Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 3 September 2026

M. T端xen
M端nster Univ. of Appl. Sciences
H. Tschofenig
H-BRS
T. Reddy
Nokia

2 March 2026

Using Datagram Transport Layer Security (DTLS) for Key Management for the Stream Control Transmission Protocol (SCTP) DTLS Chunk
draft-ietf-tsvwg-dtls-chunk-key-management-01

Abstract

This document defines how Datagram Transport Layer Security (DTLS) 1.3 is used to establish keys for securing SCTP using the DTLS Chunk mechanism. It specifies how a DTLS handshake is used to establish the initial security context for an SCTP association and describes procedures for key updates and post-handshake authentication. The goal is to enable authenticated, and confidential communication over SCTP using the DTLS Chunk, leveraging standardized DTLS features for key management and rekeying.

Table of Contents

1. Introduction

The Stream Control Transmission Protocol (SCTP) is a transport protocol designed to support message-oriented communication with features such as multi-streaming and multi-homing. In many deployments, particularly those telecommunication networks and WebRTC data channels, it is essential to provide confidentiality, integrity, and peer authentication for SCTP traffic.

[RFC6083] defines a mechanism for securing SCTP by encapsulating it over DTLS 1.0/1.2, establishing a secure channel between SCTP endpoints. However, with the introduction of DTLS 1.3 [RFC9147], the protocol underwent significant changes, including removal of renegotiation, a new key schedule, and support for post-handshake operations. Without additional description, RFC 6083 cannot be used with DTLS 1.3.

Additionally, [I-D.ietf-tsvwg-sctp-dtls-chunk] defines a new method for secure and confidential transfer for SCTP packets but leaves key management to a separate specification.

This document complements [I-D.ietf-tsvwg-sctp-dtls-chunk] by specifying how DTLS 1.3 is used to derive and manage the cryptographic keys required for use with the DTLS Chunk, addressing key management aspects not covered in that specification. Specifically, it describes:

* How the DTLS 1.3 handshake establishes the initial security context between SCTP endpoints.

* How keying material is derived and associated with the SCTP association.

* How DTLS 1.3 features, such as key updates and post-handshake authentication, are leveraged to support long-lived secure sessions.

The goal of this specification is to modernize and extend the approach defined in RFC 6083, aligning it with the architectural and cryptographic changes introduced in DTLS 1.3, while enabling a cleaner and more efficient integration with native SCTP functionality.

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Architecture

This document describes how Datagram Transport Layer Security (DTLS) 1.3 is used to establish keys for securing SCTP using the DTLS Chunk as defined in [I-D.ietf-tsvwg-sctp-dtls-chunk]. This approach combines the performance and encryption flexibility of DTLS Chunks with the integrated key management capabilities of DTLS.

The key characteristics of the solution are as follows:

* Application data is protected using DTLS Chunks.

* The DTLS handshake is used to establish keying material, algorithms, and parameters for use with DTLS Chunks.

* DTLS and relevant extensions are used to negotiate cryptographic algorithms and parameters, perform extended key updates, enable larger record sizes, and support post-handshake authentication.

* All other DTLS record-layer content types are protected using standard DTLS record framing.

Application data (except the messages for post handshake authentication) is never transmitted in DTLS record-layer application_data records. Instead, application data is sent via SCTP DATA chunks which are protected by the DTLS Chunk Protection Operator. This operator encapsulates all SCTP chunks into a DTLS Chunk, applying appropriate protection.

The figure below illustrates the architecture, highlighting the role of the upper-layer protocol (ULP), which acts as the consumer of SCTP's transport services. The ULP may interface directly with the SCTP stack or operate through the DTLS 1.3 stack. This specification builds upon DTLS 1.3 by introducing additional extensions to support enhanced functionality, including post-handshake authentication as described in [RFC9261] and the extended key update mechanism defined in [I-D.ietf-tls-extended-key-update].

