Workgroup: TSVWG
Internet-Draft:
draft-westerlund-tsvwg-dtls-over-sctp-bis-01
Obsoletes: 6083 (if approved)
Published: 22 February 2021
Intended Status: Standards Track
Expires: 26 August 2021
Authors: M. Westerlund
Ericsson
J. Preuß Mattsson
Ericsson
C. Porfiri
Ericsson
M. Tüxen
Münster Univ. of Appl. Sciences

Datagram Transport Layer Security (DTLS) over Stream Control Transmission Protocol (SCTP)

Abstract

This document describes a proposed update for the usage of the Datagram Transport Layer Security (DTLS) protocol to protect user messages sent over the Stream Control Transmission Protocol (SCTP).

DTLS over SCTP provides mutual authentication, confidentiality, integrity protection, and replay protection for applications that use SCTP as their transport protocol and allows client/server applications to communicate in a way that is designed to give communications privacy and to prevent eavesdropping and detect tampering or message forgery.

Applications using DTLS over SCTP can use almost all transport features provided by SCTP and its extensions. This document intends to obsolete RFC 6083 and removes the 16 kB limitation on user message size by defining a secure user message fragmentation so that multiple DTLS records can be used to protect a single user message. It further updates the DTLS versions to use, as well as the HMAC algorithms for SCTP-AUTH, and simplifies the implementation by some stricter requirements on the establishment procedures.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the TSVWG Working Group mailing list (tsvwg@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/tsvwg/.

Source for this draft and an issue tracker can be found at https://github.com/gloinul/draft-westerlund-tsvwg-dtls-over-sctp-bis.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

1.1. Overview

This document describes the usage of the Datagram Transport Layer Security (DTLS) protocol, as defined in [I-D.ietf-tls-dtls13], over the Stream Control Transmission Protocol (SCTP), as defined in [RFC4960] with Authenticated Chunks for SCTP (SCTP-AUTH) [RFC4895].

This specification provides mutual authentication of endpoints, confidentiality, integrity protection, and replay protection of user messages for applications that use SCTP as their transport protocol. Thus it allows client/server applications to communicate in a way that is designed to give communications privacy and to prevent eavesdropping and detect tampering or message forgery. DTLS/SCTP

uses DTLS for mutual authentication, key exchange with perfect forward secrecy for SCTP-AUTH, and confidentiality of user messages. DTLS/SCTP use SCTP and SCTP-AUTH for integrity protection and replay protection of user messages.

Applications using DTLS over SCTP can use almost all transport features provided by SCTP and its extensions. DTLS/SCTP supports:

- *preservation of message boundaries.
- *a large number of unidirectional and bidirectional streams.
- *ordered and unordered delivery of SCTP user messages.
- *the partial reliability extension as defined in [RFC3758].
- *the dynamic address reconfiguration extension as defined in [RFC5061].
- *large user messages.

The method described in this document requires that the SCTP implementation supports the optional feature of fragmentation of SCTP user messages as defined in [RFC4960]. To efficiently implement and support larger user messages it is also recommended that I-DATA chunks as defined in [RFC8260] as well as an SCTP API that supports partial user message delivery as discussed in [RFC6458].

1.1.1. Comparison with TLS for SCTP

TLS, from which DTLS was derived, is designed to run on top of a byte-stream-oriented transport protocol providing a reliable, in-sequence delivery. TLS over SCTP as described in [RFC3436] has some serious limitations:

- *It does not support the unordered delivery of SCTP user messages.
- *It does not support partial reliability as defined in [RFC3758].
- *It only supports the usage of the same number of streams in both directions.
- *It uses a TLS connection for every bidirectional stream, which requires a substantial amount of resources and message exchanges if a large number of streams is used.

1.1.2. Changes from RFC 6083

The DTLS over SCTP solution defined in RFC 6083 had the following limitation:

- The maximum user message size is 2^14 bytes, which is a single DTLS record limit.

This update that replaces RFC6083 defines the following changes:

- Removes the limitations on user messages sizes by defining a secure fragmentation mechanism.
- Defines a DTLS extension for the endpoints to declare the user message size supported to be received.
- Mandates that more modern DTLS version are required (DTLS 1.2 or 1.3)
- Mandates use of modern HMAC algorithm (SHA-256) in the SCTP authentication extension [RFC4895].
- Recommends support of [RFC8260] to enable interleaving of large SCTP user messages to avoid scheduling issues.
- Recommends support of partial message delivery API, see [RFC6458] if larger usage messages are intended to be used.
- Applies stricter requirements on always using DTLS for all user messages in the SCTP association.
- Requires that SCTP-AUTH is applied to all SCTP Chunks that can be authenticated.

