TLS Working Group
Internet Draft
Intended status: Experimental

P. Urien
Telecom Paris
Ethertrust

3 April 2026

Expires: 4 October 2026

TLS for Secure Element Recursive Authentication
draft-urien-tls-se-xauth-03

Abstract

This document defines a recursive authentication architecture based on the TLS 1.3 pre-shared key (PSK) mode. In this context, TLS servers, typically hosted within secure elements (TLS-SE), realize procedures that compute TLS 1.3 PSK-binder and Handshake Secret. These procedures allow a client to authenticate to downstream TLS servers without directly possessing the corresponding PSKs. Authentication capabilities can therefore be delegated across multiple TLS servers while maintaining protection of the underlying secrets.

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

.

Copyright Notice

Table of Contents

1 Introduction

TLS1.3 [RFC8446] supports a basic key exchange mode based on pre-shared-key (PSK) with Diffie-Hellman over either finite fields or elliptic curves (EC)DHE, in short TLS1.3-PSK for PSK with (EC)DHE. According to [RFC8446] external PSKs MAY be provisioned outside of TLS [RFC9257}.
TLS1.3-PSK server MAY be implemented in secure elements [TLS-SE].
The TLS1.3 client MAY use a TLS Identity Module [TLS-IM], that protects procedures dealing with PSK.

1.1 PSK Binder procedure

The Early Secret (ESK) is computed according to relation:
ESK =HKDF-Extract(salt=0s,PSK) = HMAC(salt=0s,PSK)

The Binder Key (BSK) for outside provisioning is computed according to the relation:
BSK = Derive-Secret(ESK, "ext binder", "")

The Finished External Key (FEK) is computed according to the relation:
FEK = KDF-Expand-Label(BSK, "finished", "", Hash.length)

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

The PSK Binder is computed as:

PSK-Binder=HMAC(FEK, transcript hash)

PSK-Binder is included in the Client Hello message.

We name binder(transcript hash) or binder(T) the procedure that computes the PSK-Binder. Because PSK relies on psk-identity inserted in the ClientHello message, we note by binder(ID,T) this relationship.

1.2 Handshake Secret Procedure

The Derived Secret (DSK) is computed according to the relation:
DSK= Derive-Secret(ESK, "derived", "").

The Handshake Secret (HS) is computed as:

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

We name derive(DHE) the procedure that computes the Handshake Secret. Because PSK relies on psk-identity inserted in the ClientHello message, we note by derive(ID,DHE) this relationship.

The recovery of HS needs the knowledge of both PSK and (EC)DHE. Therefore, even if PSK has low entropy, it is extremely difficult to retrieve the HS value.

1.3 Get psk-identity

This optional procedure, noted GetID, MAY be used to retrieve the psk-identity associated with binder(ID,T) and derive(ID,DHE)procedures (with ID=psk-identity)

2 Recursive authentication with TLS-PSK

2.1 Overview

In traditional TLS-PSK deployments the client must possess the PSK locally in order to compute the PSK-Binder and the Handshake Secret. Distributing PSKs to clients introduces operational challenges and security risks, especially in large distributed systems.

This document introduces a recursive authentication model in which TLS1.3 PSK servers perform PSK-related cryptographic computations on behalf of the client. The client invokes these procedures during the TLS1.3 handshake without learning the underlying PSKs.

