Key Update for OSCORE (KUDOS)

draft-ietf-core-oscore-key-update-01

Rikard Höglund, RISE
Marco Tiloca, RISE

IETF 113, CoRE WG, March 25th, 2022
Content Recap

› OSCORE (RFC8613) uses AEAD algorithms to provide security
  – Need to follow limits in number of encryptions and failed decryptions, before rekeying
  – Excessive use of the same key can enable breaking security properties of the AEAD algorithm*

› (1) AEAD Key Usage Limits in OSCORE
  – Defining appropriate limits for OSCORE, for a variety of algorithms
  – Defining counters for key usage; message processing details; steps when limits are reached
  – Now recommends \((q, v, l) = (2^{20}, 2^{14}, 2^8)\) for AES 128 CCM 8; details in Appendix A

› (2) Defined Key Update for OSCORE (KUDOS) — MAIN FOCUS OF TODAY
  – Loosely inspired by Appendix B.2 of OSCORE
  – Goal: Renew the Master Secret and Master Salt; derive new Sender/Recipient keys from those
  – Achieves Forward Secrecy

*See also draft-irtf-cfrg-aead-limits-03
Key Update Recap

› Method for rekeying OSCORE
  – Key Update for OSCORE (KUDOS)
  – Client and server exchange nonces R1 and R2
  – `UpdateCtx()` function for deriving new OSCORE Security Context using the nonces

