Message Size Overhead of CoAP Security Protocols
draft-mattsson-core-security-overhead-02

Abstract

This document analyzes and compares per-packet message size overheads when using different security protocols to secure CoAP. The analyzed security protocols are DTLS 1.2, DTLS 1.3, TLS 1.2, TLS 1.3, and OSCORE. DTLS and TLS are analyzed with and without compression. DTLS are analyzed with two different alternatives for header compression as well as with and without Connection ID.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 15, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of
1. Introduction

This document analyzes and compares per-packet message size overheads when using different security protocols to secure CoAP over UDP [RFC7252] and TCP [I-D.ietf-core-coap-tcp-tls]. The analyzed security protocols are DTLS 1.2 [RFC6347], DTLS 1.3 [I-D.rescorla-tls-dtls13], TLS 1.2 [RFC5246], TLS 1.3 [I-D.ietf-tls-tls13], and OSCORE [I-D.ietf-core-object-security]. The DTLS and TLS record layers are analyzed with and without compression. DTLS are analyzed with two different alternatives ([RFC7400] and [raza-6lo-compressed-dtls]) for header compression as well as with and without Connection ID [I-D.rescorla-tls-dtls-connection-id].
2. Overhead of Security Protocols

To enable comparison, all the overhead calculations in this section use AES-CCM with a tag length of 8 bytes, a plaintext of 6 bytes, and the sequence number ‘05’. This follows the example in [RFC7400], Figure 16.

2.1. DTLS

2.1.1. DTLS 1.2

This section analyzes the overhead of DTLS 1.2 [RFC6347]. The nonce follow the strict profiling given in [RFC7925]. This example is taken directly from [RFC7400], Figure 16.

DTLS 1.2 Record Layer (35 bytes, 29 bytes overhead):
17 fe fd 00 01 00 00 00 00 05 00 16 00 01 00
00 00 00 00 05 ae a0 15 56 67 92 4d ff 8a 24 e4
cb 35 b9

Content type:
17
Version:
fe fd
Epoch:
00 01
Sequence number:
00 00 00 00 00 05
Length:
00 16
Nonce:
00 01 00 00 00 00 00 05
Ciphertext:
ae a0 15 56 67 92
ICV:
4d ff 8a 24 e4 cb 35 b9

DTLS 1.2 gives 29 bytes overhead.

2.1.2. DTLS 1.2 with 6LoWPAN-GHC

This section analyzes the overhead of DTLS 1.2 [RFC6347] when compressed with [RFC7400]. The compression was done with [OlegHahm-ghc].

Note that the compressed overhead is dependent on the parameters epoch, sequence number, and length. The following is only an example.
Note that the sequence number ‘01’ used in [RFC7400], Figure 15 gives an exceptionally small overhead that is not representative.

Note that this header compression is not available when DTLS is exchanged over transports that do not use 6LoWPAN together with 6LoWPAN-GHC.

Compressed DTLS 1.2 Record Layer (22 bytes, 16 bytes overhead):
\[00\ 05\ 00\ 16\ f2\ 0e\ \text{ae\ a0}\ 15\ 56\ 67\ 92\ 4d\ ff\ 8a\ 24\ e4\ cb\ 35\ b9\]

Compressed DTLS 1.2 Record Layer Header and Nonce:
\[00\ 05\ 00\ 16\ f2\ 0e\ \text{ae\ a0}\ 15\ 56\ 67\ 92\]

Ciphertext:
\[\text{ae\ a0}\ 15\ 56\ 67\ 92\]

ICV:
\[4d\ ff\ 8a\ 24\ e4\ cb\ 35\ b9\]

When compressed with 6LoWPAN-GHC, DTLS 1.2 with the above parameters (epoch, sequence number, length) gives 16 bytes overhead.

2.1.3. DTLS 1.2 with raza-6lo-compressed-dtls

This section analyzes the overhead of DTLS 1.2 [RFC6347] when compressed with [raza-6lo-compressed-dtls].

