DTLS Chunk for SCTP

Magnus Westerlund
John Preuß Mattsson
Claudio Porfiri
Michael Tüxen

draft-ietf-tsvwg-sctp-dtls-chunk-01

IETF 124 – TSVWG

2025-11-03

Outline

- Update
  - Changes since Madrid
  - Liaison Statement
- Issues
  - Negotiation of DTLS Chunk Clarifications
  - Alignment of the DTLS Record crypto part within the DTLS Chunk
  - ASCONF will be limited
  - DTLS Record Limits
  - Invocation Limits and Application Interactions

Status Update

- WG Draft-00 was identical to -05
- WG Draft-01 included some fixes:
  - Sequence number protection key also needed to provided
  - Renamed Abstract API invocations counter functions and added counter for failed invocations.
  - Socket API much more complete

Liaison Statement

- At last meeting it was agreed to send a Liaison statement to 3GPP RAN3 and SA3
  - To inform them about that TSVWG has made progress and chosen a direction
- Magnus forgot to write the draft LS until last week
- A draft LS do now exist for the WG chairs to edit and process

Proposed LS to 3GPP RAN3 and SA3

Title: IETF TSVWG's DTLS for SCTP Progress Report

From: IETF TSVWG

To: 3GPP-TSG-RAN-WG3, 3GPP-TSG-SA-WG3

To Contacts: Peter Schmitt <Peter.Schmitt@huawei.com>
Susanna Kooistra <3GPPLiaison@etsi.org>

Cc: Transport Area Working Group Discussion List <tsvwg@ietf.org>

Gorry Fairhurst <gorry@erg.abdn.ac.uk>

Purpose: For Information

Body:

IETF's Transport Working Group (TSVWG) likes to inform 3GPP's RAN3 and SA3 WG about its progress on defining a DTLS based security solution for SCTP that has occurred since the previous exchange of liaison statements [1,2,3].

Proposed LS: Page 2

TSVWG has finally agreed after extended deliberations on a technical direction and have adopted two new WG items for this new direction. The complete solution consists of two separate components.

The first component, which is the core of the complete solution, is called the DTLS Chunk. This component specifies how an SCTP endpoint initiate the negotiation using SCTP parameters, protects its packets using DTLS records using key-contexts that have been provisioned using an API to the SCTP stack. This protects the content of each SCTP packet in its own DTLS record. This results in all application messages as well as SCTP control chunks are encrypted and authenticated. It also ensures that there is no limit on application message sizes as they are fragmented by the regular SCTP mechanism supporting unlimited message size. This component is defined in the TSVWG adopted internet draft “Stream Control Transmission Protocol (SCTP) DTLS Chunk” [4]. The draft also defines certain requirements on key-management to prevent down-grade attacks.

Proposed LS: Page 3

The second component of the complete solution is a key-management specification that can provision the DTLS chunk (i.e., the first component) with keys. Such a key-management is defined as a functionality on top of the SCTP association, using an API defined in the DTLS chunk [4] to provision keys to secure the communication. TSVWG has adopted an Internet draft defining such a key-management titled: “Using Datagram Transport Layer Security (DTLS) for Key Management for the Stream Control Transmission Protocol (SCTP) DTLS Chunk” [5]. This key-management is dependent on a new TLS 1.3 [6] functionality — the extended key-update [7] being defined by the TLS WG. It will support long lived sessions through ephemeral rekeying ensuring forward secrecy and also supports SCTP Association restart.

Splitting the solution into two components (i.e., the DTLS chunk and the key-management) allows for alternative key-management to be specified if desired to achieve certain properties. There exists an individual internet draft detailing an alternative key-management [8], which can be seen as a proof that this divided concept works.

Proposed LS: Page 4

TSVWG targets to have the DTLS chunk, the first component, published as RFC before the end of 2026. The WG key-management specification [5], the second component, is being worked on as fast as possible. However, due to the dependency on the TLS extended key-update mechanism [7], which is also under specification and requires formal analysis, it is currently difficult to estimate an approximate completion date for the second component.

TSVWG would like to inform 3GPP RAN3, following their previous observations in [2], that this technical solution does not limit the maximum size of application protocol messages.