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

<- 1 byte -> <----- s bytes --------> <- 1 byte -> <----- x bytes ----->
| s (if any) | kid context (if any) | x (if any) | id detail (if any) |
```

Figure 3: The OSCORE option value, including 'id detail'
Key Update without FS (1/2)

- Added alternative KUDOS mode without Forward Secrecy in Appendix E
  - Initially raised on the CoRE mailing list in [1]
  - It allows stateless key update on loss of state (e.g., rebooting)
    - Needed for constrained devices that cannot store information to persistent memory

- Extension to KUDOS, enabling the selection of a no-FS mode through a new bit ‘p’
  - ‘p’ set to 0 ==> run KUDOS in FS mode (original mode)
    - Devices capable of writing to persistent memory should initiate the procedure with ‘p’ set to 0
  - ‘p’ set to 1 ==> run KUDOS in no-FS mode

- New concepts defined
  - Latest Master Secret and Latest Master Salt
    - From the latest derived OSCORE Security; should be stored on disk by a device capable to do so
  - Bootstrap Master Secret & Bootstrap Master Salt
    - If provisioned they are stored on disk, and they are never changed by the device

[1] https://mailarchive.ietf.org/arch/msg/core/EL0yHxQrP2DQwHxo6ojnQedvFbY/
Key Update without FS (2/2)

› Running KUDOS in no-FS mode
  – Before starting KUDOS, the current OSCORE Context CTX_OLD is modified to ensure that
    › Master Secret = Bootstrap Master Secret ; Master Salt = Bootstrap Master Salt
  – Thus forward secrecy is sacrificed, but all other properties of KUDOS remain!

› This mode of KUDOS requires that both peers have Bootstrap Master Secret/Salt

› Agreed downgrading of mode is possible
  – If the initiator sets 'p' to 0, the responder might be unable continue (if it cannot write to disk)
    › Server responder: Return a protected 5.03 error response to Request #1, with 'p' set to 1
    › Client responder: Send a protected Request #2, with 'p' set to 1
    › In either case, abort KUDOS
  – Then, the initiator may retry with 'p' set to 1, to support the best possible common thing

› Reasonable approach? Comments? Good to move to the draft main body?
Keeping Observations (1/2)

› Scenario description:
1. The client starts an observation Obs1 by sending a request Req1 with req_piv X
2. The two peers run KUDOS and reset their Sender Sequence Number (SSN) to 0.
3. Later on, while Obs1 is still ongoing, the client sends a new request Req2 also with req_piv X. This is not necessarily an observation request.

› Problem: A notification sent by the server for Obs1 or a response to Req2 would both cryptographically match against Req1 and Req2

› Now Appendix C defines
  – A method for “long-jumping” beyond PIVs already in use for observations (more on next slide)
  – A new bit ‘b’ to signal interest in keeping observations
    › If there is no mutual interest, delete observations after key update
Keeping Observations (2/2)

› "Long-jumping" method
  – When wishing to send a first request after a KUDOS execution, the client determines the PIV* as the highest req_piv among all the ongoing observations.
  – The client updates its SSN to be (PIV* + 1)

› Issue: Client needs explicit confirmation from server to remove an ongoing observation
  – What if the client cannot get this confirmation?
  – A peer maintains a counter EPOCH for each ongoing observation it participates in, incrementing the EPOCH for every KUDOS execution
  – When EPOCH reaches MAX_EPOCH (same for both peers) the associated observation is deleted by both peers

› MAX_EPOCH
  – Need good default value to recommend

Comments?
Admit negotiation of MAX_EPOCH in KUDOS?
Appendix D defines a method to update peers’ Sender and Recipient IDs
   – Based on earlier discussions on mailing list [1][2]

Properties
   – Each peer specifies its own new Recipient ID (similar to EDHOC)
   – Accepting to update the Sender/Recipient IDs is optional; explicit confirmation is needed
   – **This procedure can be embedded in a KUDOS execution or run standalone**
   – Possible for both client and server to initiate this procedure
   – Changing IDs practically triggers derivation of new OSCORE key material
   – Must **not** be done immediately following a reboot (e.g., KUDOS must be run first)

[1] https://mailarchive.ietf.org/arch/msg/core/GXsKO4wKdt3RTZnQZxOzRdIG9QI/
[2] https://mailarchive.ietf.org/arch/msg/core/CiwcSF0BUVxDas8BpgT0WY1yQrY/
