TLS Working Group
Internet Draft
Intended status: Experimental

P. Urien
Telecom Paris
Ethertrust

28 May 2026

Expires: November 2026

TLS 1.3 Identity Module Trusted Exporter
draft-urien-tls-im-trusted-exporter-00.txt

Abstract

The Transport Layer Security (TLS) 1.3 protocol supports external Pre-Shared Keys (PSKs), which are provisioned out of band. A PSK binder, included in the ClientHello message, is computed as an HMAC over a transcript hash using a key called the Finished External Key (FEK). For the "PSK with (EC)DHE" key exchange mode, where Diffie-Hellman is performed over either finite fields or elliptic curves, the Handshake Secret (HS) is computed from the (EC)DHE shared secret using HKDF-Extract with a key called the Derived Secret Key (DSK), which is derived from the PSK. A TLS identity module SHOULD be used to protect procedures involving keys bound to the PSK, such as the FEK or the DSK. TLS defines keying material exporters, which rely on secrets produced during the handshake protocol. This draft introduces an Exporter Trusted Key (ETK), which is securely stored and used within a TLS identity module. The ETK transforms exporter secrets into trusted values that cannot be recovered by TLS software. A trusted exporter is similar to the legacy TLS exporter, but it uses an additional trusted secret.

Requirements Language

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.

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 November 2026.

Copyright Notice

Table of Contents

1 Introduction

TLS 1.3 [RFC8446] defines keying material exporters. An exporter relies on a secret produced during the handshake protocol. This secret is either:

- the early-exporter-master-secret, computed from the PSK as follows:

  Derive-Secret(Early-Secret, "e exp master", ClientHello)

- or the exporter-master-secret, computed from the master secret as follows:

  Derive-Secret(master-secret, "exp master", ClientHello...Server Finished)

The main objective of this draft is to define exporters that can only be computed within a TLS identity module [IM].

This draft defines the Exported Trusted Key (ETK), derived from the PSK and securely stored and used within the TLS identity module. This key modifies the exporter secrets required by the TLS Trusted Exporter according to the following relation:

  Trusted-Secret = HKDF-Extract(Secret, ETK) = HMAC(Secret, ETK)

The TLS Trusted Exporter is similar to the legacy TLS exporter, but it uses a Trusted-Secret.

2 TLS PSK with (EC)DHE

2.1 PSK Binder procedure

According to [RFC8446], external PSKs MAY be provisioned outside TLS.

The Early Secret (ESK) is computed as follows:

  ESK = HKDF-Extract(salt = 0s, PSK) = HMAC(salt = 0s, PSK)

The Binder Key (BSK) for external provisioning is computed as follows:

  BSK = Derive-Secret(ESK, "ext binder", "")

The Finished External Key (FEK) is computed as follows:

  FEK = KDF-Expand-Label(BSK, "finished", "", Hash.length)

For Derive-Secret procedures, "" is equivalent to Hash(empty), whose size is Hash.length.

The PSK binder is computed as follows:

```
PSK-Binder = HMAC(FEK, transcript_hash)
```

The PSK binder is included in clear text in the ClientHello message. It can therefore be used in brute-force attacks to recover the PSK value.

2.2 Handshake Secret

The Derived Secret Key (DSK) is computed as follows:

```
DSK = Derive-Secret(ESK, "derived", "")
```

The Handshake Secret (HS) is computed as follows:

```
HS = HKDF-Extract(salt = DSK, (EC)DHE)
```

2.3 Identity Module

A TLS identity module [IM] securely computes the PSK binder and the Handshake Secret.

2.4 TLS For Secure Element

TLS for secure elements [TLSSE] is a TLS 1.3 server using the "PSK with (EC)DHE" exchange mode and running inside a secure element, i.e., a tamper-resistant device.

The Trusted Exporter MAY be used to export wrapping keys that cannot be recovered by client software, without a TLS identity module.

3 Exporter Trusted Key

The Exporter Trusted Key (ETK) is always stored and used within the TLS identity module [IM].

It is computed according to the relation

ETK = Derive-Secret(ESK, "trusted exporter", "")

4 TLS-Exporter Secrets

4.1 Early Exporter Master Secret

EEMS= early-exporter-master-secret = Derive-Secret(ESK, "e exp master", ClientHello)

4.2 Exporter Master Secret

DS = Derive-Secret(HS, "derived", "")

MasterSecret = MS= HKDF-Extract(DS,0s) = HMAC(DS,0s)

EMS= exporter_master_secret=
Derive-Secret(MS, "exp master", ClientHello...server Finished)

4 TLS-Trusted-Exporter Secrets

4.1 Trusted Early Exporter Master Secret

The trusted-early-exporter-master-secret is defined as :

trusted-early-exporter-master-secret = TEEMS =
HKDF-Extract(EEMS, ETK) = HMAC(EEMS, ETK)

4.2 Trusted-Exporter-Master-Secret

The trusted-exporter-master-secret is defined as:

trusted-exporter-master-secret = TEMS =
HKDF-Extract(EMS, ETK) = HMAC(EMS, ETK)

5 TLS Trusted Exporter

[RFC5705] defines keying material exporters for TLS in terms of the TLS pseudorandom function (PRF). [RFC8446] replaces the PRF with HKDF, thus requiring a new construction. The exporter interface remains the same. The exporter value is computed as:

```
TLS-Exporter(label, context_value, key_length) =
  HKDF-Expand-Label(Derive-Secret(Secret, label, ""),
                    "exporter", Hash(context_value),
                     key_length)
```

According to this draft the TLS-Trusted-Exporter procedure SHOULD be executed by the TLS Identity module, and is defined as

```
TLS-Trusted-Exporter(label, context_value, key_length) =
    HKDF-Expand-Label(Derive-Secret(Trusted-Secret, label, ""),
    "exporter", Hash(context_value),
    key_length)
```

5 IANA Considerations

This draft does not require any action from IANA.

6 Security Considerations

This entire document is about security.

7 References

7.1 Normative References

[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.

[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.

7.2 Informative References

[IM] Urien, P., "Identity Module for TLS Version 1.3", draft-urien-tls-im-10.txt, January 2024.

[TLSSE] Urien, P., "Secure Element for TLS Version 1.3", draft-urien-tls-se-08.txt June 2024

8 Authors' Addresses

Pascal Urien
EtherTrust - Telecom Paris
19 place Marguerite Perey
91120 Palaiseau
France
Email: Pascal.Urien@telecom-paris.fr