1.2. Terminology

This document uses the following terms:

Association: An SCTP association.

Stream: A unidirectional stream of an SCTP association. It is uniquely identified by a stream identifier.

1.3. Abbreviations

DTLS: Datagram Transport Layer Security

MTU: Maximum Transmission Unit

PPID: Payload Protocol Identifier

SCTP: Stream Control Transmission Protocol

TCP: Transmission Control Protocol

TLS: Transport Layer Security

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. DTLS Considerations

3.1. Version of DTLS

This document is based on DTLS 1.3 [I-D.ietf-tls-dtls13], but works also for DTLS 1.2 [RFC6347]. Earlier versions of DTLS MUST NOT be used. It is expected that DTLS/SCTP as described in this document will work with future versions of DTLS.

3.2. Cipher Suites

For DTLS 1.2, the cipher suites forbidden by [RFC7540] MUST NOT be used. Cipher suites without encryption MUST NOT be used.

3.3. Message Sizes

DTLS/SCTP, automatically fragments and reassembles user messages. This specification defines how to fragment the user messages into DTLS records, where each DTLS 1.3 record allows a maximum of 2^14 protected bytes. Each DTLS record adds some overhead, thus using records of maximum possible size are recommended to minimize the overhead.

The sequence of DTLS records is then fragmented into DATA or I-DATA Chunks to fit the path MTU by SCTP. The largest possible user messages using the mechanism defined in this specification is 2^64-1 bytes.

The security operations and reassembly process requires that the protected user message, i.e. with DTLS record overhead, is buffered in the receiver. This buffer space will thus put a limit on the largest size of plain text user message that can be transferred securely.

A receiver that doesn't support partial delivery of user messages from SCTP [RFC6458] will advertise its largest supported protected

message using SCTP's mechanism for Advertised Receiver Window Credit (a_rwnd) as specified in Section 3.3.2 of [RFC4960]. Note that the a_rwnd value is across all user messages being delivered.

For a receiver supporting partial delivery of user messages a_rwnd will not limit the maximum size of the DTLS protected user message because the receiver can move parts of the DTLS protected user message from the SCTP receiver buffer into a buffer for DTLS processing. When each complete DTLS record have been received from SCTP, it can be processed and the plain text fragment can, in its turn, be partially delivered to the user application.

Thus, the limit of the largest user message is dependent on buffering allocated for DTLS processing as well as the DTLS/SCTP API to the application. To ensure that the sender have some understanding of the maximum receiver size a TLS extension "dtls_over_sctp_maximum_message_size" Section 5.2 is used to signal the endpoints receiver capability when it comes to user message size.

All implementors of this specification MUST support user messages of at least 16383 bytes. Where 16383 bytes is the supported message size in RFC 6083. By requiring this message size in this document, we ensure compatibility with existing usage of RFC 6083, not requiring the upper layer protocol to implement additional features or requirements.

Due to SCTP's capability to transmit concurrent user messages the total memory consumption in the receiver is not bounded. In cases where one or more user messages are affected by packet loss, the DATA chunks may require more data in the receiver's buffer.

The necessary buffering space for a single user message of dtls_over_sctp_maximum_message_size (MMS) is dependent on the implementation.

When no partial data delivery is supported, the message size is limited by the a_rwnd as this is the largest protected user message that can be received and then processed by DTLS and where the plain text user message is expected to be no more than the signalled MMS.

With partial processing it is possible to have a receiver implementation that is bound to use no more buffer space than MMS (for the plaintext) plus one maximum size DTLS record. The later assumes that one can realign the start of the buffer after each DTLS record has been consumed. A more realistic implementation is two maximum DTLS record sizes.

If an implementation supports partial delivery in both the SCTP API and the ULP API, and also partial processing in the DTLS/SCTP

implementation, then the buffering space in the DTLS/SCTP layer ought to be no more than two DTLS records. In which case the MMS to set is dependent on the ULP and the endpoints capabilities.

3.4. Replay Protection

SCTP-AUTH [RFC4895] does not have explicit replay protection. However, the combination of SCTP-AUTH's protection of DATA or I-DATA chunks and SCTP user message handling will prevent third party attempts to inject or replay SCTP packets resulting in impact on the received protected user message. In fact this document's solution is dependent on SCTP-AUTH and SCTP to prevent reordering of the DTLS records within each protected user message.

