EMU working group
Internet-Draft
Intended status: Standards Track
Expires: 16 August 2026

A. DeKok
InkBridge Networks

12 February 2026

Tunnel Extensible Authentication Protocol (TEAP) Version 2
draft-dekok-emu-teapv2-03

Abstract

This document defines the Tunnel Extensible Authentication Protocol (TEAP) version 2. It addresses a number of security and interoperability issues which were found during the publication of TEAPv1 ([I-D.ietf-emu-rfc7170bis]).

About This Document

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

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-dekok-emu-teapv2/.

Discussion of this document takes place on the EMU Working Group mailing list (mailto:emu@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/emu/. Subscribe at https://www.ietf.org/mailman/listinfo/emu/.

Source for this draft and an issue tracker can be found at https://github.com/inkbridgenetworks/teapv2.git.

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 16 August 2026.

Table of Contents

1. Introduction

Tunnel Extensible Authentication Protocol (TEAP) version 1 was first defined in [RFC7170]. However, implementations of that specification were found to have limited interoperability, due to the complexity of the design, and to under-specification of the cryptographic key deriviations that it defined.

TEAPv1 was updated and clarified in [I-D.ietf-emu-rfc7170bis]. That document described a large amount of potential functionality in the protocol, but also noted in [I-D.ietf-emu-rfc7170bis], Section 5.1 that only a small subset of that functionality was interoperable. In addition, the interoperable parts of the protocol had security issues which could potentially allow on-path attackers control control over the data being transported inside of the TLS tunnel.

We do not review the full security issues with TEAPv1 here. Instead, we define TEAPv2, with new and simpler cryptographic key deriviations. These derivations address all of the known issues with TEAPv1.

NOTE: For simplicity, this current draft defines TEAPv2 as a delta from TEAPv1. Once the overall framework has been agreed upon, this document will be updated to define TEAPv2 without referring to TEAPv1. i.e. Implementing TEAPv2 will not require anyone to read the previous TEAPv1 document.

1.1. Common Elements With TEAPv1

Most parts of TEAPv1 are unchanged. The message and TLV formats are the same, as are the derivations for the session_key_seed ([I-D.ietf-emu-rfc7170bis], Section 6.1). The derivations for the Master Session Key (MSK) and the Extended Master Session Key (EMSK) ([I-D.ietf-emu-rfc7170bis], Section 6.4) have had only minor changes.

The Crypto-Binding TLV [I-D.ietf-emu-rfc7170bis], Section 4.2.13 has the same format as for TEAPv1, although its value is now calculated via a different derivation. The Crypto-Binding TLV is now used only when the inner exchange is an EAP authentication method.

The largest difference between TEAPv1 and TEAPv2 is in the cryptographic calculations of the intermediate keys. The calculations are simpler than the ones defined for TEAPv1, without compromising on security.

A related problem with TEAPv1 was that the Crypto-Binding TLV calculation had significant differences between independent implementations. These differences were due to complex derivations

of multiple intermediate keys, some of which were defined ambiguously. This complexity and lack of clarity lead to interoperability problems. TEAPv2 simplifies the key deriviations, reduces the number of intermediate keys, and gives smaller (and hopefully clearer) definitions for the derivations.

NOTE: TEAPv2 will also include mandatory support for all protocol operations.

1.2. Outline of this Document

This document largely follows the same outline as [I-D.ietf-emu-rfc7170bis]. Where changes from that document are made, the same section titles are used in order to ensure easy comparison of the documents. New sections are added, with new titles. For parts of TEAP which are not mentioned herein, this document makes no changes from [I-D.ietf-emu-rfc7170bis].

1.3. Specification Requirements

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.

1.4. Terminology

Round

A round is one set of inner message exchanges. Each round finishes with an Interim-Result TLV and a Cryptographic-Binding TLV. Other TLVs may also be included in a round.

A round can include multiple inner message exchanges, e.g. EAP-TLS.

This term was used in [I-D.ietf-emu-rfc7170bis], but was not defined in that document.

2. Negotiation

TEAPv2 uses the same version negotiation method as is defined in [I-D.ietf-emu-rfc7170bis], Section 3.1, with the Version field set to two (2) for TEAPv2.

TEAPv2 MUST use TLS 1.3 or later. TEAPv2 MUST NOT use TLS-PSK.

3. Cryptographic Calculations

The crytographic calculations for TEAPv2 have been substantially simplified from those defined in [I-D.ietf-emu-rfc7170bis], Section 6.

3.1. TEAP Authentication Phase 1: Key Derivations

The session key seed is the same as defined in [I-D.ietf-emu-rfc7170bis], Section 6.2.1 for TEAPv2. That definition is reproduced here verbatim:

```
session_key_seed = TLS-Exporter(
                    "EXPORTER: teap session key seed",, 40)
```

3.2. Inner Methods

There are multiple possible message exchanges for Inner Methods, including passwords, PKCS#7 data, and EAP methods. Not all of these inner methods will provide an MSK or an EMSK.

