Clarifications and Updates on using Static Context Header Compression (SCHC) for the Constrained Application Protocol (CoAP)

draft-tiloca-schc-8824-update-01

Marco Tiloca, RISE
Laurent Toutain, IMT Atlantique
Ivan Martinez, Nokia Bell Labs
Ana Minaburo

IETF 117 meeting – San Francisco – July 27th, 2023
Motivation

› RFC 8824 – SCHC compression of CoAP headers (June 2021)
  – Covering all the CoAP options defined at the time (except for Hop-Limit)
  – CoAP messages unprotected or protected end-to-end with OSCORE (RFC 8613)

› Reading RFC 8824 in 2023
  – Some recent CoAP options are missing
  – The CoAP payload marker is always tacitly “not-sent” (i.e., it’s never in the rules)
  – How does this work with CoAP proxies?
    › No processing workflow, no examples
    › Non trivial to “just figure out” in case OSCORE is used

› Errata have been filed for RFC 8824
  – Some can be addressed in a follow-up document
Contribution

› Proposed update to RFC 8824
  – Clarifications on CoAP options Size1, Size2, Proxy-URI, Proxy-Scheme
    › Related to the Erratum at https://www.rfc-editor.org/errata/eid7391
  – Defined SCHC compression for the CoAP option Hop-Limit
  – Defined SCHC compression for recent CoAP options (Echo, Request-Tag, EDHOC)
  – Defined SCHC compression for the updated CoAP option OSCORE
  – Clarified handling of CoAP payload marker 0xFF
    › Both with and without OSCORE end-to-end security
  – Defined SCHC compression with CoAP proxies
    › Both with and without OSCORE end-to-end security

› Approach, design choices, and features of SCHC compression are not changed
Since version -00

› Ana Minaburo joined the author list!

› No changes to the overall scope and spirit of the document

› Terminology for TV in rule descriptions, when MO = “ignore” and CDA = “value-sent”
  – Related to https://www.rfc-editor.org/errata/eid7391
  – Discussed at the SCHC interim meeting in June [1]
  – Resolution: TV is not set, now used consistently

› Improved compression of the Hop-Limit option (RFC 8768)
  – Originally only (TV not set; MO = “ignore”; CDA = “value-sent”)
  – Admitted also (TV = 16; MO = “equal”; CDA = “not-sent”) – 16 is the default value

[1] https://datatracker.ietf.org/meeting/interim-2023-schc-01/session/schc
Since version -00

› Improved compression of the Echo option (RFC 9175)
  – Originally only (TV not set; MO = “ignore”; CDA = “value-sent”)
  – Admitted also (MO = “MSB”; CDA = “LSB”)
    › Useful if the server generates the Echo value with a persistent counter
    › Usable until passing the threshold value that produces an MSB-matching

› Improved compression of the Request-Tag option (RFC 9175)
  – Originally only (TV not set; MO = “ignore”; CDA = “value-sent”)
  – Admitted also (MO = “match-mapping”; CDA = “mapping-sent”)
    › Useful if a pre-defined set of Request-Tag values used by the client is known
Since version -00

› Use of SCHC with CoAP proxies
  – Most of the new content in version -01
  – Clarifications and detailed examples

› Same rationale outlined in version -00
  – Without OSCORE: the SCHC Rules are hop-by-hop
  – With OSCORE: the inner processing is end-to-end; the outer processing is hop-by-hop

› Addressed comment received at IETF 116 from Pascal, see Section 5.0
  – SCHC is not necessarily used between each pair of hops
  – E.g., used between origin client and proxy; not used between proxy and origin server
Since version -00

- Detailed examples of message exchange through a CoAP proxy, see Section 6
  - Same style of the examples in RFC 8824
  - Focus on compression/decompression of CoAP headers
  - One example without OSCORE, one example with OSCORE

- Same example topology from Section 2 of RFC 8824
  - A Device over LPWAN communicates with a network gateway (NGW)
  - The Device talks to an Application Server on the Internet through the NGW
  - The CoAP proxy is co-located at the NGW

- The original CoAP request includes
  - The Uri-Host option \(\rightarrow\) (TV not set; MO = “ignore”; CDA = “value-sent”)
  - The Uri-Path option \(\rightarrow\) (TV = “temperature”; MO = “equal”; CDA = “not-sent”)
  - The Proxy-Scheme option \(\rightarrow\) (TV = “coap”; MO = “equal”; CDA = “not-sent”)