DTLS optionally supports record replay detection. Such replay detection could result in the DTLS layer dropping valid messages received outside of the DTLS replay window. As DTLS/SCTP provides replay protection even without DTLS replay protection, the replay detection of DTLS MUST NOT be used.

3.5. Path MTU Discovery

DTLS Path MTU Discovery MUST NOT be used. Since SCTP provides own Path MTU discovery and fragmentation/reassembly for user messages, and according to Section 3.3, DTLS can send maximum sized DTLS Records.

3.6. Retransmission of Messages

SCTP provides a reliable and in-sequence transport service for DTLS messages that require it. See Section 4.4. Therefore, DTLS procedures for retransmissions MUST NOT be used.

4. SCTP Considerations

4.1. Mapping of DTLS Records

The SCTP implementation MUST support fragmentation of user messages using DATA [RFC4960], and optionally I-DATA [RFC8260] chunks.

DTLS/SCTP works as a shim layer between the user message API and SCTP. The fragmentation works similar as the DTLS fragmentation of handshake messages. On the sender side a user message fragmented into fragments m0, m1, m2, each no larger than 2^14 - 1 = 16383 bytes.

m0 | m1 | m2 | ... = user_message

The resulting fragments are protected with DTLS and the records are concatenated

user_message' = DTLS(m0) | DTLS(m1) | DTLS(m2) ...

The new user_message', i.e the protected user message, is the input to SCTP.

On the receiving side DTLS is used to decrypt the records. If a DTLS decryption fails, the DTLS connection and the SCTP association are terminated. Due to SCTP-AUTH preventing delivery of corrupt fragments of the protected user message this should only occur in case of implementation errors or internal hardware failures.

The DTLS Connection ID SHOULD NOT be negotiated (Section 9 of [I-D.ietf-tls-dtls13]). If DTLS 1.3 is used, the length field MUST NOT be omitted and a 16 bit sequence number SHOULD be used.

4.2. DTLS Connection Handling

The DTLS connection MUST be established at the beginning of the SCTP association and be terminated when the SCTP association is terminated, (i.e. there's only one DTLS connection within one association). A DTLS connection MUST NOT span multiple SCTP associations.

As it is required to establish the DTLS connection at the beginning of the SCTP association, either of the peers should never send any SCTP user messages that are not protected by DTLS. So the case that an endpoint receives data that is not either DTLS messages on Strea 0 or protecetd user messages in the form of a sequence of DTLS Records on any stream is a protocol violation. The receiver MAY terminate the SCTP association due to this protocol violation.

4.3. Payload Protocol Identifier Usage

SCTP Payload Protocol Identifiers are assigned by IANA. Application protocols using DTLS over SCTP SHOULD register and use a separate Payload Protocol Identifier (PPID) and SHOULD NOT reuse the PPID that they registered for running directly over SCTP.

Using the same PPID does not harm as long as the application can determine whether or not DTLS is used. However, for protocol analyzers, for example, it is much easier if a separate PPID is used.

This means, in particular, that there is no specific PPID for DTLS.

4.4. Stream Usage

All DTLS Handshake, Alert, or ChangeCipherSpec (DTLS 1.2 only) messages MUST be transported on stream 0 with unlimited reliability and with the ordered delivery feature.

DTLS messages of the record protocol, which carries the protected user messages, SHOULD use multiple streams other than stream 0; they MAY use stream 0 as long as the ordered message semantics is acceptable. On stream 0 protected user messages as well as any DTLS messages that isn't record protocol will be mixed, thus the additional head of line blocking can occur.

4.5. Chunk Handling

DATA chunks of SCTP MUST be sent in an authenticated way as described in [RFC4895]. All other chunks that can be authenticated, i.e. all chunk types that can be listed in the Chunk List Parameter [RFC4895], MUST also be sent in an authenticated way. This makes sure that an attacker cannot modify the stream in which a message is sent or affect the ordered/unordered delivery of the message.

If PR-SCTP as defined in [RFC3758] is used, FORWARD-TSN chunks MUST also be sent in an authenticated way as described in [RFC4895]. This makes sure that it is not possible for an attacker to drop messages and use forged FORWARD-TSN, SACK, and/or SHUTDOWN chunks to hide this dropping.

I-DATA chunk type as defined in [RFC8260] is RECOMMENDED to be supported to avoid some of the down sides that large user messages have on blocking transmission of later arriving high priority user messages. However, the support is not mandated and negotiated independently from DTLS/SCTP. If I-DATA chunks are used then they MUST be sent in an authenticated way as described in [RFC4895].

