OSCORE-SCHC Compression

Authors:
Ricardo Andreasen <randreasen@fi.uba.ar>
Laurent Toutain <Laurent.Toutain@imt-atlantique.fr>
Ana Minaburo <ana@ackl.io>
OSCORE – Main mechanism

- Splits message into an inner Plaintext and an outer OSCORE message

Original CoAP message

Plaintext

Payload

Encryption

Ciphertext

Outer header

OSCORE opt
SCHC Compression

- Idea: compress both – Inner and Outer SCHC Compression

Original CoAP message

Payload

Compressed Plaintext

Compressed OSCORE Message

Ciphertext

OSCORE Message

Inner SCHC Compression

Outer SCHC Compression

Plaintext
Inner Compression

<table>
<thead>
<tr>
<th>Field</th>
<th>DI</th>
<th>Target Value</th>
<th>MO</th>
<th>CDA</th>
<th>Sent bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>up</td>
<td>1</td>
<td>equal</td>
<td>Not-sent</td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td>dw</td>
<td>[69, 132]</td>
<td>Match-map</td>
<td>Match-sent</td>
<td>c</td>
</tr>
<tr>
<td>Uri-Path</td>
<td>up</td>
<td>temperature</td>
<td>equal</td>
<td>Not-sent</td>
<td></td>
</tr>
<tr>
<td>Option-End</td>
<td>dw</td>
<td>0xFF</td>
<td>equal</td>
<td>Not-sent</td>
<td></td>
</tr>
</tbody>
</table>
## Outer Compression

<table>
<thead>
<tr>
<th>Field</th>
<th>DI</th>
<th>Target Value</th>
<th>MO</th>
<th>CDA</th>
<th>Sent bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>bi</td>
<td>01</td>
<td>equal</td>
<td>Not-sent</td>
<td></td>
</tr>
<tr>
<td>Type up</td>
<td>0</td>
<td>equal</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type dw</td>
<td>2</td>
<td>equal</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TKL bi</td>
<td>1</td>
<td>equal</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code up</td>
<td>2</td>
<td>equal</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code dw</td>
<td>68</td>
<td>equal</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MID bi</td>
<td>0000</td>
<td>MSB(12)</td>
<td>LSB</td>
<td>MMMM</td>
<td></td>
</tr>
<tr>
<td>Token bi</td>
<td>0x80</td>
<td>MSB(5)</td>
<td>LSB</td>
<td>TTT</td>
<td></td>
</tr>
<tr>
<td>OSCORE_piv up</td>
<td>0x0900</td>
<td>MSB(12)</td>
<td>LSB</td>
<td>PPPPP</td>
<td></td>
</tr>
<tr>
<td>OSCORE_kid dw</td>
<td>b’client’</td>
<td>MSB(52)</td>
<td>LSB</td>
<td>KKKK</td>
<td></td>
</tr>
<tr>
<td>OSCORE_piv dw</td>
<td>b’’</td>
<td>equal</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Option-End dw</td>
<td>0xFF</td>
<td>equal</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The diagram illustrates the structure of the outer compression process, with fields such as Version, Type, TKL, MID, Token, and additional OSCORE options. The table provides specific values for these fields, indicating whether they are sent (`sent`) or not (`Not-sent`). The structure also highlights the relationship between different parts of the compression, with arrows pointing upwards indicating the flow from `Ruleid` to `Residue` and then to Ciphertext.
OSCORE option

\[
\begin{array}{c|c|c|c}
0 & 0 & 0 & h \ k \ n \\
\hline
s & \text{kid context} & \hline
& \text{kid} & \hline
\end{array}
\]

- **OSCORE_piv**
  - 0 0 0 h k n
  - piv

- **OSCORE_kid_context**
  - s \ kid context

- **OSCORE_kid**
  - kid

LSB
OSCORE option: example

OSCORE_piv

OSCORE_kid

TV = 0x0900, MO = MSB(12), CDA= LSB

TV = b’tclient\x00’, MO = MSB(52), CDA= LSB
GET with and without OSCORE

Compressed message (protected):
================================================
0x14d2527a6023d9f2ee6434
0x00 = Rule ID