Following the initial SCTP association setup, a DTLS 1.3 handshake is performed to mutually authenticate the endpoints and to derive keying material for the DTLS Chunk Protection Operator. The TLS exporter, as defined in Section 7.5 of [RFC8446], is used to derive this keying material. It leverages the same cryptographic algorithms that were negotiated during the DTLS handshake for use with the DTLS Record Layer, thereby eliminating the need for separate algorithm negotiation for the DTLS Chunk. However, this approach requires that only algorithm suites compatible with both DTLS 1.3 and DTLS Chunks be configured and supported.

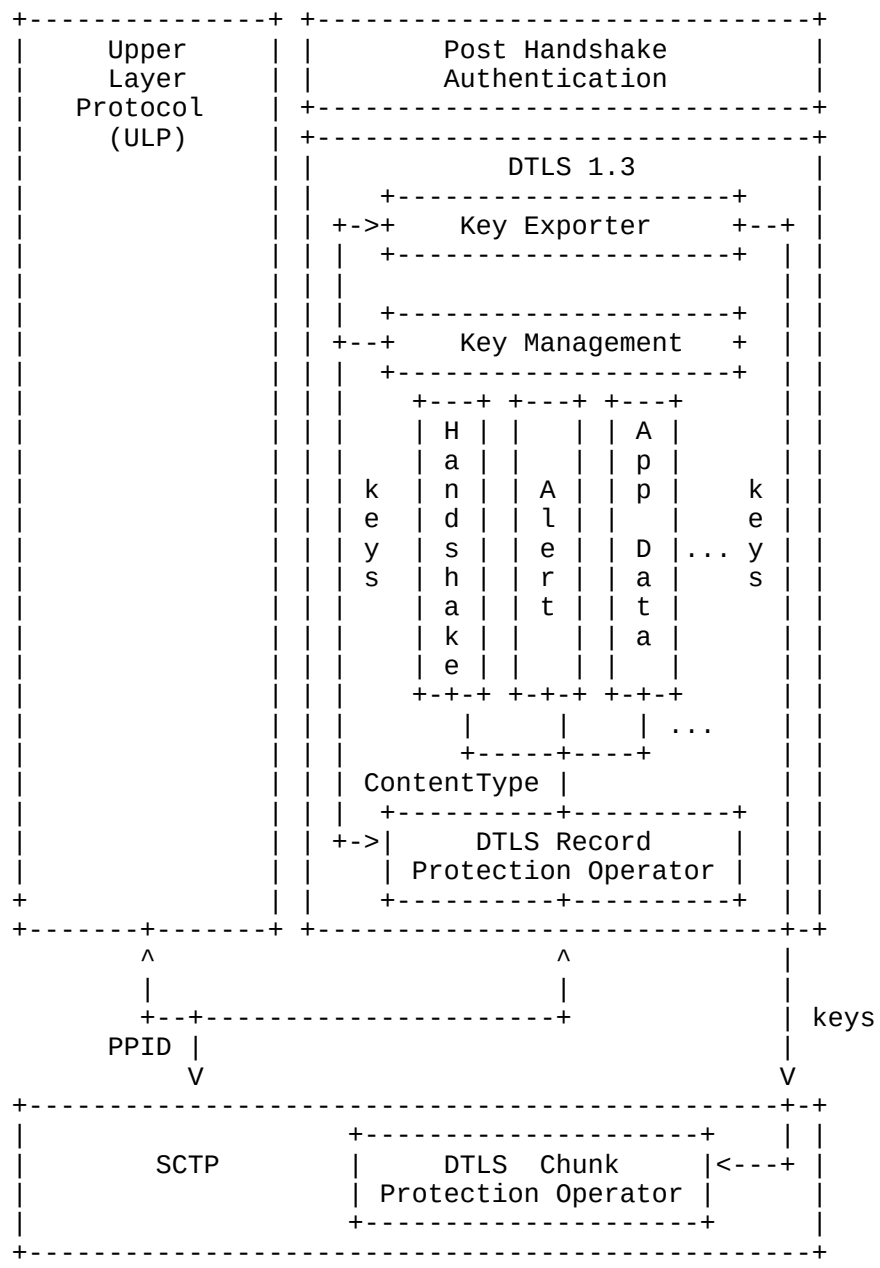

Figure 1: Architecture

4. Setting the Keys Initially

The following figure shows when the key exporter is used to get key material from the DTLS connection. Then keys are derived from that and the diagram also shows when these keys are configured for the DTLS chunk and also when the SCTP stack is instructed to discard received SCTP packets, when they are unprotected.