4.6. SCTP-AUTH Hash Function

When using DTLS/SCTP, the SHA-256 Message Digest Algorithm MUST be supported in the SCTP-AUTH [RFC4895] implementation. SHA-1 MUST NOT be used when using DTLS/SCTP. [RFC4895] requires support and inclusion of of SHA-1 in the HMAC-ALGO parameter, thus, to meet both requirements the HMAC-ALGO parameter will include both SHA-256 and SHA-1 with SHA-256 listed prior to SHA-1 to indicate the preference.

4.7. Renegotiation

Renegotiation enables rekeying and reauthentication inside an DTLS 1.2 connection. It is up to the upper layer to use/allow it or not. Application writers should be aware that allowing renegotiations may result in changes of security parameters. Renegotiation has been removed from DTLS 1.3 and partly replaced with Post-Handshake messages such as KeyUpdate. See Section 7 for security considerations regarding rekeying.

4.8. DTLS Epochs

In general, DTLS implementations SHOULD discard records from earlier epochs, as described in Section 4.2.1 of [I-D.ietf-tls-dtls13]. To avoid discarding messages, the processing guidelines in Section 4.2.1 of DTLS 1.3 [I-D.ietf-tls-dtls13] or Section 4.1 or DTLS 1.2 [RFC6347] should be followed.

4.9. Handling of Endpoint-Pair Shared Secrets

SCTP-AUTH [RFC4895] is keyed using Endpoint-Pair Shared Secrets. In SCTP associations where DTLS is used, DTLS is used to establish these secrets. The endpoints MUST NOT use another mechanism for establishing shared secrets for SCTP-AUTH.

The endpoint-pair shared secret for Shared Key Identifier 0 is empty and MUST be used when establishing a DTLS connection. In DTLS 1.2, whenever the main secret changes, a 64-byte shared secret is derived from every main secret and provided as a new endpoint-pair shared secret by using the TLS-Exporter. In DTLS 1.3, the exporter_secret never change. For DTLS 1.3, the exporter is described in [RFC8446]. For DTLS 1.2, the exporter is described in [RFC5705]. The exporter MUST use the label given in Section Section 6 and no context. The new Shared Key Identifier MUST be the old Shared Key Identifier incremented by 1. If the old one is 65535, the new one MUST be 1.

Before sending the DTLS Finished message, the active SCTP-AUTH key MUST be switched to the new one.

Once the corresponding Finished message from the peer has been received, the old SCTP-AUTH key SHOULD be removed.

4.10. Shutdown

To prevent DTLS from discarding DTLS user messages while it is shutting down, a CloseNotify message MUST only be sent after all outstanding SCTP user messages have been acknowledged by the SCTP peer and MUST NOT be revoked by the SCTP peer.

Prior to processing a received CloseNotify, all other received SCTP user messages that are buffered in the SCTP layer MUST be read and processed by DTLS.

5. DTLS over SCTP Service

The adoption of DTLS over SCTP according to the current description is meant to add to SCTP the option for transferring encrypted data. When DTLS over SCTP is used, all data being transferred MUST be protected by chunk authentication and DTLS encrypted. Chunks that need to be received in an authenticated way will be specified in the

CHUNK list parameter according to [RFC4895]. Error handling for authenticated chunks is according to [RFC4895].

5.1. Adaptation Layer Indication in INIT/INIT-ACK

At the initialization of the association, a sender of the INIT or INIT ACK chunk that intends to use DTLS/SCTP as specified in this specification MUST include an Adaptation Layer Indication Parameter with the IANA assigned value TBD to inform its peer that it is able to support DTLS over SCTP per this specification.

5.2. DTLS/SCTP "dtls_over_sctp_maximum_message_size" Extension

The endpoint's DTLS/SCTP maximum message size is declared in the "dtls_over_sctp_maximum_message_size" TLS extension. The ExtensionData of the extension is MessageSizeLimit:

```
uint64 MessageSizeLimit;
```

The value of MessageSizeLimit is the maximum plaintext user message size in octets that the endpoint is willing to receive. When the "dtls_over_sctp_maximum_message_size" extension is negotiated, an endpoint MUST NOT send a user message larger than the MessageSizeLimit value it receives from its peer.

This value is the length of the user message before DTLS fragmentation and protection. The value does not account for the expansion due to record protection, record padding, or the DTLS header.

