Key Update for OSCORE (KUDOS)

draft-ietf-core-oscore-key-update-04

Rikard Höglund, RISE
Marco Tiloca, RISE

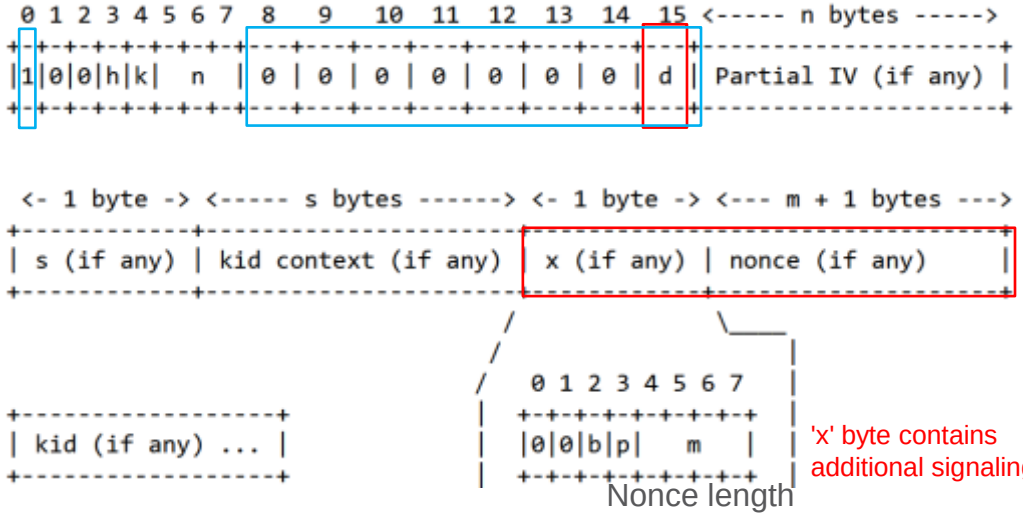
IETF 116 meeting – Yokohama – March 28th, 2023

Recap

- (1) Key Update for OSCORE (KUDOS)
  - Renew the Master Secret and Master Salt; derive new Sender/Recipient keys
  - No change to the ID Context; can achieve Perfect Forward Secrecy
  - Loosely inspired by Appendix B.2 of OSCORE
- (2) AEAD Key Usage Limits in OSCORE
  - Content moved to new document *draft-ietf-core-oscore-key-limits*
- (3) Update of OSCORE Sender/Recipient IDs
  - Exchanging desired new Recipient ID through a new CoAP Option