Note that the compressed overhead is dependent on the parameters epoch and sequence number. The following is only an example.

Note that this header compression is not available when DTLS is exchanged over transports that do not use 6LoWPAN together with raza-6lo-compressed-dtls.

Compressed DTLS 1.2 Record Layer (19 bytes, 13 bytes overhead):
\[01\ 00\ 05\ \text{ae\ a0}\ 15\ 56\ 67\ 92\ 4d\ ff\ 8a\ 24\ e4\ cb\ 35\ b9\]

NHC
\[90\]

Compressed DTLS 1.2 Record Layer Header and Nonce:
\[01\ 00\ 05\ \text{ae\ a0}\ 15\ 56\ 67\ 92\]

Ciphertext:
\[\text{ae\ a0}\ 15\ 56\ 67\ 92\]

ICV:
\[4d\ ff\ 8a\ 24\ e4\ cb\ 35\ b9\]

When compressed with raza-6lo-compressed-dtls, DTLS 1.2 with the above parameters (epoch, sequence number) gives 13 bytes overhead.
2.1.4. DTLS 1.3

This section analyzes the overhead of DTLS 1.3 [I-D.rescorla-tls-dtls13]. The only change compared to DTLS 1.2 is that the DTLS 1.3 record layer does not have an explicit nonce.

DTLS 1.3 Record Layer (27 bytes, 21 bytes overhead):
17 fe fd 00 01 00 00 00 00 05 00 0e ae a0 15
56 67 92 4d ff 8a 24 e4 cb 35 b9

Content type:
17
Version:
fe fd
Epoch:
00 01
Sequence number:
00 00 00 00 00 05
Length:
00 0e
Ciphertext:
ae a0 15 56 67 92
ICV:
4d ff 8a 24 e4 cb 35 b9

DTLS 1.3 gives 21 bytes overhead.

2.1.5. DTLS 1.3 with 6LoWPAN-GHC

This section analyzes the overhead of DTLS 1.3 [I-D.rescorla-tls-dtls13] when compressed with [RFC7400] [OlegHahm-ghc].

Note that the overhead is dependent on the parameters epoch, sequence number, and length. The following is only an example.

Note that this header compression is not available when DTLS is exchanged over transports that do not use 6LoWPAN together with 6LoWPAN-GHC.
Compressed DTLS 1.3 Record Layer (20 bytes, 14 bytes overhead):
b0 c3 11 05 00 0e ae a0 15 56 67 92 4d ff 8a 24
e4 cb 35 b9

Compressed DTLS 1.3 Record Layer Header and Nonce:
b0 c3 11 05 00 0e
Ciphertext:
ae a0 15 56 67 92
ICV:
4d ff 8a 24 e4 cb 35 b9

When compressed with 6LoWPAN-GHC, DTLS 1.3 with the above parameters
epoch, sequence number, length) gives 14 bytes overhead.

2.1.6. DTLS 1.3 with raza-6lo-compressed-dtls

This section analyzes the overhead of DTLS 1.3
[I-D.rescorla-tls-dtls13] when compressed with
[raza-6lo-compressed-dtls].

Note that the compressed overhead is dependent on the parameters
epoch and sequence number. The following is only an example.

Note that this header compression is not available when DTLS is
exchanged over transports that do not use 6LoWPAN together with raza-
6lo-compressed-dtls.

Compressed DTLS 1.3 Record Layer (19 bytes, 13 bytes overhead):
90 17 01 00 05 ae a0 15 56 67 92 4d ff 8a 24 e4
cb 35 b9

NHC
90
Compressed DTLS 1.3 Record Layer Header and Nonce:
17 01 00 05
c3 03 05 00 16 f2 0e
Ciphertext:
ae a0 15 56 67 92
ICV:
4d ff 8a 24 e4 cb 35 b9

When compressed with raza-6lo-compressed-dtls, DTLS 1.3 with the
above parameters (epoch, sequence number) gives 13 bytes overhead.
2.2. DTLS with Connection ID

This section analyzes the overhead of DTLS with Connection ID [I-D.rescorla-tls-dtls-connection-id]. The overhead calculations in this section uses Connection ID = ‘42’. DTLS with a Connection ID = ’’ (the empty string) is equal to DTLS without Connection ID.