The "dtls_over_sctp_maximum_message_size" MUST be used to negotiate maximum message size for DTLS/SCTP. A DTLS/SCTP endpoint MUST treat the omission of "dtls_over_sctp_maximum_message_size" as a fatal error unless supporting RFC 6083 fallback Section 5.6, and it SHOULD generate an "illegal_parameter" alert. Endpoints MUST NOT send a "dtls_over_sctp_maximum_message_size" extension with a value smaller than 16383. An endpoint MUST treat receipt of a smaller value as a fatal error and generate an "illegal_parameter" alert.

The "dtls_over_sctp_maximum_message_size" MUST NOT be send in TLS or in DTLS versions earlier than 1.2. In DTLS 1.3, the server sends the "dtls_over_sctp_maximum_message_size" extension in the EncryptedExtensions message.

During resumption, the maximum message size is renegotiated.

5.3. DTLS over SCTP Initialization

Initialization of DTLS/SCTP requires all the following options to be part of the INIT/INIT-ACK handshake:

RANDOM: defined in [RFC4895]

CHUNKS: list of permitted chunks, defined in [RFC4895]

HMAC-ALGO: defined in [RFC4895]

ADAPTATION-LAYER-INDICATION: defined in [RFC5061]

When all the above options are present, the Association will start with support of DTLS/SCTP. The set of options indicated are the DTLS/SCTP Mandatory Options. No data transfer is permitted before DTLS handshake is complete. Chunk bundling is permitted according to [RFC4960]. The DTLS handshake will enable authentication of both the peers and also have the declare their support message size.

The extension described in this document is given by the following message exchange.

```
--- INIT[RANDOM; CHUNKS; HMAC-ALGO; ADAPTATION-LAYER-IND] --->
<- INIT-ACK[RANDOM; CHUNKS; HMAC-ALGO; ADAPTATION-LAYER-IND] -
----------------------- COOKIE-ECHO ----------------------->
<------------------------ COOKIE-ACK ------------------------
---------------- AUTH; DATA[DTLS Handshake] ----------------->
                              ...
                              ...
<--------------- AUTH; DATA[DTLS Handshake] ------------------
```

5.4. Client Use Case

When a SCTP Client initiates an Association with DTLS/SCTP Mandatory Options, it can receive an INIT-ACK also containing DTLS/SCTP Mandatory Options, in that case the Association will proceed as specified in the previous Section 5.3 section. If the peer replies with an INIT-ACK not containing all DTLS/SCTP Mandatory Options, the Client can decide to keep on working with RFC 6083 fallback, plain data only, or to ABORT the association.

5.5. Server Use Case

If a SCTP Server supports DTLS/SCTP, when receiving an INIT chunk with all DTLS/SCTP Mandatory Options it must reply with INIT-ACK also containing the all DTLS/SCTP Mandatory Options, then it must follow the sequence for DTLS initialization Section 5.3 and the related traffic case. If a SCTP Server supports DTLS, when receiving an INIT chunk with not all DTLS/SCTP Mandatory Options, it can decide to continue by creating an Association with RFC 6083 fallback, plain data only or to ABORT it.

5.6. RFC 6083 Fallback

This section discusses how an endpoint supporting this specification can fallback to follow the DTLS/SCTP behavior in RFC 6083. It is recommended to define a setting that represents the policy to allow fallback or not. However, the possibility to use fallback is based on the ULP can operate using user messages that are no longer than 16383 bytes and where the security issues can be mitigated or considerd acceptable. Fallback is NOT RECOMMEND to be enabled as it enables downgrade to weaker algorithms and versions of DTLS.

A SCTP client that receives an INIT-ACK that is not compliant according this specification may in certain cases potentially perform an fallback to RFC 6083 behavior. The first case is when the SCTP client receives an INIT-ACK doesn't contain the SCTP-Adaptation-Indication parameter with the DTLS/SCTP adaptation layer codepoint but do include the SCTP-AUTH parameters on a server that are expected to provide services using DTLS. The second case is when the INIT-ACK do contain the SCTP-Adaptation-Indication parameter with the correct code point, however the HMAC-ALGO or the Chunks parameters values are such that do not fullfil the requirement of this specification but do meet the requirements of RFC 6083. In either of these cases the client could attempt DTLS per RFC 6083 as fallback. However, the fallback attempt should only be performed if policy says that is acceptable.

If fallback is allowed it is possible that the client will send plain text user messages prior to DTLS handshake as it is allowed per RFC 6083. So that needs to be part of the consideration for a policy allowing fallback. When performing the the DTLS handshake, the server is required accepting that lack of the TLS extension "dtls_over_sctp_maximum_message_size" and can't treat it as fatal error. In case the "dtls_over_sctp_maximum_message_size" TLS extension is present in the handshake the server SHALL continue the handshake including the extension with its value also, and from that point follow this specification. In case the TLS option is missing RFC 6083 applies.

