Network Working Group
Internet-Draft
Updates: 5216 (if approved)
Intended status: Standards Track
Expires: July 12, 2018

J. Mattsson
M. Sethi
Ericsson

January 8, 2018

Using EAP-TLS with TLS 1.3
draft-mattsson-eap-tls13-01

Abstract

This document specifies the use of EAP-TLS with TLS 1.3 while remaining backwards compatible with existing implementations of EAP-TLS. TLS 1.3 provides significantly improved security, privacy, and reduced latency when compared to earlier versions of TLS.

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 July 12, 2018.

Table of Contents

1. Introduction 2
1.1. Requirements and Terminology 3
2. Protocol Overview 3
2.1. Overview of the EAP-TLS Conversation 3
2.1.1. Base Case 3
2.1.2. Resumption 5
2.1.3. Termination 6
2.1.4. Privacy 8
2.1.5. Fragmentation 10
2.2. Identity Verification 10
2.3. Key Hierarchy 10
2.4. Parameter Negotiation and Compliance Requirements 10
3. Detailed Description of the EAP-TLS Protocol 11
4. IANA considerations 11
5. Security Considerations 11
5.1. Security Claims 11
5.2. Peer and Server Identities 12
5.3. Certificate Validation 12
5.4. Certificate Revocation 12
5.5. Packet Modification Attacks 12
6. References 12
6.1. Normative References 12
6.2. Informative references 13
Appendix A. Updated references 14
Appendix B. Acknowledgements 15
Authors' Addresses 15

1. Introduction

The Extensible Authentication Protocol (EAP), defined in [RFC3748], provides a standard mechanism for support of multiple authentication methods. EAP-Transport Layer Security (EAP-TLS) [RFC5216] specifies an EAP authentication method with certificate-based mutual authentication and key derivation utilizing the TLS handshake protocol for cryptographic algorithms and protocol version negotiation, mutual authentication and establishment of shared secret keying material. EAP-TLS is widely supported for authentication in IEEE 802.11 [IEEE-802.11] networks (Wi-Fi) using IEEE 802.1X [IEEE-802.1X] and has been selected for certificate based authentication in 3GPP 5G networks. EAP-TLS [RFC5216] references TLS 1.0 [RFC2246] and TLS 1.1 [RFC4346], but works perfectly also with

TLS 1.2 [RFC5246].

Weaknesses found in previous versions of TLS, as well as new requirements for security, privacy, and reduced latency has led to the development of TLS 1.3 [I-D.ietf-tls-tls13], which in large parts

is a complete remodeling of the TLS handshake protocol including a different message flow, different handshake messages, different key schedule, different cipher suites, different resumption, and different privacy protection. This means that significant parts of the normative text in the previous EAP-TLS specification [RFC5216] no longer apply, and aspects such as privacy handling, resumption, and key derivation need to be handled differently.

This document defines how to use EAP-TLS with TLS 1.3 (or higher) and does not change how EAP-TLS is used with older versions of TLS. While this document updates EAP-TLS [RFC5216], it remains backwards compatible with it and existing implementations of EAP-TLS. This document only describes differences compared to [RFC5216].

In addition to the improved security and privacy offered by TLS 1.3, there are significant latency benefits of using EAP-TLS with TLS 1.3. When EAP-TLS is used with support for privacy, TLS 1.3 requires two fewer round-trips when compared to earlier versions of TLS.

1.1. Requirements and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]. Readers are expected to be familiar with the terms and concepts used in EAP-TLS [RFC5216] and TLS 1.3 [I-D.ietf-tls-tls13].

2. Protocol Overview

2.1. Overview of the EAP-TLS Conversation

2.1.1. Base Case

TLS 1.3 changes both the message flow and the handshake messages compared to earlier versions of TLS. Therefore, much of Section 2.1 of RC5216 [RFC5216] does not apply for TLS 1.3 (or higher).