2.2.1. DTLS 1.2 with Connection ID

This section analyzes the overhead of DTLS 1.2 [RFC6347] with Connection ID [I-D.rescorla-tls-dtls-connection-id].

Note that the overhead is dependent on the parameter Connection ID. The following is only an example.

DTLS 1.2 Record Layer (35 bytes, 29 bytes overhead):
17 fe fd 00 01 00 00 00 00 00 05 42 00 16 00 01
00 00 00 00 00 05 ae a0 15 56 67 92 4d ff 8a 24
e4 cb 35 b9

Content type:
17
Version:
fe fd
Epoch:
00 01
Sequence number:
00 00 00 00 00 05
Connection ID:
42
Length:
00 16
Nonce:
00 01 00 00 00 00 00 05
Ciphertext:
ae a0 15 56 67 92
ICV:
4d ff 8a 24 e4 cb 35 b9

DTLS 1.2 with Connection ID gives 30 bytes overhead.

2.2.2. DTLS 1.2 with Connection ID and 6LoWPAN-GHC

This section analyzes the overhead of DTLS 1.2 [RFC6347] with Connection ID [I-D.rescorla-tls-dtls-connection-id] when compressed with [RFC7400] [OlegHahm-ghc].
Note that the compressed overhead is dependent on the parameters epoch, sequence number, Connection ID, and length. The following is only an example.

Note that the sequence number ‘01’ used in [RFC7400], Figure 15 gives an exceptionally small overhead that is not representative.

Note that this header compression is not available when DTLS is exchanged over transports that do not use 6LoWPAN together with 6LoWPAN-GHC.

Compressed DTLS 1.2 Record Layer (23 bytes, 17 bytes overhead):
```
b0 c3 04 05 42 00 16 f2 0e ae a0 15 56 67 92 4d
ff 8a 24 e4 cb 35 b9
```

Compressed DTLS 1.2 Record Layer Header and Nonce:
```
b0 c3 04 05 42 00 16 f2 0e
```
Ciphertext:
```
ae a0 15 56 67 92
```
ICV:
```
4d ff 8a 24 e4 cb 35 b9
```

When compressed with 6LoWPAN-GHC, DTLS 1.2 with the above parameters (epoch, sequence number, Connection ID, length) gives 17 bytes overhead.

2.2.3. DTLS 1.3 with Connection ID

This section analyzes the overhead of DTLS 1.3 [I-D.rescorla-tls-dtls13] with Connection ID [I-D.rescorla-tls-dtls-connection-id].

Note that the overhead is dependent on the parameter Connection ID. The following is only an example.
DTLS 1.3 Record Layer (28 bytes, 22 bytes overhead):
17 fe fd 00 01 00 00 00 00 05 42 00 0e ae a0
15 56 67 92 4d ff 8a 24 e4 cb 35 b9

Content type:
17
Version:
fe fd
Epoch:
00 01
Sequence number:
00 00 00 00 00 05
Connection ID:
42
Length:
00 0e
Ciphertext:
ae a0 15 56 67 92
ICV:
4d ff 8a 24 e4 cb 35 b9

DTLS 1.3 gives 22 bytes overhead.

2.2.4. DTLS 1.3 with Connection ID and 6LoWPAN-GHC

This section analyzes the overhead of DTLS 1.3
[I-D.rescorla-tls-dtls13] with Connection ID
[I-D.rescorla-tls-dtls-connection-id] when compressed with [RFC7400]
[OlegHahm-ghc].