6. IANA Considerations

6.1. TLS Exporter Label

RFC 6083 defined a TLS Exporter Label registry as described in [RFC5705]. IANA is requested to update the reference for the label "EXPORTER_DTLS_OVER_SCTP" to this specification.

6.2. DTLS "dtls_over_sctp_buffer_size_limit" Extension

This document registers the "dtls_over_sctp_maximum_message_size" extension in the TLS "ExtensionType Values" registry established in

[RFC5246]. The "dtls_over_sctp_maximum_message_size" extension has been assigned a code point of TBD. This entry [[will be|is]] marked as recommended ([RFC8447] and marked as "Encrypted" in (D)TLS 1.3 [I-D.ietf-tls-dtls13]. The IANA registry [RFC8447] [[will list|lists]] this extension as "Recommended" (i.e., "Y") and indicates that it may appear in the ClientHello (CH) or EncryptedExtensions (EE) messages in (D)TLS 1.3 [I-D.ietf-tls-dtls13].

6.3. SCTP Parameter

IANA is requested to assign a Adaptation Code Point for DTLS/SCTP.

7. Security Considerations

The security considerations given in [I-D.ietf-tls-dtls13], [RFC4895], and [RFC4960] also apply to this document.

7.1. Cryptographic Considerations

Over the years, there have been several serious attacks on earlier versions of Transport Layer Security (TLS), including attacks on its most commonly used ciphers and modes of operation. [RFC7457] summarizes the attacks that were known at the time of publishing and BCP 195 [RFC7525] provides recommendations for improving the security of deployed services that use TLS.

When DTLS/SCTP is used with DTLS 1.2 [RFC6347], DTLS 1.2 MUST be configured to disable options known to provide insufficient security. HTTP/2 [RFC7540] gives good minimum requirements based on the attacks that where publicly known in 2015. DTLS 1.3 [I-D.ietf-tls-dtls13] only define strong algorithms without major weaknesses at the time of publication. Many of the TLS registries have a "Recommended" column. Parameters not marked as "Y" are NOT RECOMMENDED to support.

DTLS 1.3 requires rekeying before algorithm specific AEAD limits have been reached. The AEAD limits equations are equally valid for DTLS 1.2 and SHOULD be followed for DTLS/SCTP, but are not mandated by the DTLS 1.2 specification. HMAC-SHA-256 as used in SCTP-AUTH has a very large tag length and very good integrity properties. The SCTP-AUTH key can be used until the DTLS handshake is re-run at which point a new SCTP-AUTH key is derived using the TLS-Exporter.

DTLS/SCTP is in many deployments replacing IPsec. For IPsec, NIST (US), BSI (Germany), and ANSSI (France) recommends very frequent re-run of Diffie-Hellman to provide Perfect Forward Secrecy. ANSSI writes "It is recommended to force the periodic renewal of the keys, e.g. every hour and every 100 GB of data, in order to limit the impact of a key compromise." [ANSSI-DAT-NT-003].

For many DTLS/SCTP deployments the DTLS connections are expected to have very long lifetimes of months or even years. For connections with such long lifetimes there is a need to frequently re-authenticate both client and server.

When using DTLS 1.2 [RFC6347], AEAD limits, frequant re-authentication and frequent re-run of Diffie-Hellman can be achieved with frequent renegotiation, see TLS 1.2 [RFC5246]. When renegotiation is used both clients and servers MUST use the renegotiation_info extension [RFC5746] and MUST follow the renegotiation guidelines in BCP 195 [RFC7525].

In DTLS 1.3 renegotiation has been removed from DTLS 1.3 and partly replaced with Post-Handshake KeyUpdate. When using DTLS 1.3 [I-D.ietf-tls-dtls13], AEAD limits and frequent rekeying can be achieved by sending frequent Post-Handshake KeyUpdate messages. Symmetric rekeying gives less protection against key leakage than re-running Diffie-Hellman. After leakage of application_traffic_secret_N, a passive attacker can passively eavesdrop on all future application data sent on the connection including application data encrypted with application_traffic_secret_N+1, application_traffic_secret_N+2, etc. The is no way to do Post-Handshake server authentication or Ephemeral Diffie-Hellman inside a DTLS 1.3 connection. Note that KeyUpdate does not update the exporter_secret.