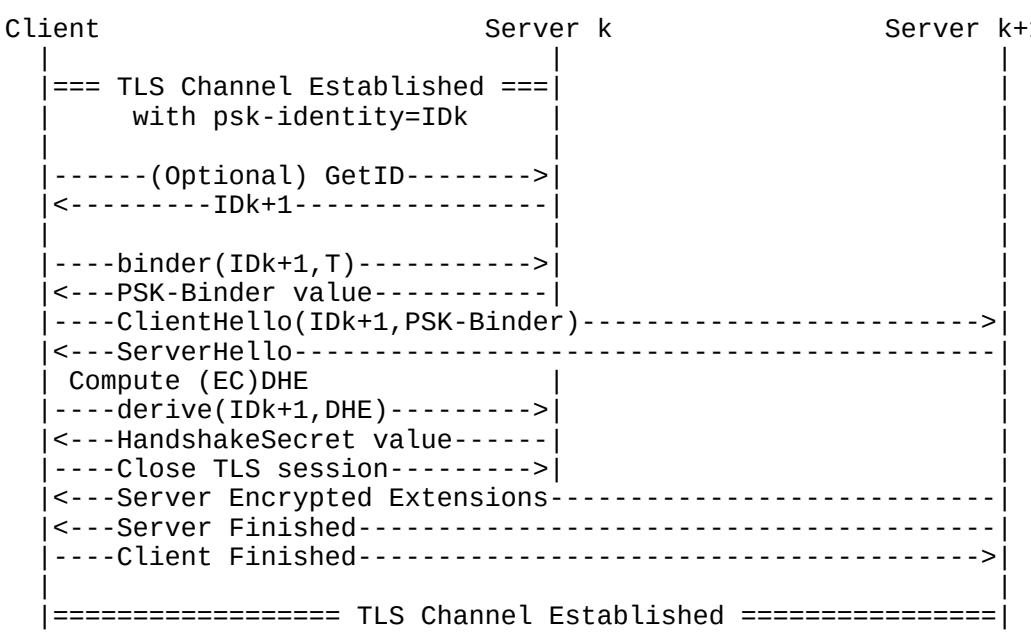
After authenticating to one TLS secure element server [TLS-SE], the client uses delegated cryptographic services from that server to authenticate to additional TLS-SE servers. Authentication relationships therefore form a recursive chain across multiple servers.

2.2 Recursive Authentication

The schema below illustrates recursive authentication with two TLS1.3 servers, server k and server k+1. Two procedures binder; which computes the PSK-Binder value, and derive, which computes the Handshake Secret are realized by server k and used for connection to server k+1.

GetID MAY be used to retrieve the next psk-identity (IDk+1) associated with binder and derive procedures.

It should be noticed that server k generates Handshake Secret for server k+1, and therefore COULD decrypt further data exchanges between client and server k+1.



2.3 Root Server

The root server (RS) is the first server to which client is connected; it provides binder and derive procedures to be used for connection to the second server (server 2). According to [TLS-SE] root server MAY be implemented in secure elements [ISO7816}.

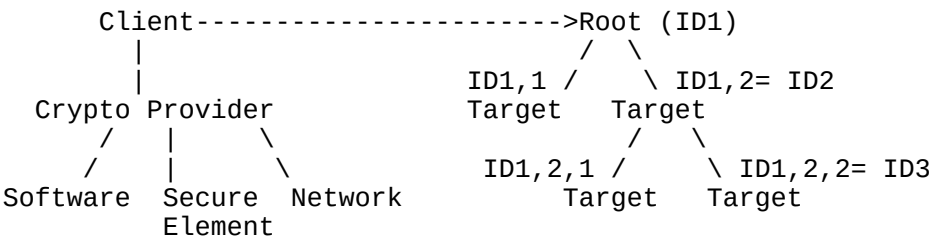
2.4 Crypto Provider

We define a Crypto Provider (CP) as the entity that holds the pre-shared key (PSK) used with the root server (Server 1), and is therefore capable of performing the binder(T) computation and the derive(DHE) procedures. A Crypto Provider can be implemented in multiple computing environments, such as:
-Classical software implementations that use the PSK value to perform binder and derive procedures. In this case, the PSK is exposed in software memory, making it susceptible to extraction and cloning.
-Tamper-resistant computing devices, such as secure elements compliant with the ISO 7816 standards. The IETF draft [TLS-IM] defines ISO 7816 interfaces for executing binder and derive procedures within secure elements. This environment provides strong key protection, ensures that secrets are non-extractable, and supports portable identity.
- Remote procedure calls hosted on network devices. These devices are similar to connected smart card readers capable of transporting ISO 7816 requests and responses over protocols such as TLS. This enables remote access to secure elements implementing binder and

derive functions as defined in [TLS-IM], while still benefiting from hardware-backed key protection. According to the IETF draft [RACS], clients and servers are identified using X.509 certificates, targeting virtual machine environments with PKI support. According to the IETF draft [RACSL], clients and servers use TLS 1.3 with pre-shared key authentication, typically targeting embedded environments.

