Encrypted Partial IV in the Constrained Application Protocol (CoAP) OSCORE Option

draft-tiloca-core-oscore-piv-enc-00

Marco Tiloca, RISE
John Preuß Mattsson, Ericsson

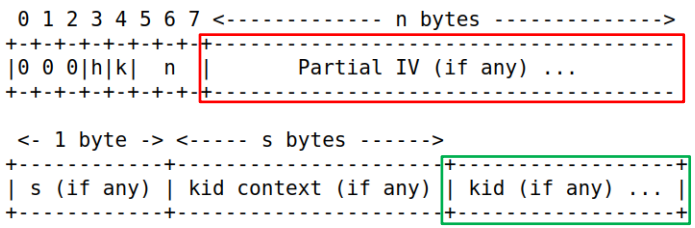
IETF 123 meeting – Madrid – July 22nd, 2025

Motivation

- **OSCORE (RFC 8613) protects CoAP messages end-to-end at the application layer**
  - Protected messages include the CoAP OSCORE option
  - The OSCORE option includes a Partial IV, i.e., the sequence number of the sender

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

- **The Partial IV is exposed in plain**
  - Usable for tracking endpoints across different network paths



- **Encrypting the 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 OSCORE IDs (their encryption or their update)
    - The method in [2] defines how to update OSCORE IDs

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

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

Contribution

- **Document intended to formally update RFC 8613**

- **Method to encrypt the Partial IV in the OSCORE option**
  - Simple and lightweight construct
  - No in-band signaling in OSCORE-protected messages
  - Applicable to both OSCORE and Group OSCORE [3]

- **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 is encrypted in all messages that include one

- **Rationale:**
  - Encrypt as much as reasonably possible
  - Combined with other methods, counteract tracking attacks

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

PIV Encryption 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
- **Derivation based on HKDF**
  - See Section 3.2.1 of RFC 8613
- **PIV Encryption 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' = "PIVEKey"
    - '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,
]
```

Encryption of PIV

After OSCORE protection:

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

2. **Compose INPUT**
   - If SAMPLE is less than 16 bytes in size, INPUT takes SAMPLE left-padded with zeroes to exactly 16 bytes
   - If SAMPLE is 16 bytes in size, INPUT takes SAMPLE

3. **Compute IV_KEYSTREAM**
   - IV_KEYSTREAM = AES-ECB(PIV Encryption Key, INPUT)
     - AES in ECB mode [4]; PIV Encryption Key from the Common Context; INPUT from Step 2

4. **Compute encrypted Partial IV**
   - XOR the plain Partial IV (Q bytes in size) with the first Q bytes of IV_KEYSTREAM

5. **Replace the Partial IV in the OSCORE option with the encrypted Partial IV**

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

Considerations

- **Operational**
  - The purpose would be defeated when transitioning to a different Partial IV size
  - Transitions are likely to happen from 1 byte to 2 bytes, and from 2 bytes to 3 bytes
  - Proposed remedy: if an endpoint expects to send more than 256 messages including the Partial IV, the endpoint should initialize its Sender Sequence Number to 65536 (3-byte Partial IV)
- **Security – Confidentiality of the Partial IV**
  - Ensured against passive adversaries
  - An active adversary has more leeway
    - It could make a guess about the plain Partial IV and then try to confirm the guess
    - Leverage timing side channels on (provoked) actions at the recipient endpoint, e.g.:
      - A message is immediately discarded if detected as a replay, without attempting decryption
      - Different error response are sent, depending on the specific failure at the server
- **Privacy – Impact on endpoint trackability**
  - Helpful against tracking across network migration, if combined with the update of other identifying information
  - E.g., OSCORE IDs [2] and source addressing information (IP address, link layer address)

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

Means to coordinate

- **The use of this method is indicated by the OSCORE Security Context**
  - Establishing the Security Context is a separate task
  - OSCORE is agnostic of how exactly that happened
- **At least point to practical usable means for the peers to coordinate**
  - Placeholders for the “usual suspects” are listed in Section 6
- **For OSCORE**
  - Pre-provisioning
  - Running EDHOC (RFC 9528)
  - Using the OSCORE profile of the ACE framework (RFC 9203)
  - Bootstrapping in OMA Lightweight Machine-to-Machine (LwM2M)
- **For Group OSCORE**
  - Through the OSCORE Group Manager, e.g., the realization based on the ACE framework [5]
  - A CoAP server self-managing the OSCORE group for its group observations [6]

[5] https://datatracker.ietf.org/doc/draft-ietf-ace-key-groupcomm-oscore/

[6] https://datatracker.ietf.org/doc/draft-ietf-core-observe-multicast-notifications/

Next steps

- **Describe means for coordinating endpoints on protecting the Partial IV or not**
  - For OSCORE: build on the placeholders in Section 6.1
  - For Group OSCORE: build on the placeholders in Section 6.2
  - Related requests for IANA registrations

- **Comments are welcome!**

Thank you!

Comments/questions?

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