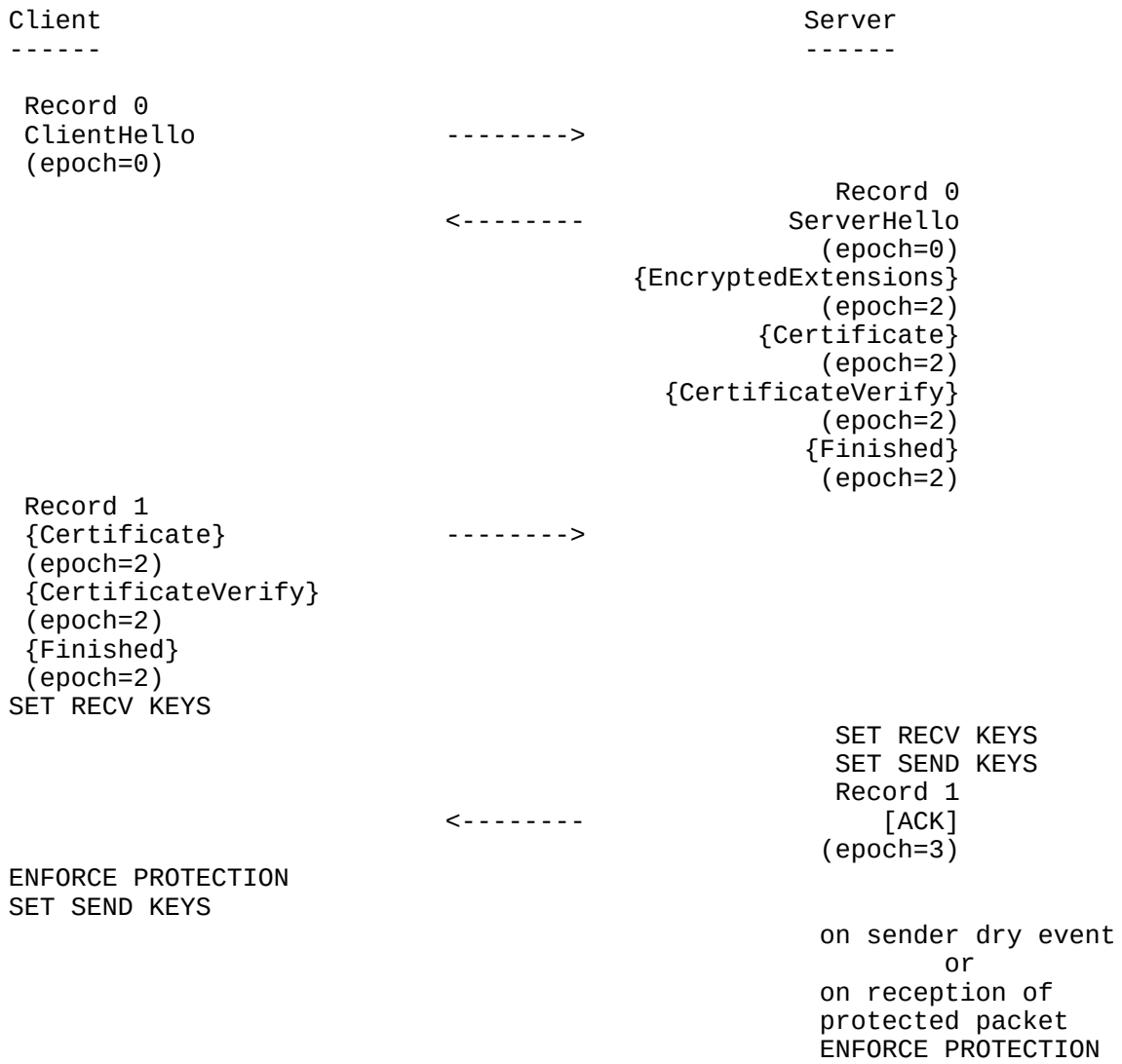

Figure 2: Usage of Key Exporter

The key derivation takes into account which protection solution identifiers have been sent and received. This way the communication is protected against downgrade attacks against the SCTP handshake.