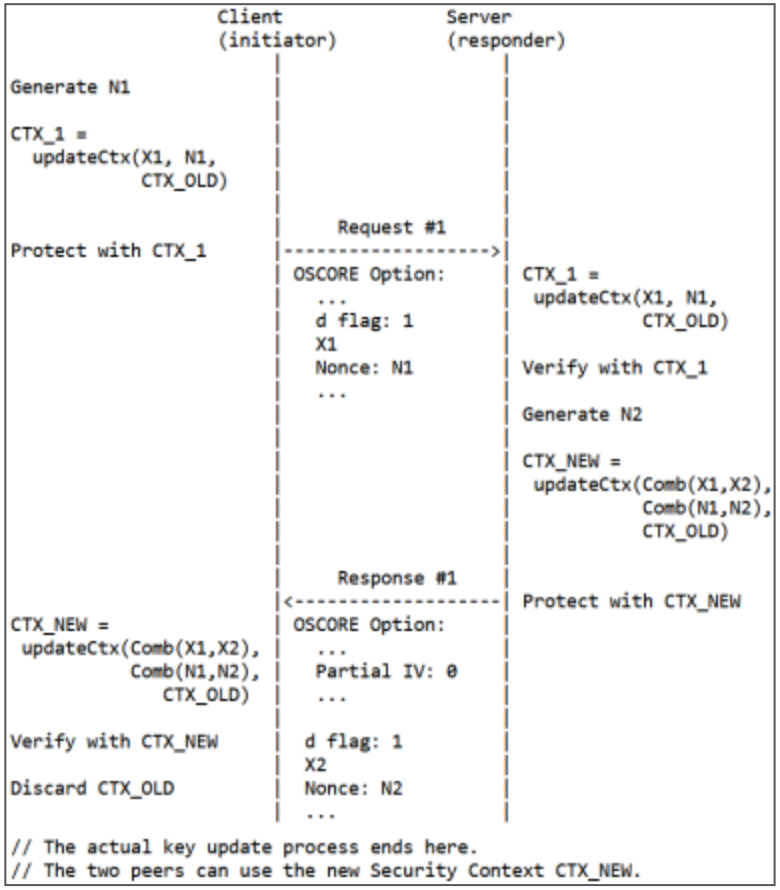
Rekeying procedure

- **Key Update for OSCORE (KUDOS)**
  - Message exchange to share nonces N1 and N2
  - Nonces are placed in new field in OSCORE CoAP option
  - *UpdateCtx()* function for deriving new OSCORE Security Context using the nonces and ‘x’ bytes
  - Extended OSCORE Option



'x' byte contains additional signaling flags

KUDOS forward message flow



Update summary: KUDOS

- **Security Context CTX_OLD should not be used for sending messages**
  - After a peer has transitioned to using CTX_NEW, the context CTX_OLD should not be used
- **Switch of terminology from client/server initiated to forward/reverse message flow**
  - Client-initiated -> Forward message flow
  - Server-initiated -> Reverse message flow
- **Stress that usage the EDHOC EAD item KUDOS_EAD is optional**
  - It can be used to learn the capabilities and KUDOS modes supported by the other peer
  - However, peers can always rely on signaling using the "No Forward Secrecy" bit, 'p', in the 'x' byte of the OSCORE Option

EAD items are optional data that can be exchanged during an EDHOC execution

Update summary: KUDOS

- **KUDOS can be used for “active” rekeying**
  - A peer can decide that it wants to run the KUDOS procedure at any point and for any reason

- **Better handling of message exchanges overlapping with KUDOS execution**
  - For instance: A client sends a request and shortly after that executes KUDOS
  - In such case the CoAP request is protected with CTX_OLD, while the CoAP response from the server is protected with CTX_NEW
  - Thus, the client must be ready CoAP responses protected with a different OSCORE Security Context than what was used to protect the corresponding request
  - For the reverse message flow, this can happen if the client uses NSTART > 1 and one of the requests results to be a KUDOS trigger. Then the other requests from the client will be served by the server after KUDOS has completed, and thus the responses will be protected with CTX_NEW.

Update summary: KUDOS

- **Registration of a /.well-known/kudos resource**
  - A peer can use this resource as target for KUDOS messages that are requests
  - Registered resource type *core.kudos* which can be used to discover KUDOS resources
- **Added also EDHOC-KeyUpdate as a key update method**
  - In the current list of methods for Rekeying OSCORE
- **Define what is considered “hard” limits for rekeying**
  - Specifically, expiration of an OSCORE Security Context and exceeding the key usage limit serves as hard limits, at which point a peer MUST run KUDOS

Update summary: KUDOS

- **Section about preventing deadlocks**
  - Listed a number of scenarios that can cause deadlocks, and how the peers must act to prevent this from occurring
  - These could otherwise occur when KUDOS fails to complete on one peer but succeeds on the other, and the peer where KUDOS completed initiates a new execution of KUDOS