TEAPv2 uses the Crypto-Binding TLV to protect both parties from the attacks outlined in [RFC6677]. As those attacks are only possible when the inner method is EAP, TEAPv2 defines key derivations only when then inner method is EAP. TEAPv2 does not define key deriviations or use the Crypto-Binding TLV for other inner method exchanges.

The key deriviations described below MUST be done when the inner method is EAP (i.e. when the parties exchange EAP-Payload TLVs). The key deriviations described below MUST NOT be done when the inner method is anything other than EAP (i.e. when the inner method does not include EAP-Payload TLVs).

Since the Crypto-Binding TLV depends on the key derivations, it has the same requirements. A Crypto-Binding TLV MUST be used when the inner method is EAP. A Crypto-Binding TLV MUST NOT be used when the inner method is anything other than EAP.

Inner EAP methods MUST produce an MSK and/or an EMSK. Inner EAP methods which produce neither MSK or EMSK (e.g. EAP-GTC) MUST NOT be used. A party which sees an attempt to use a forbidden inner EAP method MUST either respond with an EAP NAK that suggests one or more acceptable EAP methods, or else respond with a fatal error, and an Error-TLV contain value "Invalid EAP Method" (TBD).

3.3. Intermediate Compound Key Derivations

The intermediate key derivation in TEAPv2 proceeds via the following steps:

* Combine the seed from the previous round with the MSK and/or the EMSK from the current round. If the MSK or EMSK is not available, the relevant field is set to zero. The resulting data is the "RoundSeed", which is used to seed the intermediate key derivations for this round.

* At the conclusion of an inner round, call the TLS-Exporter() function ([RFC8446], Section 7.5) with the RoundSeed as the "context_value" in order to derive intermediate keying data.

* Split the resulting intermediate keying data into two subkeys. One subkey is used as the seed to the next round. The other subkey is the Compound MAC Key (CMK) which is used to calculate the Crypto-Binding TLV.

The following sections explain how the round seed is created (Section 3.3.1), how the seed is used to derive an intermediate key (Section 3.3.2), how the seed is updated after each round (Section 3.3.3), and finally how the master keys are generated from the final round seed (Section 3.5).

3.3.1. Intermediate Key Seeding

The intermediate compound key deriviations for TEAPv2 depend on the following RoundSeed structure. The RoundSeed structure is defined using the same syntax as is used for TLS [RFC8446]:

```
struct {
    opaque PrevRoundKey[40]
    opaque MSK[32];
    opaque EMSK[32]
 } RoundSeed
```

The RoundSeed structure fields have the following definitions:

PrevRoundKey

A key which ties the previous round to the current round.

MSK

The Master Session Key (MSK) from the current round.

EMSK

The Extended Master Session Key (EMSK) from the current round.

At the start of Phase 2, the first 40 octets of the session_key_seed MUST be copied to the PrevRoundKey field of the RoundSeed structure. All other fields MUST be set to zero.

The MSK and EMSK are taken from the inner authentication method for the current round. If the MSK or the EMSK are not available, the corresponding field in RoundSeed is set to zero.

The RoundSeed structure is updated after the DerivedKey structure (Section 3.3.2, below) is calculated. The process to fill in and update the RoundSeed structure is defined below in Section 3.3.3.

3.3.2. Key Derivation

After a successful inner round, a DerivedKey is calculated. The DerivedKey depends on the current value of RoundSeed via the following calculation.

```
DerivedKey = TLS-Exporter(
             "EXPORTER: TEAPv2 Inner Methods Compound Keys",
             RoundSeed, 72)
```

The DerivedKey is 72 octets in length, and is assigned to the following data structure:

```
struct {
    opaque RoundKey[40];
    opaque CMK[32]
} DerivedKey
```

The DerivedKey structure fields have the following definitions:

RoundKey

A key which is associated with this round.

This field is copied to the PrevRoundKey field of the RoundSeed structure.

CMK

The Compound MAC Key (CMK)

The CMK is mixed with with data from the TEAP negotiation to create the Compound-MAC ({#computing-compound-mac}) which is used in the Crypto-Binding TLV.

3.3.3. Updating RoundSeed

The RoundSeed structure MUST be updated after the DerivedKey structure has been calculated. Each field is updated via the process defined below.

Note that the RoundSeed value is updated at the end of each inner round, before the Crypto-Binding TLV is calculated. This update ensures that the final master key calculation (Section 3.5, below) uses a different value for the RoundSeed structure than was used in the last inner exchange.

PrevRoundKey

The RoundKey field from the DerivedKey structure is copied to the PrevRoundKey field.

MSK

If the inner round did not define an MSK, this field is set to all zeros.

If the inner method derived an MSK, then only the first 32 octets of that MSK are copied to this field.

EMSK

If the inner round did not define an EMSK, this field is set to all zeros.