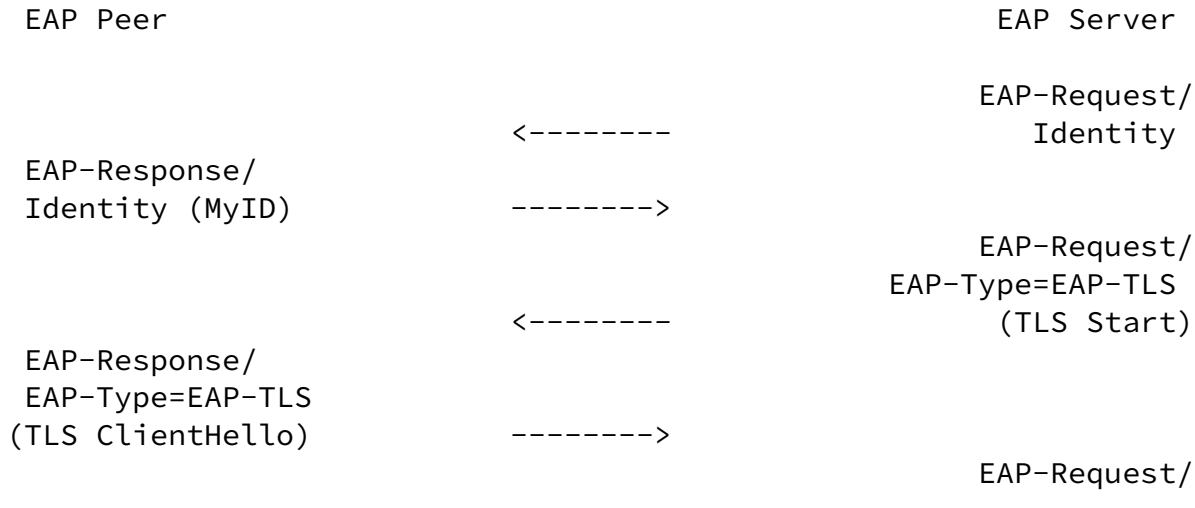
After receiving an EAP-Request packet with EAP-Type=EAP-TLS as described in [RFC5216] the conversation will continue with the TLS handshake protocol encapsulated in the data fields of EAP-Response and EAP-Request packets. When EAP-TLS is used with TLS version 1.3 or higher, the formatting and processing of the TLS handshake SHALL be done as specified in that version of TLS. This document only lists additional and different requirements, restrictions, and processing compared to [I-D.ietf-tls-tls13] and [RFC5216].

The EAP server MUST authenticate with a certificate and SHOULD require the EAP peer to authenticate with a certificate.

Certificates can be of any type supported by TLS including raw public keys. Pre-Shared Key (PSK) authentication SHALL not be used except for resumption. SessionID is deprecated in TLS 1.3 and the EAP server SHALL ignore the legacy_session_id field if TLS 1.3 is negotiated. Resumption is handled as described in Section 2.1.2. After the TLS handshake has completed, the EAP server sends EAP-Success.

As stated in [RFC5216], the TLS cipher suite shall not be used to protect application data. This applies also for early application data. When EAP-TLS is used with TLS 1.3, early application data SHALL NOT be used.

In the case where EAP-TLS with mutual authentication is successful, the conversation will appear as shown in Figure 1.