- **Added section with expected message overhead for KUDOS**

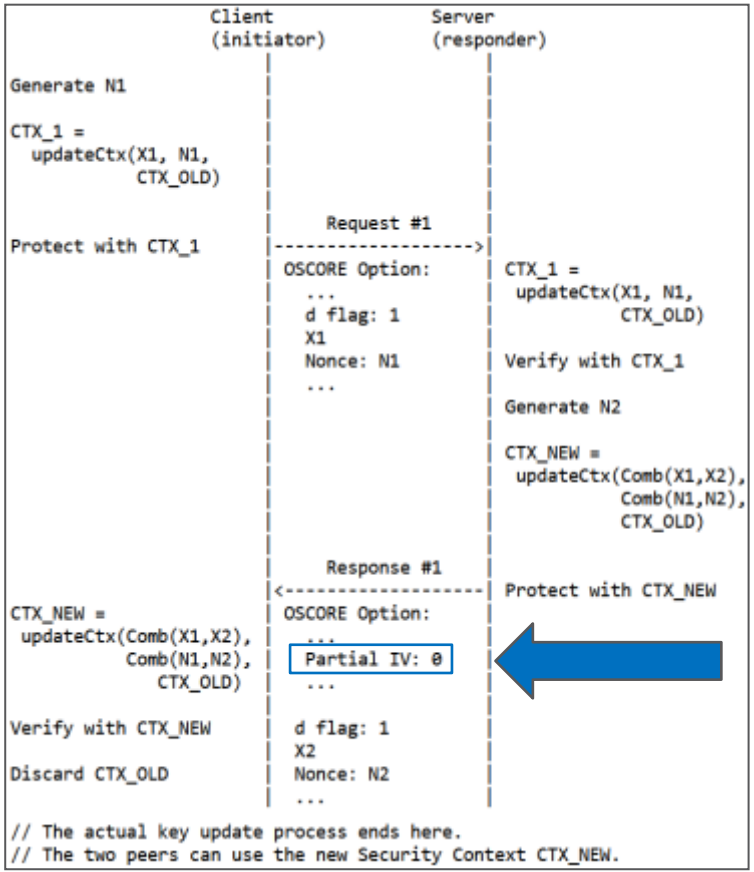
Nonce size	Overhead of one KUDOS message	Overhead of one KUDOS execution
1	3	6
8	10	20
16	18	36

Update summary: KUDOS

- **In this case, the Server MUST include a PIV in Response #1**
- This prevents a reuse of the same pair (AEAD nonce, key) from the server, as otherwise shown in this table:

Peer	Message	Nonce	Sender key from	Pair reuse
Client	Request #1	A	CTX_1	No
Server	Response #1	A	CTX_NEW	No
Client	Request #2	A	CTX_NEW	No
Server	Response #2	A	CTX_NEW	YES

- **General fix for this added with text updating RFC8613**
  - This was also raised in the IETF 115 meeting
  - *If the server is using a different Security Context for the response compared to what was used to verify the request, then the server MUST include its Sender Sequence Number as Partial IV in the response and use it to build the AEAD nonce to protect the response.*



Update summary: ID Update

- **Preservation of observations when updating OSCORE Sender/Recipient IDs**
  - A peer must store the value of the 'kid' parameter from the original Observe request
  - 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 when verifying/protecting notifications

- **Motivate use of the procedure for update of OSCORE Sender/Recipient IDs**
  - Helps mitigate the ability of an adversary to correlate the two peer's communication
  - For instance, two peers may want to use this procedure before switching to a different network for their communication

Next steps

- **Work on creating an implementation**
  - Building on OSCORE Java implementation w. Appendix B.2 support
- **Expand section about OSCORE ID Update**
  - Step-by-step explanation of current examples
- **Address pending open points**
- **Comments and reviews are welcome!**

Thank you!

Comments/questions?

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

Update of Sender/Recipient IDs

- **Method for updating peers' OSCORE Sender/Recipient IDs**
  - Based on earlier discussions on the mailing list [1][2] and on [3]
  - This procedure can be embedded in a KUDOS execution or run standalone
  - This procedure can be initiated by a client or by a server
  - Content moved from old appendix to document body and improved (Section 5)

No.	C	U	N	R	Name	Format	Length	Default
TBD1					Recipient-ID	opaque	0-7	(none)

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

- **Properties**
  - The sender indicates its new wished Recipient ID in the new Recipient-ID Option (class E)
  - Both peers have to opt-in and agree in order for the IDs to be updated
  - Changing IDs practically triggers derivation of new OSCORE Security Context
  - Must not be done immediately following a reboot (e.g., KUDOS must be run first)
  - Offered Recipient ID must be not used yet under (Master Secret, Master Salt, ID Context)
  - Received Recipient ID must not be used yet as own Sender ID under the same triple

- Examples are provided in Sections 5.1.1 and 5.1.2

[1] https://mailarchive.ietf.org/arch/msg/core/GXsKO4wKdt3RTZnQZxOzRdIG9QI/
[2] https://mailarchive.ietf.org/arch/msg/core/ClwcSF0BUVxDas8BpgT0WY1yQrY/
[3] https://github.com/core-wg/oscore/issues/263#issue-946989659