Renew Sender/Recipient IDs (2/3)

- Defined new CoAP Option to carry the desired Recipient ID
  - Proposed option number 24 (00011000)
  - The option value is the selected new Recipient ID of the message sender
  - The peer selects and offers a free Recipient ID for the used ID Context
  - Class E option for OSCORE processing

<table>
<thead>
<tr>
<th>No.</th>
<th>C</th>
<th>U</th>
<th>N</th>
<th>R</th>
<th>Name</th>
<th>Format</th>
<th>Length</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Recipient-ID</td>
<td>opaque</td>
<td>0-7</td>
<td>(none)</td>
</tr>
</tbody>
</table>

C=Critical, U=Unsafe, N=NoCacheKey, R=Repeatable
Renew Sender/Recipient IDs (3/3)

Client (initiator)

CTX_A {
  SID = 1
  RID = 0
}

Protect with CTX_A

Request #1

OSCORE Option: ...
   kid: 1
   Encrypted_Payload {
     RecipientID: 42
     ... Application Payload
   }

Response #1

Verify with CTX_A

OSCORE Option: ...
   Encrypted_Payload {
     Recipient-ID: 78
     ... Application Payload
   }

Server (responder)

CTX_A {
  SID = 0
  RID = 1
}

Verify with CTX_A

Protect with CTX_A

CTX_B {
  SID = 78
  RID = 42
}

CTX_B {
  SID = 42
  RID = 78
}
Renew Sender/Recipient IDs (3/3)

Client (initiator)

<table>
<thead>
<tr>
<th>CTX_A</th>
<th>SID = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RID = 0</td>
</tr>
</tbody>
</table>

Request #1

Protect with CTX_A

OSCORE Option: ..., kid:1
Encrypted_Payload {
  RecipientID: 42
  ... Application Payload
}

Response #1

Verify with CTX_A

OSCORE Option: ...
Encrypted_Payload {
  Recipient-ID: 78
  ... Application Payload
}

Server (responder)

<table>
<thead>
<tr>
<th>CTX_A</th>
<th>SID = 0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RID = 1</td>
</tr>
</tbody>
</table>

Request #2

Protect with CTX_B

OSCORE Option: ..., kid:78
Encrypted_Payload {
  ... Application Payload
}

Response #2

Verify with CTX_B

 OSCORE Option: ...
Encrypted_Payload {
  ... Application Payload
}

Discard CTX_A

Request #3

Protect with CTX_B

OSCORE Option: ..., kid:78
Encrypted_Payload {
  ... Application Payload
}

Verify with CTX_B

Discard CTX_A

Objections? Alternatives?
Alternatives for signaling

› Currently 3 bits are defined
  – "ID Detail Flag", 'd'
    › Signals inclusion of ID Detail in OSCORE option
  – "No Forward Secrecy", 'p'
    › Signals the use of the no-FS mode
  – "Preserve Observations", 'b'
    › Signals preservation of CoAP Observations

› Where to put bits ‘b’ & ‘p’, and integrity protect them?
  – In the 1 byte 'x' following 'kid context', originally encoding the size of 'id detail'
  – Recommended size of nonces R1 & R2 (carried in ‘id detail’) is 8 bytes → Number of bits available in the ‘x’ byte is still sufficient to indicate the size of 'id detail'
  – The ‘x’ value is taken as input in the derivation of the new OSCORE Security Context

![Diagram of the OSCORE option value, including 'id detail'](image)

Comments?
Summary and next steps

› Latest updates
  – Suggested key update **without forward secrecy** (Appendix E)
  – Suggested method for **preserving observations** across key updates (Appendix C)
  – Suggested procedure to **update OSCORE Sender/Recipient IDs** (Appendix D)
  – Proposed alternative placement for signaling bits
  – Improvements in message processing
  – Optional storing optimization for ‘count_q’ (Appendix B)

› Address open points and issues – Feedback is welcome!
  – Improve the suggestions above, move to document body
  – Clarify which KUDOS messages can contain actionable payload

› Implementation
  – To build on existing implementation of OSCORE in Java Californium
Thank you!

Comments/questions?

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

- Defined a new method for rekeying OSCORE
  - Key Update for OSCORE (KUDOS)
  - Client and server exchange nonces R1 and R2
  - `UpdateCtx()` function for deriving new OSCORE Security Context using the nonces

- Properties
  - Can be initiated by either the client or server
  - Completes in one round-trip (after that, the new Security Context can be used)
  - Only one intermediate Security Context is derived
  - The ID Context does not change
  - Robust and secure against peer rebooting
  - Compatible with prior key establishment using the EDHOC protocol
  - NEW: Mode with PFS (stateful) and without PFS (stateless)
  - NEW: Possibility to update Recipient/Sender IDs

---

IETF 113 | CoRE WG | 2022-03-25 | Page 15