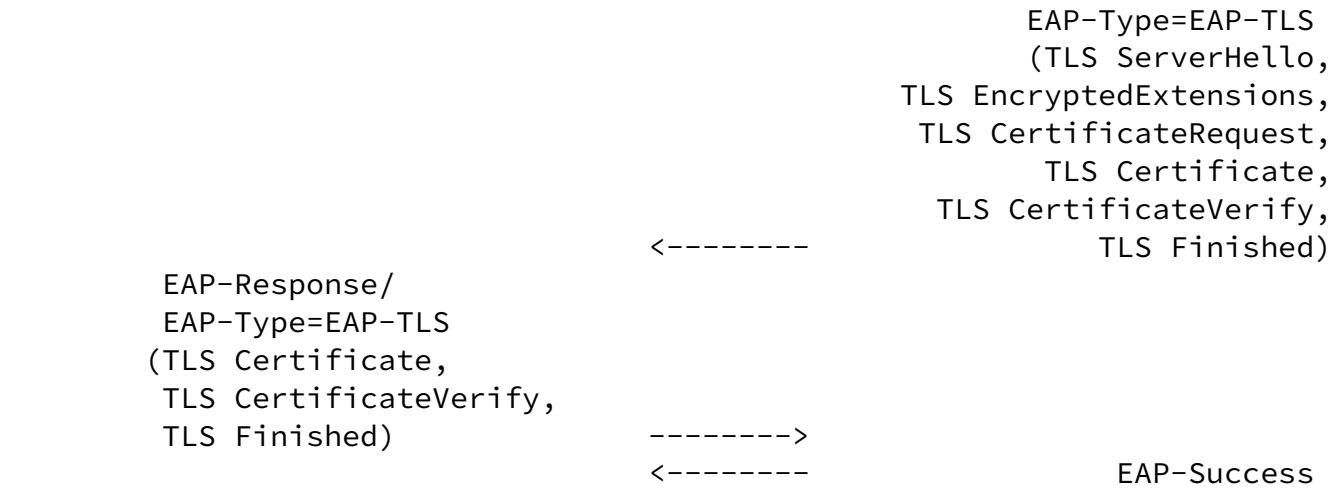
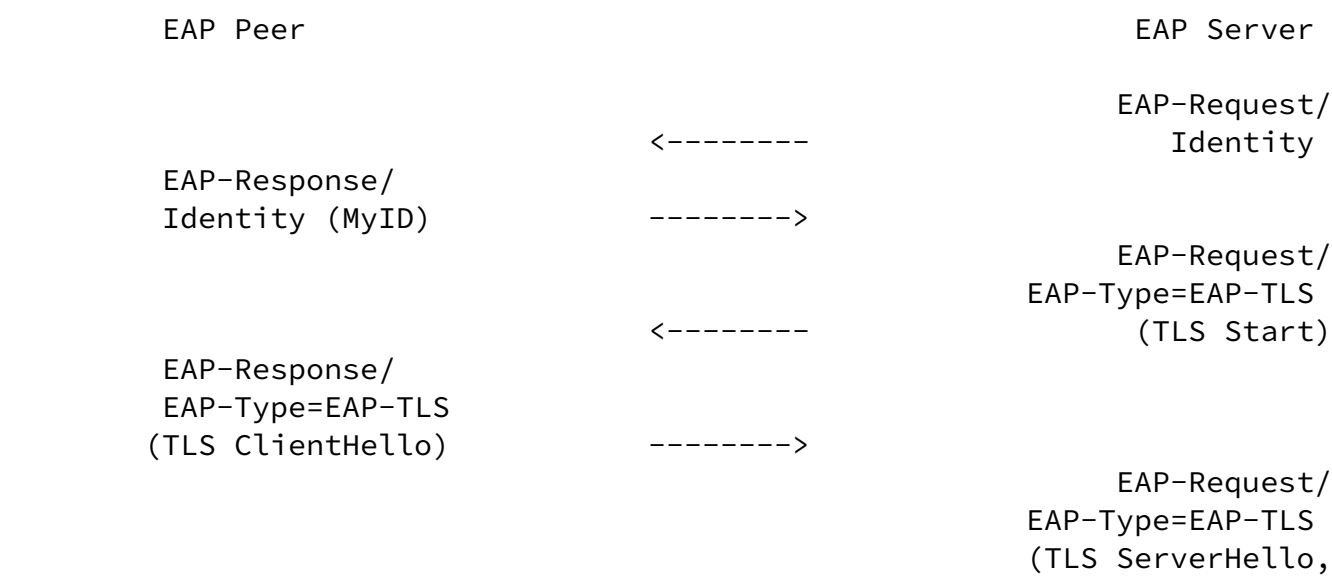


Figure 1: EAP-TLS mutual authentication

When using EAP-TLS with TLS 1.3, the EAP server MUST indicate support of resumption in the initial authentication. To indicate support of resumption, the EAP server sends a NewSessionTicket message

(containing a PSK and other parameters) after it has received the Finished message.