TSVWG plans to inform 3GPP RAN3 and SA3 when each of the components have been requested to be published. TSVWG welcome any feedback on its work items.

Proposed LS: Page 5

References:

[1] TSVWG LS to RAN3 and SA3: DTLS for SCTP next steps and request for input, https://datatracker.ietf.org/liaison/1851/

[2] R3-234497 - Reply LS on DTLS for SCTP next steps and request for input, https://datatracker.ietf.org/liaison/1858/

[3] S3-234160 - Reply LS on DTLS for SCTP, https://datatracker.ietf.org/liaison/1854/

[4] draft-ietf-tsvwg-sctp-dtls-chunk-01, https://datatracker.ietf.org/doc/draft-ietf-tsvwg-sctp-dtls-chunk/

[5] draft-ietf-tsvwg-dtls-chunk-key-management-00, https://datatracker.ietf.org/doc/draft-ietf-tsvwg-dtls-chunk-key-management/

[6] The Transport Layer Security (TLS) Protocol Version 1.3, RFC 8446, Aug 2018, https://datatracker.ietf.org/doc/rfc8446/

[7] Extended Key Update for Transport Layer Security (TLS) 1.3, draft-ietf-tls-extended-key-update-06, Oct 2025, https://datatracker.ietf.org/doc/draft-ietf-tls-extended-key-update/

[8] draft-westerlund-tsvwg-sctp-dtls-handshake-05, https://datatracker.ietf.org/doc/draft-westerlund-tsvwg-sctp-dtls-handshake/

Issues

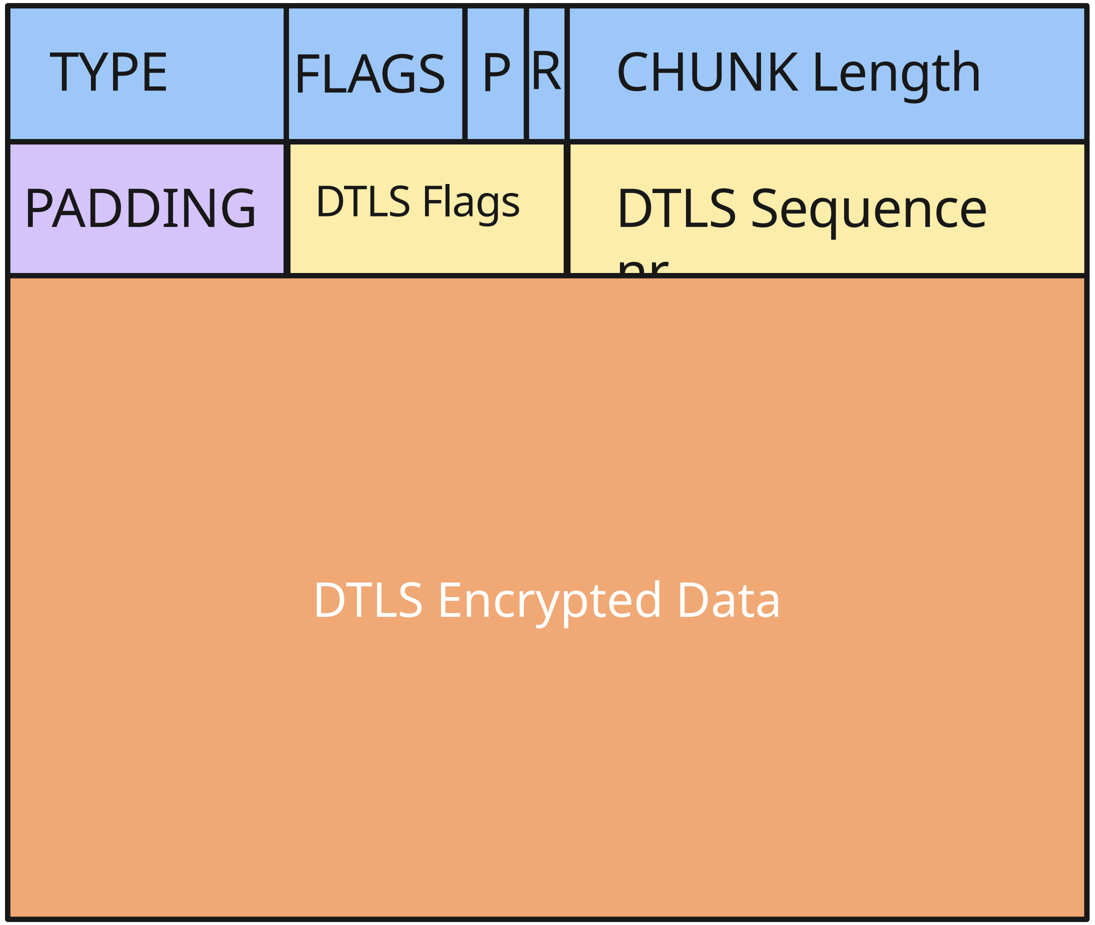
- Negotiation of DTLS Chunk Clarifications
- Alignment of the DTLS Record crypto part within the DTLS Chunk
- ASCONF will be limited
- DTLS Record Limits
- Invocation Limits and Application Interactions