7.2. Downgrade Attacks

A peer supporting DTLS/SCTP according to this specification, DTLS/SCTP according to [RFC6083] and/or SCTP without DTLS may be vulnerable to downgrade attacks where on on-path attacker interferes with the protocol setup to lower or disable security. If possible, it is RECOMMENDED that the peers have a policy only allowing DTLS/SCTP according to this specification.

7.3. DTLS/SCTP Message Sizes

The DTLS/SCTP maximum message size extension enables secure negation of a message sizes that fit in the DTLS/SCTP buffer, which improves security and availability. Very small plain text user fragment sizes might generate additional work for senders and receivers, limiting throughput and increasing exposure to denial of service.

The maximum message size extension does not protect against peer nodes intending to negatively affect the peer node through flooding attacks. The attacking node can both send larger messages than the expressed capability as well as initiating a large number of concurrent user message transmissions that never are concluded. For

the target of the attack it is more straight forward to determine that a peer is ignoring the node's stated limitation.

7.4. Authentication and Policy Decisions

DTLS/SCTP MUST be mutually authenticated. It is RECOMMENDED that DTLS/SCTP is used with certificate based authentication. All security decisions MUST be based on the peer's authenticated identity, not on its transport layer identity.

It is possible to authenticate DTLS endpoints based on IP addresses in certificates. SCTP associations can use multiple IP addresses per SCTP endpoint. Therefore, it is possible that DTLS records will be sent from a different source IP address or to a different destination IP address than that originally authenticated. This is not a problem provided that no security decisions are made based on the source or destination IP addresses.

7.5. Privacy Considerations

[RFC6973] suggests that the privacy considerations of IETF protocols be documented.

For each SCTP user message, the user also provides a stream identifier, a flag to indicate whether the message is sent ordered or unordered, and a payload protocol identifier. Although DTLS/SCTP provides privacy for the actual user message, the other three information fields are not confidentiality protected. They are sent as clear text, because they are part of the SCTP DATA chunk header.

It is RECOMMENDED that DTLS/SCTP is used with certificate based authentication in DTLS 1.3 [I-D.ietf-tls-dtls13] to provide identity protection. DTLS/SCTP MUST be used with a key exchange method providing Perfect Forward Secrecy. Perfect Forward Secrecy significantly limits the amount of data that can be compromised due to key compromise.

7.6. Pervasive Monitoring

As required by [RFC7258], work on IETF protocols needs to consider the effects of pervasive monitoring and mitigate them when possible.

Pervasive Monitoring is widespread surveillance of users. By encrypting more information including user identities, DTLS 1.3 offers much better protection against pervasive monitoring.

Massive pervasive monitoring attacks relying on key exchange without forward secrecy has been reported. By mandating perfect forward secrecy, DTLS/SCTP effectively mitigate many forms of passive

pervasive monitoring and limits the amount of compromised data due to key compromise.

In addition to the privacy attacks discussed above, surveillance on a large scale may enable tracking of a user over a wider geographical area and across different access networks. Using information from DTLS/SCTP together with information gathered from other protocols increases the risk of identifying individual users.

8. Acknowledgments

The authors of RFC 6083 which this document is based on are Michael Tuexen, Eric Rescorla, and Robin Seggelmann.

The RFC 6083 authors thanked Anna Brunstrom, Lars Eggert, Gorry Fairhurst, Ian Goldberg, Alfred Hoenes, Carsten Hohendorf, Stefan Lindskog, Daniel Mentz, and Sean Turner for their invaluable comments.

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>.

[RFC3758] Stewart, R., Ramalho, M., Xie, Q., Tuexen, M., and P. Conrad, "Stream Control Transmission Protocol (SCTP) Partial Reliability Extension", RFC 3758, DOI 10.17487/RFC3758, May 2004, <https://www.rfc-editor.org/info/rfc3758>.

[RFC4895] Tuexen, M., Stewart, R., Lei, P., and E. Rescorla, "Authenticated Chunks for the Stream Control Transmission Protocol (SCTP)", RFC 4895, DOI 10.17487/RFC4895, August 2007, <https://www.rfc-editor.org/info/rfc4895>.

[RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, DOI 10.17487/RFC4960, September 2007, <https://www.rfc-editor.org/info/rfc4960>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/

RFC5246, August 2008, <https://www.rfc-editor.org/info/rfc5246>.

[RFC5705] Rescorla, E., "Keying Material Exporters for Transport Layer Security (TLS)", RFC 5705, DOI 10.17487/RFC5705, March 2010, <https://www.rfc-editor.org/info/rfc5705>.