In the case where EAP-TLS with mutual authentication and ticket establishment is successful, the conversation will appear as shown in Figure 2.



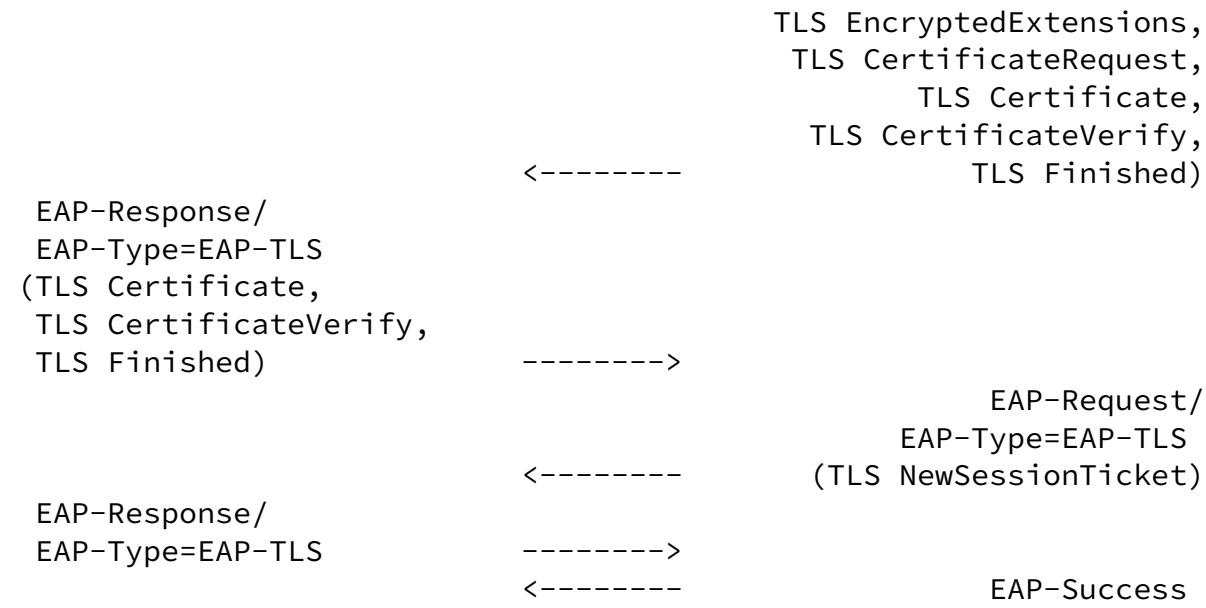


Figure 2: EAP-TLS ticket establishment

2.1.2. Resumption

TLS 1.3 replaces the session resumption mechanisms in earlier versions of TLS with a new PSK exchange. When EAP-TLS is used with TLS version 1.3 or higher, EAP-TLS SHALL use a resumption mechanism compatible with that version of TLS.

For TLS 1.3, resumption is described in Section 2.2 of [I-D.ietf-tls-tls13]. If the client has received a NewSessionTicket message from the server, the client can use the PSK identity received in the ticket to negotiate the use of the associated PSK. If the server accepts it, then the security context of the new connection is tied to the original connection and the key derived from the initial handshake is used to bootstrap the cryptographic state instead of a full handshake. It is left up to the EAP peer whether to use resumption, but a EAP peer SHOULD use resumption as long as it has a valid ticket cached. An EAP server SHOULD accept resumption as long as the ticket is valid, but MAY require a full authentication.

A subsequent authentication using resumption, where both sides authenticate successfully is shown in Figure 3.