If the inner method derived an EMSK, then only the first 32 octets of that EMSK are copied to this field.

3.4. Computing the Compound-MAC

The Compound-MAC used in the Crypto-Binding TLV is calculated via method as for TEAPv1. That definition is reproduced here verbatim:

Compound-MAC = the first 20 octets of MAC(CMK, BUFFER)

CMK is taken from the DerivedKey structure which was calculated for this round. The definition of BUFFER is the same as in [I-D.ietf-emu-rfc7170bis], Section 6.3 for TEAPv1.

For TEAPv2, only one CMK is derived for each inner message. This limitation also means that only one Compound-MAC is derived for the Crypto-Binding TLV ({#crypto-binding}).

3.5. EAP Master Session Key Generation

The final MSK and ESMK are generated via the following derivation:

```
MSK  = the first 64 octets of TLS-PRF(RoundSeed,
       "Session Key Generating Function")
EMSK = the first 64 octets of TLS-PRF(RoundSeed,
       "Extended Session Key Generating Function")
```

This construction ensures that the cryptographic binding requirements of [RFC6677] are satisfied, while using a much simpler key derivation than was done for TEAPv1.

4. Message Formats

The following sections describe the message formats used in TEAPv2. The fields are transmitted from left to right in network byte order.

4.1. EAP MTU

[RFC3748], Section 3.1 defines a minimum EAP MTU of 1020 octets. However, it defines no maximum MTU, [RFC3748] also provides no way for EAP methods to negotiate an MTU.

Implementation and operational experience has shown that there is a need to define a maximum MTU. Without a defined maximum, EAP peers may assume that the MTU is limited by the local network, which could be as large as 9K octets. Such large EAP packets can have issues when transiting subsequent systems in the network.

For example, the RADIUS protocol often carries EAP, and [RFC2865], Section 3 defines a maximum RADIUS packet length of 4096 octets. This limit can therefore be in conflict with different local network limits seen by the EAP peer.

As a result, TEAPv2 mandates a maximum MTU. TEAPv2 impementations are limited to sending EAP packets which are no larger than 1280 octets in size. That is, the EAP packet Length field ([RFC3748], Section 4) MUST contain a value which is no larger than 1280.

This size was chosen to be small enough that the TEAPv2 packets can transit nearly all networks, but large enough to not require too many message exchanges.

4.2. TEAPv2 Message Format

The TEAPv2 message format is identical to that of TEAPv1 ([I-D.ietf-emu-rfc7170bis], Section 4.1) with only one change: the Ver field is set to "2", to indicate that this is TEAPv2.

4.3. TEAPv2 TLVs

The TEAPv2 TLV format and TLV definitions are identical to that for TEAPv1 ([I-D.ietf-emu-rfc7170bis], Section 4.2), with only the changes and additions noted below.

4.3.1. Crypto-Binding TLV

The format of the TEAPv2 Crypto-Binding TLV is the same as for TEAPv1 ([I-D.ietf-emu-rfc7170bis], Section 4.2.13), with the following changes:

* The Version field MUST set to two (2).

* The Received-Ver field MUST be set to two (2), to indicate TEAPv2.

* The Flags field MUST have value 2, to indicate that only the MSK Compound-MAC is present.

* The Nonce is updated as with TEAPv1.

* The ESMK Compound-MAC field is not used. It SHOULD be set to zeros by the sender. The receiver MUST ignore it.

* The MSK Compound-MAC field is calculated as described above in Section 3.4.

5. Security Considerations

6. IANA Considerations

7. References

7.1. Normative References

[I-D.ietf-emu-rfc7170bis]
DeKok, A., "Tunnel Extensible Authentication Protocol (TEAP) Version 1", Work in Progress, Internet-Draft, draft-ietf-emu-rfc7170bis-22, 28 May 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-emu-rfc7170bis-22>.

[RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, Ed., "Extensible Authentication Protocol (EAP)", RFC 3748, DOI 10.17487/RFC3748, June 2004, <https://www.rfc-editor.org/rfc/rfc3748>.

[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/rfc/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/rfc/rfc8446>.

7.2. Informative References

[RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, DOI 10.17487/RFC2865, June 2000, <https://www.rfc-editor.org/rfc/rfc2865>.

[RFC6677] Hartman, S., Ed., Clancy, T., and K. Hoeper, "Channel-Binding Support for Extensible Authentication Protocol (EAP) Methods", RFC 6677, DOI 10.17487/RFC6677, July 2012, <https://www.rfc-editor.org/rfc/rfc6677>.

[RFC7170] Zhou, H., Cam-Winget, N., Salowey, J., and S. Hanna, "Tunnel Extensible Authentication Protocol (TEAP) Version 1", RFC 7170, DOI 10.17487/RFC7170, May 2014, <https://www.rfc-editor.org/rfc/rfc7170>.

Author’s Address

Alan DeKok
InkBridge Networks
Email: alan.dekok@inkbridge.io