Stand-in Key Identifier and Encrypted Partial IV in the Constrained Application Protocol (CoAP) OSCORE Option

draft-tiloca-core-oscore-piv-enc-01

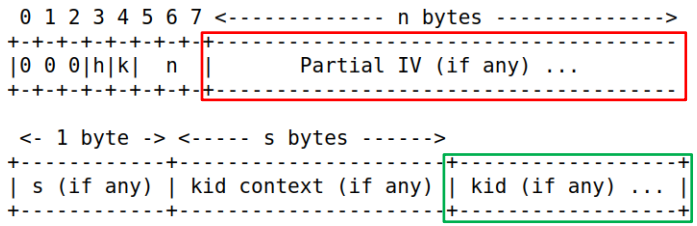
Marco Tiloca, RISE
John Preuß Mattsson, Ericsson
Rikard Höglund, RISE
Göran Selander, Ericsson

IETF 124 meeting – Montreal – November 7th, 2025

Motivation

- **OSCORE (RFC 8613) protects CoAP messages end-to-end at the application layer**
  - Protected messages convey the CoAP OSCORE option, which includes:
  - A ”Partial IV” field, i.e., the sequence number of the sender
  - A “kid” field, i.e., the OSCORE Sender ID of the sender

- **The OSCORE option is not encrypted**
  - Visible for enabling the message processing at the recipient



- **“Partial IV” and “kid” are exposed in plaintext**
  - Possible to infer behavioral patterns of endpoints
  - Possible to track endpoints across different network paths

- **Encrypting “Partial IV” was mentioned at [1]**
  - Possible future update to consider
  - The mechanics has to be simple and efficient
  - Not much worth it if nothing is done for the “kid” values (encryption or update)

[1] https://github.com/core-wg/oscore/issues/263

Contribution

- **Document intended to update RFC 8613**

- **Method to obfuscate fields of the OSCORE option**
  - Encrypt the “Partial IV” field – Same construct of version -00
  - Overwrite the “kid” field with a stand-in, ephemeral identifier ***NEW***

- **Features**
  - Simple and lightweight construct
  - No in-band signaling in OSCORE-protected messages
  - With minor adaptations, also applicable to Group OSCORE [2]

- **Intended as a pluggable “add-on”**
  - Its use is indicated as a property of the specific OSCORE Security Context
  - Establishing the Security Context includes setting whether to use this method or not
    - Without explicit setting, this method is by default not used
  - If used, the “Partial IV” and “kid” are obfuscated in all messages where they are present

[2] https://datatracker.ietf.org/doc/draft-ietf-core-oscore-groupcomm/

Obfuscation Key

- **One additional symmetric key, established with the OSCORE Security Context**
  - Stored in the Common Context
  - Same derivation construct used for the Sender/Recipient Keys

Same as in version -00, except for renaming

- **Derivation based on HKDF**
  - See Section 3.2.1 of RFC 8613

- **Obfuscation Key = HKDF(salt, IKM, info, L)**
  - 'salt' = Master Salt
  - 'IKM '= Master Secret
  - 'info'
    - 'id' = empty byte string
    - 'id_context' = ID Context from the Common Context
    - 'alg_aead' = encryption algorithm from the Common Context
    - 'type' = "OBFKey"
    - 'L' = length in bytes of the algorithm in 'alg_aead'
  - 'L' = like 'L' within 'info'

```
info = [
  id : bstr,
  id_context : bstr / nil,
  alg_aead : int / tstr,
  type : tstr,
  L : uint,
]
```

Sender side (1/2)

After OSCORE protection:

1. **Compose SAMPLE_1**
   - First N bytes of the CoAP payload in the protected message
   - N = min(LENGTH, 16), with LENGTH ≥ 9 being the payload size in bytes

Same as in version -00, except for renaming

2. **Compose INPUT_1 (padded SAMPLE_1)**
   - If SAMPLE_1 is less than 16 bytes in size, INPUT_1 takes SAMPLE_1 left-padded with zeroes to exactly 16 bytes
   - If SAMPLE_1 is 16 bytes in size, INPUT_1 takes SAMPLE_1

3. **Compute PIV_KEYSTREAM**
   - PIV_KEYSTREAM = AES-ECB(ENC_KEY, INPUT_1)
     - AES in ECB mode [3]; ENC_KEY is the Obfuscation Key from the Common Context; INPUT_1 from Step 2

4. **Compute ENC_PIV, by XORing:**
   - The first Q bytes of PIV_KEYSTREAM from Step 3; and
   - The plain content of the “Partial IV” field (Q bytes in size)
     - Note: Q ≥ 3; the Sender Sequence Number used must be at least 65536

5. **Overwrite the “Partial IV” field in the OSCORE option with ENC_PIV from Step 4**

[3] https://doi.org/10.6028/NIST.FIPS.197-upd1

Sender side (2/2)

6. Compose INPUT_2

- Take INPUT_1 from Step 2 (16-byte padded payload sample)
- INPUT_2 takes INPUT_1 with the last bit negated

Ok, this is new!