Note that the overhead is dependent on the parameters epoch, sequence
number, Connection ID, and length. The following is only an example.

Note that this header compression is not available when DTLS is
exchanged over transports that do not use 6LoWPAN together with
6LoWPAN-GHC.

Compressed DTLS 1.3 Record Layer (21 bytes, 15 bytes overhead):
b0 c3 12 05 42 00 0e ae a0 15 56 67 92 4d ff 8a
24 e4 cb 35 b9

Compressed DTLS 1.3 Record Layer Header and Nonce:
b0 c3 12 05 42 00 0e
Ciphertext:
ae a0 15 56 67 92
ICV:
4d ff 8a 24 e4 cb 35 b9
When compressed with 6LoWPAN-GHC, DTLS 1.3 with the above parameters (epoch, sequence number, Connection ID, length) gives 15 bytes overhead.

2.3. TLS

2.3.1. TLS 1.2

This section analyzes the overhead of TLS 1.2 [RFC5246]. The changes compared to DTLS 1.2 is that the TLS 1.2 record layer does not have epoch and sequence number, and that the version is different.

TLS 1.2 Record Layer (27 bytes, 21 bytes overhead):
17 03 03 00 16 00 00 00 00 00 00 00 05 ae a0 15
56 67 92 4d ff 8a 24 e4 cb 35 b9

Content type:
17
Version:
03 03
Length:
00 16
Nonce:
00 00 00 00 00 00 00 05
Ciphertext:
ea a0 15 56 67 92
ICV:
4d ff 8a 24 e4 cb 35 b9

TLS 1.2 gives 21 bytes overhead.

2.3.2. TLS 1.2 with 6LoWPAN-GHC

This section analyzes the overhead of TLS 1.2 [RFC5246] when compressed with [RFC7400] [OlegHahm-ghc].

Note that the overhead is dependent on the parameters epoch, sequence number, and length. The following is only an example.

Note that this header compression is not available when TLS is exchanged over transports that do not use 6LoWPAN together with 6LoWPAN-GHC.
Compressed TLS 1.2 Record Layer (23 bytes, 17 bytes overhead):
05 17 03 03 00 16 85 0f 05 ae a0 15 56 67 92 4d
ff 8a 24 e4 cb 35 b9

Compressed TLS 1.2 Record Layer Header and Nonce:
05 17 03 03 00 16 85 0f 05
Ciphertext:
ae a0 15 56 67 92
ICV:
4d ff 8a 24 e4 cb 35 b9

When compressed with 6LoWPAN-GHC, TLS 1.2 with the above parameters
(epoch, sequence number, length) gives 17 bytes overhead.

2.3.3. TLS 1.3

This section analyzes the overhead of TLS 1.3 [I-D.ietf-tls-tls13].
The change compared to TLS 1.2 is that the TLS 1.3 record layer uses
a different version.

TLS 1.3 Record Layer (27 bytes, 21 bytes overhead):
17 03 01 00 16 00 00 00 00 00 00 00 05 ae a0 15
56 67 92 4d ff 8a 24 e4 cb 35 b9

Content type:
17
Version:
03 01
Length:
00 16
Nonce:
00 00 00 00 00 00 00 00 05
Ciphertext:
ae a0 15 56 67 92
ICV:
4d ff 8a 24 e4 cb 35 b9

TLS 1.3 gives 21 bytes overhead.

2.3.4. TLS 1.3 with 6LoWPAN-GHC

This section analyzes the overhead of TLS 1.3 [I-D.ietf-tls-tls13]
when compressed with [RFC7400] [OlegHahm-ghc].

Note that the overhead is dependent on the parameters epoch, sequence
number, and length. The following is only an example.
Note that this header compression is not available when TLS is exchanged over transports that do not use 6LoWPAN together with 6LoWPAN-GHC.