Compression residue:
0b 0001 010 011 0100 -> 14 bits
   mid  tkn  piv  kid

Payload
0x949e9808f67cbb990d

Original msg length: 17
Protected msg length: 25
Compressed msg length: 12

Compressed message (no OSCORE):
================================================
0x0114
0x01 = Rule ID

Compression residue:
0b 0001 010 011 0100
> 14 bits

mid  tkn  piv  kid

Original msg length: 17
Compressed msg length: 2

So cost of security was 10
CONTENT with and without OSCORE

Compressed message (protected):

0x0014ce9840307f91ef8cb05b90d2981e
0x00 = Rule ID

Compression residue:
0b 0001 010 -> 7 bits
mid  tkn

Payload
0x674c20183fc8f7c6582dc8694c0f

Original msg length: 10
Protected msg length: 22
Compressed msg length: 16

Compressed message (no OSCORE):

0x010a32332043
0x01 = Rule ID

Compression residue:
0b00001010

Payload
0x32332043

Original msg length: 10
Compressed msg length: 6

So cost of security was 10
Annex
OSCORE

OSCORE is an application-layer protection of CoAP using COSE (CoAP Object Signing and Encryption). It provides:

- End-to-end encryption
- Integrity
- Replay protection

This it does by sorting CoAP fields into one of 3 classes:
- Class E: Encrypted and integrity protected
- Class I: Integrity protected
- Class U: Unprotected
CoAP Field Classification

Token: can be seen as a connection in TCP

Options

Options

Options

0xFF

Data
CoAP Field Classification

Token: can be seen as a connection in TCP

Options: Class E (encrypted)

Options: Class I (integrity protected)

Options: Class U (unprotected)

Data
CoAP Field Classification

Token: can be seen as a connection in TCP

Options: Class E (encrypted)
Options: Class I (integrity protected)
Options: Class U (unprotected)

Data

version | type | token size | code  | class | detail | message ID

01 | XX | | | | |
Prepare target OSCORE message

Token: can be seen as a connection in TCP

Options: Class I (integrity protected)

Options: Class U (unprotected)

Object Security Option (OSC)

0xFF

(to be filled with encrypted data)
OSCORE Plaintext

First byte

CoAP Code

Options: Class E (encrypted)

FF

Data (original CoAP message payload)

Options are reordered and re-compressed with delta encoding as per CoAP
OSCORE option: example

Token OSC Resp.

OptionLength = 0 (sends empty O_S option)

→ TV = b'', MO = equal, CDA= not-sent.
Examples: GET - CONTENT

Original message:
===============
0x4101000182bb74656d7065726174757265

Header:
0x4101
01 Ver
00 CON
0001 tkl
00000001 Request Code 1 "GET"

0x0001 = mid
0x82 = token

Options:
0xbb74656d7065726174757265
Option 11: URI_PATH
Value = temperature

Original msg length: 17 bytes.

Original message:
===============
0x6145000182ff32332043

Header:
0x6145
01 Ver
10 ACK
0001 tkl
01000101 Successful Response Code 69 "2.05 Content"

0x0001 = mid
0x82 = token

0xFF Payload marker
Payload:
0x32332043

Original msg length: 10 bytes.
Analysis: GET Request

Original message:
0x4101000182bb74656d7065726174757265

Opt 11: URI_PATH
Value = temperature

OSCORE Plaintext:
0x01bb74656d7065726174757265

Compressed Plaintext:
0x03

OSCORE Message:
0x4102000182d7080903636c69656e74ff949e9808f67cbb990d

Compression Residue:
0b 0001 010 011 0100

Compressed message:
0x0014d2527a6023d9f2ee6434
Analysis: CONTENT Response

CoAP Header
-token
-Payload

0x6145000182ff32332043 Original message

CoAP Code
-mid
-Payload Marker

0x45ff32332043 Plaintext → 0x031919902180 Compressed Plaintext

Empty OSCORE option

0x6144000182d008ff674c20183fc8f7c6582dc8694c0f OSCORE Message

Ciphertext

0b 0001 010 Compression Residue

0x0014ce9840307f91ef8cb05b90d2981e Compressed message

Rule ID

LPWAN@IETF102