7. Compute KID_KEYSTREAM

- KID_KEYSTREAM = AES-ECB(ENC_KEY, INPUT_2)
  - AES in ECB mode [3]; ENC_KEY is the Obfuscation Key from the Common Context; INPUT_2 from Step 6

8. Compute the 2-byte STAND_IN_KID value, by XORing:

- The first 2 bytes of KID_KEYSTREAM from Step 7; and
- The first 2 bytes of LATEST_PIV, which is either:
  - ENC_PIV from Step 4, if this OSCORE option includes "Partial IV"; or otherwise
  - The value from "Partial IV" of the OSCORE option in the corresponding request
    - This message is a response with "kid" but without "Partial IV"
    - Unlikely in OSCORE, but typical in Group OSCORE

9. Overwrite the "kid" field in the OSCORE option with STAND_IN_KID from Step 8

- This might alter the length of the option value; if so "Option Length" must be updated accordingly

[3] https://doi.org/10.6028/NIST.FIPS.197-upd1

Recipient side

Who's the sender? What Security Context should I use? Is this OSCORE option obfuscated at all?

- **At first, assume that the OSCORE option is not obfuscated**
  - As usual, try to retrieve a "vanilla" Security Context that does not include an Obfuscation Key
  - If any is found, use it to decrypt and verify the message
  - If none is found or no decryption succeeds, assume obfuscation and continue

- **Inspect each Security Context CTX that does include an Obfuscation Key**

- **Revert the obfuscation performed by the sender**
  - Compose INPUT_2 and compute KID_KEYSTREAM using the Obfuscation Key in CTX
  - XOR KID_KEYSTREAM and LATEST_PIV (i.e., the latest ENC_PIV)
  - If the result is different from what is in the "kid" field, try another Security Context
  - If the result is equal to what is in the "kid" field, CTX is (likely) the right Security Context
    - Compute PIV_KEYSTREAM and restore the plain "Partial IV" field
    - Use CTX to decrypt and verify the message
    - If decryption fails, try with another Security Context
    - If decryption succeeds… it's done!

AES-ECB encryptions are lightweight

AND

We don't expect several Security Contexts to check in the first place

What else?

- **Processing with OSCORE:**
  - The fields “s” and “kid context” are not included in the OSCORE option
- **Section 3.3 – Special cases for OSCORE**
  - EDHOC + OSCORE request [4] --- That particular request cannot be obfuscated
  - Key update for OSCORE (KUDOS) [5] --- Use the Obfuscation Key from the Security Context CTX_OLD to update
  - 6TiSCH and CoJP [6] --- The fields “s” and “kid context” are expected …
    - The sender sets “s” to 0 and removes “kid context”; the recipient rebuilds those from the Security Context
- **Section 4 – Group OSCORE**
  - The Security Context is retrieved as usual using “kid context”; the group as a whole either uses this method or not
  - The inspection process goes through the Recipient Contexts of that Security Context
  - The Obfuscation Key is used to derive:
    - 1 Obfuscation Sender Key, used by the sender to compute the keystreams
    - 1 Obfuscation Recipient Key per Recipient Context, to compute the keystreams when trying that Recipient Context
  - If no Recipient Context is found, the endpoint might want to dynamically create Recipient Contexts
    - This is still possible, with the usual assistance from the Group Manager

[4] https://datatracker.ietf.org/doc/html/rfc9668
[5] https://datatracker.ietf.org/doc/draft-ietf-core-oscore-key-update/
[6] https://datatracker.ietf.org/doc/html/rfc9031

Next steps

- **Consider stand-in KIDs to be of 3 bytes instead of 2**
  - This greatly reduces the chances of false positives when trying a Security Context
- **Enforce two separate agreements: a) encrypting "Partial IV"; b) obfuscating "kid"**
  - Only if there is agreement on (a), there may be additional agreement on (b) – Thanks Martine!
  - Avoid a less efficient variant of what some use cases achieve differently, e.g., with header compression
- **Add more content in Section 4 about Group OSCORE**
  - Section 4.4 "External Signature Checker"
  - Special case: Deterministic Request for cacheable OSCORE [7]
- **Describe means for coordinating endpoints on using this method or not**
  - For OSCORE: build on the placeholders in Section 5.1
  - For Group OSCORE: build on the placeholders in Section 5.2
  - Related requests for IANA registrations in Section 7
- **Gather (more) implementation experience**
- **Comments are welcome!**

[7] https://datatracker.ietf.org/doc/draft-ietf-core-cacheable-oscore/

Thank you!

Comments/questions?

https://gitlab.com/crimson84/draft-tiloca-core-oscore-piv-enc