Negotiation of DTLS Chunk Clarifications

- Issues: https://github.com/gloinul/draft-westerlund-tsvwg-sctp-dtls-chunk/issues/81
- Details of the initial negotiation procedures was unclear
  - Also to clarify what information goes into down-grade protection
- **DTLS 1.3 Chunk Protected Association**
  - Negotiates the usage of the defined DTLS Chunk and key-management method, i.e. the Protection Solution Identifiers
  - Initiator sends full list of support methods in priority order
  - Responder sends parameter with only selected method
- Down grade Protection
  - Include in key-derivation: Offer list + Selected answer

Alignment of the DTLS Record crypto part within the DTLS Chunk

- https://github.com/gloinul/draft-westerlund-tsvwg-sctp-dtls-chunk/issues/45
- SCTP chunks are 32-bit aligned. Implementation might depend on this (FreeBSD kernel implementation does).
- Desirable to do in place encryption and decryption
- Proposal provide a selective amount 0-3 bytes of padding prior to the DTLS record inside the DTLS chunk
  - Uses 2 bits of chunk flags (P)
- Enables the start of the encrypted part of the DTLS record to be aligned on a 32-bit boundary



ASCONF and DTLS Chunk

- https://github.com/gloinul/draft-westerlund-tsvwg-sctp-dtls-chunk/issues/28
- Issue what to do with ASCONF interaction (RFC 5061) as that requires SCTP-AUTH (RFC 4895) to authenticate the ASCONF packet
- ASCONF chunk include Address Parameter
  - Known address for Association
  - Used to lookup Association
- DTLS Chunk encryption prevent identifying association if packet comes from unknown peer address
  - Break before Make will fail

- Proposal:
  - Fulfill ASCONF's authentication requirement with DTLS Chunk
    - Means updating RFC 5061
  - Will limit ASCONF to make before break:
    - Send additional Interface address over existing path

DTLS Record Limits

- https://github.com/gloinul/draft-westerlund-tsvwg-sctp-dtls-chunk/issues/50
- RFC 8449 Defines (D)TLS Record Limit negotiation.
  - Peers agrees on sending a limited max size of DTLS records less than 16384 bytes
- This option intended mostly for limited implementation struggling to support 16384 records

- Proposal
  - Not provide any support for this option
  - Endpoints assumed to support 16384 bytes record.

Invocation Limits and Application Interactions

- The Abstract API specifies get methods:
  - **Get AEAD Encryption Invocations**
  - **Get AEAD Decryption Invocations**
  - **Get Failed AEAD Decryption Invocations**
- Is it beneficial to provide an Event Based Interaction?
  - Notify when total encryption Invocation has reached N
  - Notify when total decryption Invocations reach M
  - Notify when failed decryption Invocations reached O
- Any considerations around impact on the usage of these values?

Next Steps

- Incorporate the resolution of these issue in the draft
- Draft also needs editorial fixing to clean up unnecessary text
  - Expect next version to be a bit shorter
- Target new draft version before the end of the year