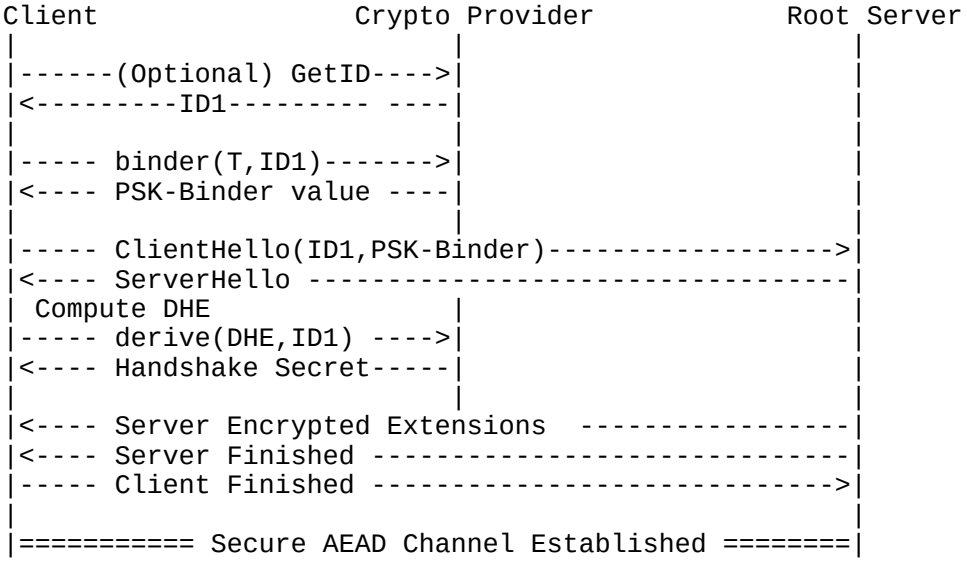
2.5 Target server

Recursive Authentication creates a tree of servers, the root node is the root server, while other are referred as target servers. The client uses a crypto provider for connection to root. According to [TLS-SE] target server MAY be implemented in secure elements [ISO7816].