Compressed TLS 1.3 Record Layer (23 bytes, 17 bytes overhead):
02 17 03 c3 01 16 85 0f 05 ae a0 15 56 67 92 4d
ff 8a 24 e4 cb 35 b9

Compressed TLS 1.3 Record Layer Header and Nonce:
02 17 03 c3 01 16 85 0f 05
Ciphertext:
ae a0 15 56 67 92
ICV:
4d ff 8a 24 e4 cb 35 b9

When compressed with 6LoWPAN-GHC, TLS 1.3 with the above parameters (epoch, sequence number, length) gives 17 bytes overhead.

2.4. OSCORE

This section analyzes the overhead of OSCORE [I-D.ietf-core-object-security].

Note that the overhead is dependent on the included CoAP Option numbers as well as the length of the OSCORE parameters Sender ID and sequence number.

Note that Sender ID = ‘’ (empty string) can only be used by one client per server.

The examples below assume that the original messages does not have payload (note that this does not affect the overhead).

The below calculation Option Delta = ‘9’, Sender ID = ‘’ (empty string), and Sequence Number = ‘05’, and is only an example.
OSCORE Request (19 bytes, 13 bytes overhead):
92 09 05
ff ec ae a0 15 56 67 92 4d ff 8a 24 e4 cb 35 b9

CoAP Option Delta and Length
92
Option Value (flag byte and sequence number):
09 05
Payload Marker
ff
Ciphertext (including encrypted code):
ec ae a0 15 56 67 92
ICV:
4d ff 8a 24 e4 cb 35 b9

The below calculation Option Delta = ‘9’, Sender ID = ‘42’, and Sequence Number = ’05’, and is only an example.

OSCORE Request (20 bytes, 14 bytes overhead):
93 09 05 42
ff ec ae a0 15 56 67 92 4d ff 8a 24 e4 cb 35 b9

CoAP Option Delta and Length
93
Option Value (flag byte, sequence number, and Sender ID):
09 05 42
Payload Marker
ff
Ciphertext (including encrypted code):
ec ae a0 15 56 67 92
ICV:
4d ff 8a 24 e4 cb 35 b9

The below calculation uses Option Delta = ’9’ and is only an example.
OSCORE Response (17 bytes, 11 bytes overhead):
90
ff ec ae a0 15 56 67 92 4d ff 8a 24 e4 cb 35 b9

CoAP Delta and Option Length:
90
Option Value
-
Payload Marker
ff
Ciphertext (including encrypted code):
ec ae a0 15 56 67 92
ICV:
4d ff 8a 24 e4 cb 35 b9

OSCORE with the above parameters gives 13-14 bytes overhead for requests and 11 bytes overhead for responses.

Unlike DTLS and TLS, OSCORE has much smaller overhead for responses than requests.

3. Overhead with Different Parameters

The DTLS overhead is dependent on the parameter Connection ID. The following overheads apply for all Connection IDs with the same length.

The compression overhead (GHC) is dependent on the parameters epoch, sequence number, Connection ID, and length. The following overheads should be representative for sequence numbers and Connection IDs with the same length.

The compression overhead (raza-6lo-compressed-dtls) is dependent on the length of the parameters epoch and sequence number. The following overheads apply for all sequence numbers with the same length.

The OSCORE overhead is dependent on the included CoAP Option numbers as well as the length of the OSCORE parameters Sender ID and sequence number. The following overheads apply for all sequence numbers and Sender IDs with the same length.
## 4. Summary

DTLS 1.2 has quite a large overhead as it uses an explicit sequence number and an explicit nonce. DTLS 1.3, TLS 1.2, and TLS 1.3 have significantly less (but not small) overhead.

Both DTLS compression methods provide very good compression. raza-6lo-compressed-dtls achieves slightly better compression but requires state. GHC is stateless but provides slightly worse compression. As DTLS 1.3 uses the same version number as DTLS 1.2, both GHC and raza-6lo-compressed-dtls work well also for DTLS 1.3.

