with CTX_A
Encrypted Payload {
...
Recipient-ID-Ack: 0x42
Application Payload
}


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

Other changes for v-03 & v-04

- **Establishing new OSCORE identifiers ahead of network migration**
  - Peers SHOULD NOT begin using the 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
  - The peers must not begin using the OSCORE Security Context CTX_B until after the network migration has taken place

- **Add security considerations**
  - Inherit security considerations from OSCORE RFC and KUDOS draft
  - Change of IDs alone might not completely prevent adversaries from recognizing traffic patterns
    - New OSCORE Security Contexts start their Partial IV at 0
  - Other information such as addressing information, may still be used to track the peers
    - Start using the new OSCORE IDs upon network migration; 1) Changing the network address and 2) changing the link-layer address

Summary and next steps

- **Add more examples, including failure cases**
  - As message flow examples in Appendix
- **Re-consider how the ID update procedure works when ran integrated with an execution of KUDOS**
  - Text describing how this should work
  - Add message flow examples
- **Consider possible optimizations of the new design**
- **Implement the OSCORE ID Update procedure, building on our Java OSCORE implementation**
- **Comments and reviews are welcome!**

Thank you!

Comments/questions?

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

Backup