OSCORE Option update

OSCORE Option: defined the use of flag bit 1 to signal presence of flag bits 8-15

Defined flag bit 15 -- 'd' -- to indicate:
- This is a OSCORE key update message
- "id detail" is specified (length + value); used to transport a nonce for the key update

![Diagram of OSCORE option value]

Figure 3: The OSCORE option value, including 'id detail'
Key limits (1/3)

› Recap on AEAD limits
  – Discussed in draft-irtf-cfrg-aead-limits-03
  – Limits key use for encryption (q) and invalid decryptions (v)
  – This draft defines fixed values for ‘q’, ‘v’, and ‘l’ and from those calculate CA & IA probabilities
    › IA & CA probabilities must be acceptably low

› Now explicit size limit of protected data to be sent in a new OSCORE message
  – The probabilities are influenced by ‘l’, i.e., maximum message size in cipher blocks
  – Implementations should not exceed ‘l’, and it has to be easy to avoid doing so
  – New text: the total size of the COSE plaintext, authentication Tag, and possible cipher padding for a message may not exceed the block size for the selected algorithm multiplied with ‘l’

› New table (Figure 3) showing values of ‘l’ not just in cipher blocks but actual bytes
Increased value of ‘l’ (message size in blocks) for algos except AES_128_CCM_8
- Increasing ‘l’ from $2^8$ to $2^{10}$ should maintain secure CA and IA probabilities
- draft-irtf-cfrg-aead-limits mentions aiming for CA & IA lower than to $2^{-50}$
  - They have added a table in that document with calculated ‘q’ and ‘v’ values

$q = 2^{20}, v = 2^{20},$ and $l = 2^{10}$

<table>
<thead>
<tr>
<th>Algorithm name</th>
<th>IA probability</th>
<th>CA probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>AEAD_AES_128_CCM</td>
<td>$2^{-64}$</td>
<td>$2^{-66}$</td>
</tr>
<tr>
<td>AEAD_AES_128_GCM</td>
<td>$2^{-97}$</td>
<td>$2^{-89}$</td>
</tr>
<tr>
<td>AEAD_AES_256_GCM</td>
<td>$2^{-97}$</td>
<td>$2^{-89}$</td>
</tr>
<tr>
<td>AEAD_CHACHA20_POLY1305</td>
<td>$2^{-73}$</td>
<td>-</td>
</tr>
</tbody>
</table>

Intent is to increase 'q', 'v' and/or 'l' further. Should we?
- Since we are well below $2^{-50}$ for CA & IA currently
Key limits (3/3)

- Updated table of ‘q’, ‘v’ and ‘l’ for AES_128_CCM_8
  - Added new value for ‘v’, still leaving CA and IA less than $2^{\text{-50}}$
  - Is it ideal to aim for CA & IA close to $2^{\text{-50}}$ as defined in the CRFG document?

<table>
<thead>
<tr>
<th>'q', 'v' and 'l'</th>
<th>IA probability</th>
<th>CA probability</th>
<th>'q', 'v' and 'l'</th>
<th>IA probability</th>
<th>CA probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^{20}$, $v=2^{20}$, $l=2^{8}$</td>
<td>$2^{\text{-44}}$</td>
<td>$2^{\text{-70}}$</td>
<td>$2^{20}$, $v=2^{20}$, $l=2^{6}$</td>
<td>$2^{\text{-44}}$</td>
<td>$2^{\text{-74}}$</td>
</tr>
<tr>
<td>$2^{15}$, $v=2^{20}$, $l=2^{8}$</td>
<td>$2^{\text{-44}}$</td>
<td>$2^{\text{-80}}$</td>
<td>$2^{15}$, $v=2^{20}$, $l=2^{6}$</td>
<td>$2^{\text{-44}}$</td>
<td>$2^{\text{-84}}$</td>
</tr>
<tr>
<td>$2^{10}$, $v=2^{20}$, $l=2^{8}$</td>
<td>$2^{\text{-44}}$</td>
<td>$2^{\text{-90}}$</td>
<td>$2^{10}$, $v=2^{20}$, $l=2^{6}$</td>
<td>$2^{\text{-44}}$</td>
<td>$2^{\text{-94}}$</td>
</tr>
<tr>
<td>$2^{20}$, $v=2^{15}$, $l=2^{8}$</td>
<td>$2^{\text{-49}}$</td>
<td>$2^{\text{-70}}$</td>
<td>$2^{20}$, $v=2^{15}$, $l=2^{6}$</td>
<td>$2^{\text{-49}}$</td>
<td>$2^{\text{-74}}$</td>
</tr>
<tr>
<td>$2^{15}$, $v=2^{15}$, $l=2^{8}$</td>
<td>$2^{\text{-49}}$</td>
<td>$2^{\text{-80}}$</td>
<td>$2^{15}$, $v=2^{15}$, $l=2^{6}$</td>
<td>$2^{\text{-49}}$</td>
<td>$2^{\text{-84}}$</td>
</tr>
<tr>
<td>$2^{10}$, $v=2^{15}$, $l=2^{8}$</td>
<td>$2^{\text{-49}}$</td>
<td>$2^{\text{-90}}$</td>
<td>$2^{10}$, $v=2^{15}$, $l=2^{6}$</td>
<td>$2^{\text{-49}}$</td>
<td>$2^{\text{-94}}$</td>
</tr>
<tr>
<td>$2^{20}$, $v=2^{14}$, $l=2^{8}$</td>
<td>$2^{\text{-50}}$</td>
<td>$2^{\text{-70}}$</td>
<td>$2^{20}$, $v=2^{14}$, $l=2^{6}$</td>
<td>$2^{\text{-50}}$</td>
<td>$2^{\text{-74}}$</td>
</tr>
<tr>
<td>$2^{15}$, $v=2^{14}$, $l=2^{8}$</td>
<td>$2^{\text{-50}}$</td>
<td>$2^{\text{-80}}$</td>
<td>$2^{15}$, $v=2^{14}$, $l=2^{6}$</td>
<td>$2^{\text{-50}}$</td>
<td>$2^{\text{-84}}$</td>
</tr>
<tr>
<td>$2^{10}$, $v=2^{14}$, $l=2^{8}$</td>
<td>$2^{\text{-50}}$</td>
<td>$2^{\text{-90}}$</td>
<td>$2^{10}$, $v=2^{14}$, $l=2^{6}$</td>
<td>$2^{\text{-50}}$</td>
<td>$2^{\text{-94}}$</td>
</tr>
</tbody>
</table>
"Long-Jumping"
“Skipping”