### Figure 1: Overhead as a function of sequence number
(Connection/Sender ID = '')

<table>
<thead>
<tr>
<th>Sequence Number</th>
<th>'05'</th>
<th>'1005'</th>
<th>'100005'</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTLS 1.2</td>
<td>29</td>
<td>29</td>
<td>29</td>
</tr>
<tr>
<td>DTLS 1.3</td>
<td>21</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>TLS 1.2</td>
<td>21</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>TLS 1.3</td>
<td>21</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>DTLS 1.2 (Raza)</td>
<td>13</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>DTLS 1.3 (Raza)</td>
<td>13</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>DTLS 1.2 (GHC)</td>
<td>16</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>DTLS 1.3 (GHC)</td>
<td>14</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>TLS 1.2 (GHC)</td>
<td>17</td>
<td>18</td>
<td>19</td>
</tr>
<tr>
<td>TLS 1.3 (GHC)</td>
<td>17</td>
<td>18</td>
<td>19</td>
</tr>
<tr>
<td>OSCORE Request</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>OSCORE Response</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
</tbody>
</table>

### Figure 2: Overhead as a function of Connection/Sender ID
(Sequence Number = '05')

<table>
<thead>
<tr>
<th>Connection/Sender ID</th>
<th>''</th>
<th>'42'</th>
<th>'4002'</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTLS 1.2</td>
<td>29</td>
<td>30</td>
<td>31</td>
</tr>
<tr>
<td>DTLS 1.3</td>
<td>21</td>
<td>22</td>
<td>23</td>
</tr>
<tr>
<td>DTLS 1.2 (GHC)</td>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>DTLS 1.3 (GHC)</td>
<td>14</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>OSCORE Request</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>OSCORE Response</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
</tbody>
</table>
The Generic Header Compression (6LoWPAN-GHC) can in addition to DTLS 1.2 handle DTLS 1.3, DTLS with Connection ID, TLS 1.2, and TLS 1.3. The Generic Header Compression (6LoWPAN-GHC) works very well for Connection ID and the overhead seems to increase exactly with the length of the Connection ID (which is optimal). The compression of TLS is not as good as the compression of DTLS (as the static dictionary is more or less a DTLS record layer). Similar compression levels as for DTLS could be achieved also for TLS, but this would require different static dictionaries for each version of TLS (as TLS 1.2 and TLS 1.3 uses different version numbers). GHC works as good for DTLS 1.3 as for DTLS 1.2 as the version number is the same.

raza-6lo-compressed-dtls is not able to handle DTLS with Connection ID or TLS, all extensions requires an updated mechanism.

The header compression is not available when (D)TLS is exchanged over transports that do not use 6LoWPAN together with 6LoWPAN-GHC or raza-6lo-compressed-dtls.

OSCORE has much lower overhead than DTLS and TLS. The overhead of OSCORE is smaller than DTLS over 6LoWPAN with compression, and this small overhead is achieved even on deployments without 6LoWPAN or 6LoWPAN without DTLS compression. OSCORE is lightweight because it makes use of some excellent features in CoAP, CBOR, and COSE.

5. Security Considerations

This document is purely informational.

6. Informative References

[I-D.ietf-core-coap-tcp-tls]

[I-D.ietf-core-object-security]

[I-D.ietf-tls-tls13]
[I-D.rescorla-tls-dtls-connection-id]

[I-D.rescorla-tls-dtls13]

[OlegHahm-ghc]

[raza-6lo-compressed-dtls]


Acknowledgments

The authors want to thank Ari Keraenen, Francesca Palombini, and Goeran Selander for reviewing previous versions of the draft.

All 6LoWPAN-GHC compression was done with [OlegHahm-ghc].

Author’s Address

John Mattsson
Ericsson AB
Faeroegatan 6
Kista SE-164 80 Stockholm
Sweden

Email: john.mattsson@ericsson.com