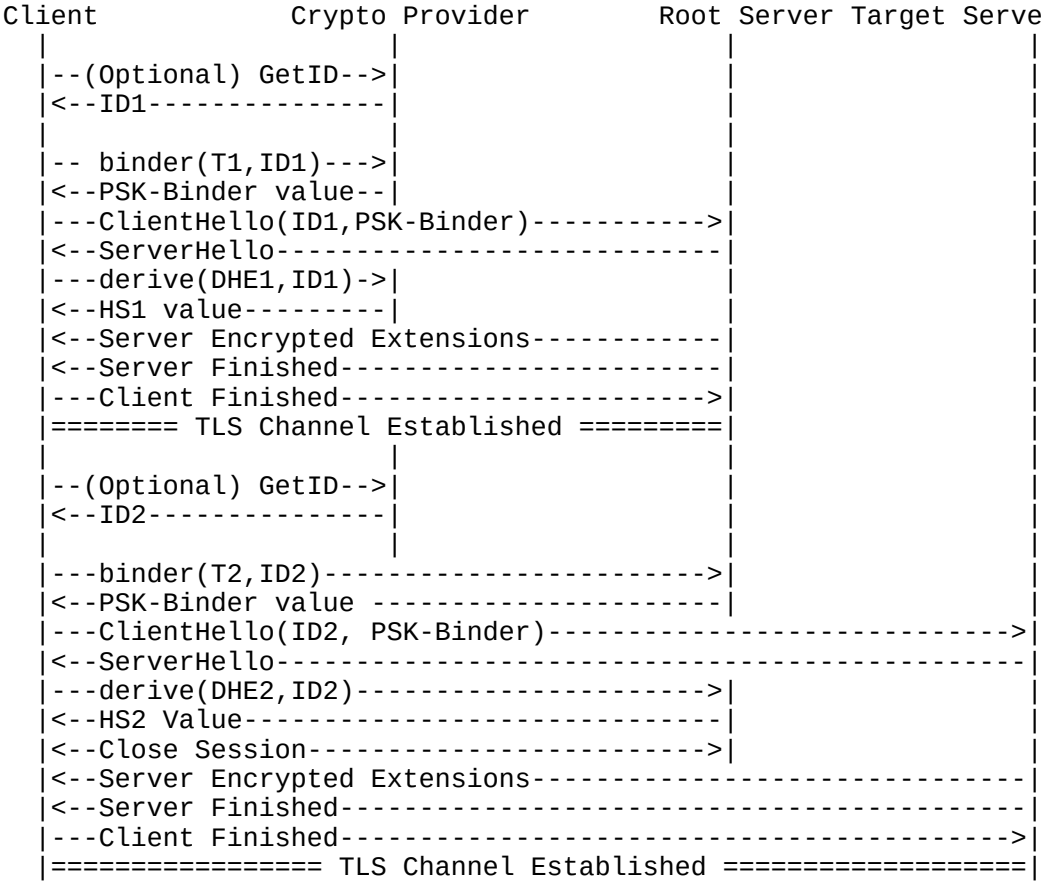


3 1-HOP session

A 1-HOP session establishes a session between a client and a root server.



4 2-HOPs session



A 2-HOPs session establishes a session with a target server. The root server typically acts as a gateway that checks the client rights; it MAY allocate a new psk-identity ID2, according to a particular role based access control.

An example of 2-HOPs client, is available at [TLSSE-Client].

5 n-HOPs session

Client establishes a first session 1 with root server, using a crypto provider that computes binder(ID1,T1) and derive(ID1,DHE1). It MAY use the GetID procedure to get ID1.

Client establishes a session 2 with target 1 server, using binder(ID2,T2) and derive(ID2,DHE2,) provided par root server. Client closes session with root server. It MAY use the GetID procedure to get ID2.

Client establishes a session k+1 with target k server, using binder(IDk+1,Tk+1) and derive(IDk+1,DHEk+1) provided par target k-1 server. Client closes session with target k-1. It MAY use the GetID procedure to get IDk+1.

Client establishes a session n with target n-1 server, using binder(IDn,Tn) and derive(IDn,DHEn) provided par target n-2 server. Client closes session with target n-2. It MAY use the GetID procedure to get IDn.

At depth k in the tree, IDk can be expressed with k indexes:
IDk=ID(1,I2,..,Ik),
where Ij (j between 2 and k) indicates the child rank of node j-1, between 1 and number of child.

6 IANA Considerations

This draft does not require any action from IANA.

7 Security Considerations

Recursive Authentication MAY be realized with classical software TLS-PSK server, but in these platforms PSK is exposed in software memory, and cloning is possible.

Recursive Authentication within TLS-SE servers has the followings security benefits :

- PSK Protection :PSKs remain protected inside secure elements and MUST NOT be exposed to clients.

- Delegated Authentication: Authentication capabilities are delegated through controlled cryptographic procedures rather than through direct key distribution.

- Compartmentalization: Each TLS-SE server manages its own PSKs and delegation policies. Compromise of one server does not automatically compromise unrelated servers.

- Revocation: Removing a server from the authentication chain prevents further delegation to downstream servers.

8 References

8.1 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>

[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.

[RFC9257] HOUSLEY, R., HOYLAND, J., SETHI, M., et al. Rfc 9257: Guidance for external pre-shared key (psk) usage in tls. 2022.

[ISO7816] ISO 7816, "Cards Identification - Integrated Circuit Cards with Contacts", The International Organization for Standardization (ISO).

8.2 Informative References

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

[RACS] "Remote APDU Call Secure (RACS)", draft-urien-core-racs-19.txt, February 2024

[RACSL] "Remote APDU Call Secure Lite (RACSL)", draft-urien-core-racsl-00.txt, March 2026

[TLS-SE] "Secure Element for TLS Version 1.3", draft-urien-tls-se-08, December 2024

[TLSSE-Client] "2-Hops Recursive Authentication Client", https://github.com/purien/pnhsm

9 Authors’ Addresses

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