[RFC5746] Rescorla, E., Ray, M., Dispensa, S., and N. Oskov, "Transport Layer Security (TLS) Renegotiation Indication Extension", RFC 5746, DOI 10.17487/RFC5746, February 2010, <https://www.rfc-editor.org/info/rfc5746>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/info/rfc6347>.

[RFC7540] Belshe, M., Peon, R., and M. Thomson, Ed., "Hypertext Transfer Protocol Version 2 (HTTP/2)", RFC 7540, DOI 10.17487/RFC7540, May 2015, <https://www.rfc-editor.org/info/rfc7540>.

[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>.

[RFC8260] Stewart, R., Tuexen, M., Loreto, S., and R. Seggelmann, "Stream Schedulers and User Message Interleaving for the Stream Control Transmission Protocol", RFC 8260, DOI 10.17487/RFC8260, November 2017, <https://www.rfc-editor.org/info/rfc8260>.

[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>.

[RFC8447] Salowey, J. and S. Turner, "IANA Registry Updates for TLS and DTLS", RFC 8447, DOI 10.17487/RFC8447, August 2018, <https://www.rfc-editor.org/info/rfc8447>.

[I-D.ietf-tls-dtls13] Rescorla, E., Tschofenig, H., and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", Work in Progress, Internet-Draft, draft-ietf-tls-dtls13-40, 20 January 2021, <http://www.ietf.org/internet-drafts/draft-ietf-tls-dtls13-40.txt>.

9.2. Informative References

[RFC3436] Jungmaier, A., Rescorla, E., and M. Tuexen, "Transport Layer Security over Stream Control Transmission

Protocol", RFC 3436, DOI 10.17487/RFC3436, December 2002, <https://www.rfc-editor.org/info/rfc3436>.

[RFC5061] Stewart, R., Xie, Q., Tuexen, M., Maruyama, S., and M. Kozuka, "Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration", RFC 5061, DOI 10.17487/RFC5061, September 2007, <https://www.rfc-editor.org/info/rfc5061>.

[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>.

[RFC6458] Stewart, R., Tuexen, M., Poon, K., Lei, P., and V. Yasevich, "Sockets API Extensions for the Stream Control Transmission Protocol (SCTP)", RFC 6458, DOI 10.17487/RFC6458, December 2011, <https://www.rfc-editor.org/info/rfc6458>.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <https://www.rfc-editor.org/info/rfc6973>.

[RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <https://www.rfc-editor.org/info/rfc7258>.

[RFC7457] Sheffer, Y., Holz, R., and P. Saint-Andre, "Summarizing Known Attacks on Transport Layer Security (TLS) and Datagram TLS (DTLS)", RFC 7457, DOI 10.17487/RFC7457, February 2015, <https://www.rfc-editor.org/info/rfc7457>.

[RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <https://www.rfc-editor.org/info/rfc7525>.

[ANSSI-DAT-NT-003] Agence nationale de la sécurité des systèmes d'information, ., "Recommendations for securing networks with IPsec", ANSSI Technical Report DAT-NT-003 , August 2015, <https://www.ssi.gouv.fr/uploads/2015/09/NT_IPsec_EN.pdf>.

Appendix A. Motivation for Changes

This document proposes a number of changes to RFC 6083 that have various different motivations:

Supporting Large User Messages: RFC 6083 allowed only user messages that could fit within a single DTLS record. 3GPP has run into this limitation where they have at least four SCTP using protocols (F1, E1, Xn, NG-C) that can potentially generate messages over the size of 16384 bytes.

New Versions: Almost 10 years has passed since RFC 6083 was written, and significant evolution has happened in the area of DTLS and security algorithms. Thus DTLS 1.3 is the newest version of DTLS and also the SHA-1 HMAC algorithm of RFC 4895 is getting towards the end of usefulness. Thus, this document mandates usage of relevant versions and algorithms.

Clarifications: Some implementation experiences has been gained that motivates additional clarifications on the specification.

*Avoid unsecured messages prior to DTLS handshake have completed.

*Make clear that all messages are encrypted after DTLS handshake.

Authors' Addresses

Magnus Westerlund
Ericsson

Email: magnus.westerlund@ericsson.com

John Preuß Mattsson
Ericsson

Email: john.mattsson@ericsson.com

Claudio Porfiri
Ericsson

Email: claudio.porfiri@ericsson.com

Michael Tüxen
Münster University of Applied Sciences
Stegerwaldstrasse 39
48565 Steinfurt
Germany

Email: tuexen@fh-muenster.de