Since version -00

Example without OSCORE, see Section 6.1

- Device → Proxy : 35 bytes are compressed to 14 bytes
- Proxy → Application Server : 29 bytes are compressed to 14 bytes
- Application Server → Proxy : 10 bytes are compressed to 7 bytes
- Proxy → Device : 10 bytes are compressed to 7 bytes

Example with OSCORE, see Section 6.2

- Device → Proxy : 39 bytes are outer-compressed to 25 bytes
- Proxy → Application Server : 33 bytes are outer-compressed to 25 bytes
- Application Server → Proxy : 21 bytes are outer-compressed to 16 bytes
- Proxy → Device : 21 bytes are outer-compressed to 16 bytes (*)

(*) Typo in Figure 25: s/Compressed message length: 15 bytes/Compressed message length: 16 bytes
Open point to confirm

› Should we **switch to a bis document** that obsoletes RFC 8824?

› First suggested at IETF 116 by Éric Vyncke. Since then:
  – John P. Mattsson supported it at IETF 116, to have all the information in one document
  – Pascal referred to this as a good prospect, during the SCHC interim meeting in June
  – The authors think that the idea is good and appropriate to pursue
  – No objections have been heard

› **Advantages**
  – Clarity, since all the content is in one place (John’s point)
  – Clean, streamlined way to address the Errata filed in March for RFC 8824
  – Opportunity to fix anything else we find, e.g.:
    › The examples in RFC 8824 use Option Number 21 (instead of 9) for the OSCORE option

Any objection to switching to a bis document?
Summary and next steps

› Update (or obsolete?) RFC 8824 - Use of SCHC for CoAP messages
  – Compression of CoAP options recently defined or extended
  – Clarifications on SCHC handling of the CoAP payload marker 0xFF
  – SCHC compression in the presence of CoAP proxies (with or without OSCORE)

› Next steps for version -02
  – If agreed, convert the draft into a bis document that obsoletes RFC 8824
  – Revise and add the YANG data model for compression of new CoAP options
    › https://gitlab.com/crimson84/draft-tiloca-schc-8824-update/-/blob/main/ietf-schc-coap@2023-03-07.yang
  – To consider: defining/revising a CoAP option → defining/revising its SCHC processing

› Comments and feedback are welcome!
Thank you!

Comments/questions?

https://gitlab.com/crimson84/draft-tiloca-schc-8824-update
Backup
CoAP options

› **Hop-Limit (RFC 8768)**
  - TV: not set; MO: "ignore"; CDA: "value-sent"
  - TV: 16; MO: "equal"; CDA: "not-sent"

› **Echo (RFC 9175)**
  - TV: not set; MO: "ignore"; CDA: "value-sent"
  - MO: "MSB"; CDA: "LSB"

› **Request-Tag (RFC 9175)**
  - TV: not set; MO: "ignore"; CDA: "value-sent"
  - MO: "match-mapping"; CDA: "mapping-sent"

› **Q-Block1 and Q-Block2 (RFC 9177)**
  - TV: not set; MO: "ignore"; CDA: "value-sent"

› **EDHOC (draft-ietf-core-oscore-edhoc)**
  - TV: empty; MO: "equal"; CDA: "not-sent"
OSCORE option

› OSCORE (RFC 8613)
  – Updated by related documents

› Defined new flag bits in the first byte
  – draft-ietf-core-oscore-key-update
  – draft-ietf-core-oscore-groupcomm

› Defined new, second flag byte
  – draft-ietf-core-oscore-key-update

› Defined new fields ‘x’ and ‘nonce’
  – draft-ietf-core-oscore-key-update
**CoAP payload marker 0xFF**

- **Never included in a compressed message**
  - Neither in the compression residue, ...
  - ... nor before the payload (if any)
  - (... nor in the compression rules!)

- **Compression result without end-to-end security**
  - Rule ID | Compression residue | Payload

- **Compression result with end-to-end security**
  - Rule ID' | Compression residue' | OSCORE ciphertext
  - Following the second, outer compression
Compression with CoAP proxies

› **Without end-to-end security**
  – Compression based on hop-by-hop rules
  – C compresses for P, then P compresses for S

› **With end-to-end security**
  – The inner compression is end-to-end
  – The outer compression is hop-by-hop
  – @C: inner C<->S compression, then
    outer C<->P compression
  – @P: outer C<->P decompression, then
    outer P<->S compression
  – @S: outer P<->S decompression, then
    inner C<->S decompression