TBD: Take the message flow into account which was presented by Magnus.

TBD: Provide formulas for deriving the keys and improve the message sequence diagram.

5. Updating the Keys

For updating the keys used for the DTLS chunk, the key updated described in [RFC9147] MUST NOT be used, since it does not update the exporter keys. Therefore the extended key update as described in [I-D.ietf-tls-extended-key-update] MUST be used. When a receive key for epoc n + 1 is installed, the receive key for epoc n - 1 SHOULD be removed. Deriving the keys from the exported keys is done similar to the initial derivation.

TBD: Provide formulas for the key derivation and a message sequence diagram for the extended key update.

6. Post-Handshake Authentication

SCTP peers may require periodic re-authentication after the DTLS handshake has completed. This is needed, for example, to demonstrate continued possession of a private key. In such cases, peers MUST use the post-handshake authentication mechanism defined in [RFC9261].

The application-layer protocol running over DTLS MUST be used to transmit the Authenticator Request. Either the SCTP client or the SCTP server may initiate this request using the established DTLS connection.

Similarly, the application-layer protocol running over DTLS MUST be used to transmit the corresponding Authenticator message.

TBD: A description of the application-layer message format(s) that carry the Authenticator Request and Authenticator will be provided in a future revision. This includes a message sequence chart illustrating the communication.

7. IANA Considerations

Note: IANA is requested to replace RFC-to-be with a reference to this document.

IANA is requested to add the following entry to the DTLS Key Management Method Identifiers registry of the Stream Control Transmission Protocol (SCTP) Parameters group:

Identifier	Key Management Method Name	Reference	Contact
1	Single DTLS Connection	RFC-to-be	Document Authors

Table 1

8. Security Considerations

TBD.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC6083] Tuexen, M., Seggelmann, R., and E. Rescorla, "Datagram Transport Layer Security (DTLS) for Stream Control Transmission Protocol (SCTP)", RFC 6083, DOI 10.17487/RFC6083, January 2011, <https://www.rfc-editor.org/info/rfc6083>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC9147] Rescorla, E., Tschofenig, H., and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", RFC 9147, DOI 10.17487/RFC9147, April 2022, <https://www.rfc-editor.org/info/rfc9147>.

[RFC9260] Stewart, R., T端xen, M., and K. Nielsen, "Stream Control Transmission Protocol", RFC 9260, DOI 10.17487/RFC9260, June 2022, <https://www.rfc-editor.org/info/rfc9260>.

[RFC9261] Sullivan, N., "Exported Authenticators in TLS", RFC 9261, DOI 10.17487/RFC9261, July 2022, <https://www.rfc-editor.org/info/rfc9261>.

[I-D.ietf-tsvwg-sctp-dtls-chunk] Westerlund, M., Mattsson, J. P., Porfiri, C., and M. T端xen, "Stream Control Transmission Protocol (SCTP) DTLS Chunk", Work in Progress, Internet-Draft, draft-ietf-tsvwg-sctp-dtls-chunk-01, 20 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-tsvwg-sctp-dtls-chunk-01>.

[I-D.ietf-tls-extended-key-update] Tschofenig, H., T端xen, M., Reddy, T., Fries, S., and Y. Rosomakho, "Extended Key Update for Transport Layer Security (TLS) 1.3", Work in Progress, Internet-Draft, draft-ietf-tls-extended-key-update-10, 2 March 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-tls-extended-key-update-10>.

Acknowledgments

The authors wish to thank Claudio Porfiri and Magnus Westerlund for their invaluable comments.

Authors' Addresses

Michael T端xen
M端nster University of Applied Sciences
Stegerwaldstrasse 39
48565 Steinfurt
Germany
Email: tuexen@fh-muenster.de

Hannes Tschofenig
University of Applied Sciences Bonn-Rhein-Sieg
Email: Hannes.Tschofenig@gmx.net

Tirumaleswar Reddy
Nokia
Email: kondtir@gmail.com