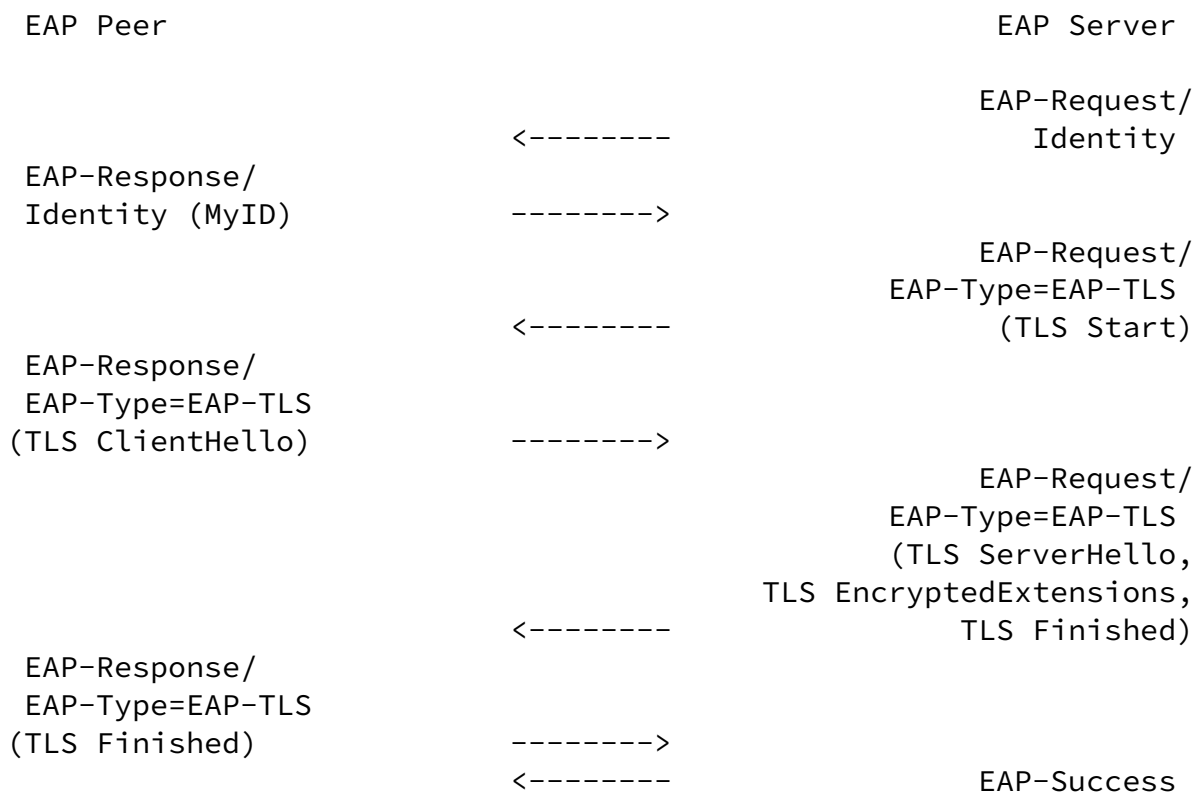


Figure 3: EAP-TLS resumption

2.1.3. Termination

TLS 1.3 changes both the message flow and the handshake messages compared to earlier versions of TLS. Therefore, some normative text in Section 2.1.3 of RC5216 [RFC5216] does not apply for TLS 1.3 or higher. The two paragraphs below replaces the corresponding paragraphs in Section 2.1.3 of RC5216 [RFC5216] when EAP-TLS is used with TLS 1.3 or higher. The other paragraphs in Section 2.1.3 of

RC5216 [RFC5216] still apply with the exception that SessionID is deprecated.

> If the EAP Server authenticates successfully the EAP Peer MUST send an EAP-Response message with EAP-Type=EAP-TLS containing TLS records confirming the processing in the version of TLS used.
>
> If the EAP Peer authenticates successfully the EAP Server MUST

send an EAP-Request packet with EAP-Type=EAP-TLS containing TLS records confirming to the processing in the version of TLS used. The message flow ends with the EAP Server sending a EAP-Success message.

In the case where server authentication is unsuccessful, the conversation will appear as follows:

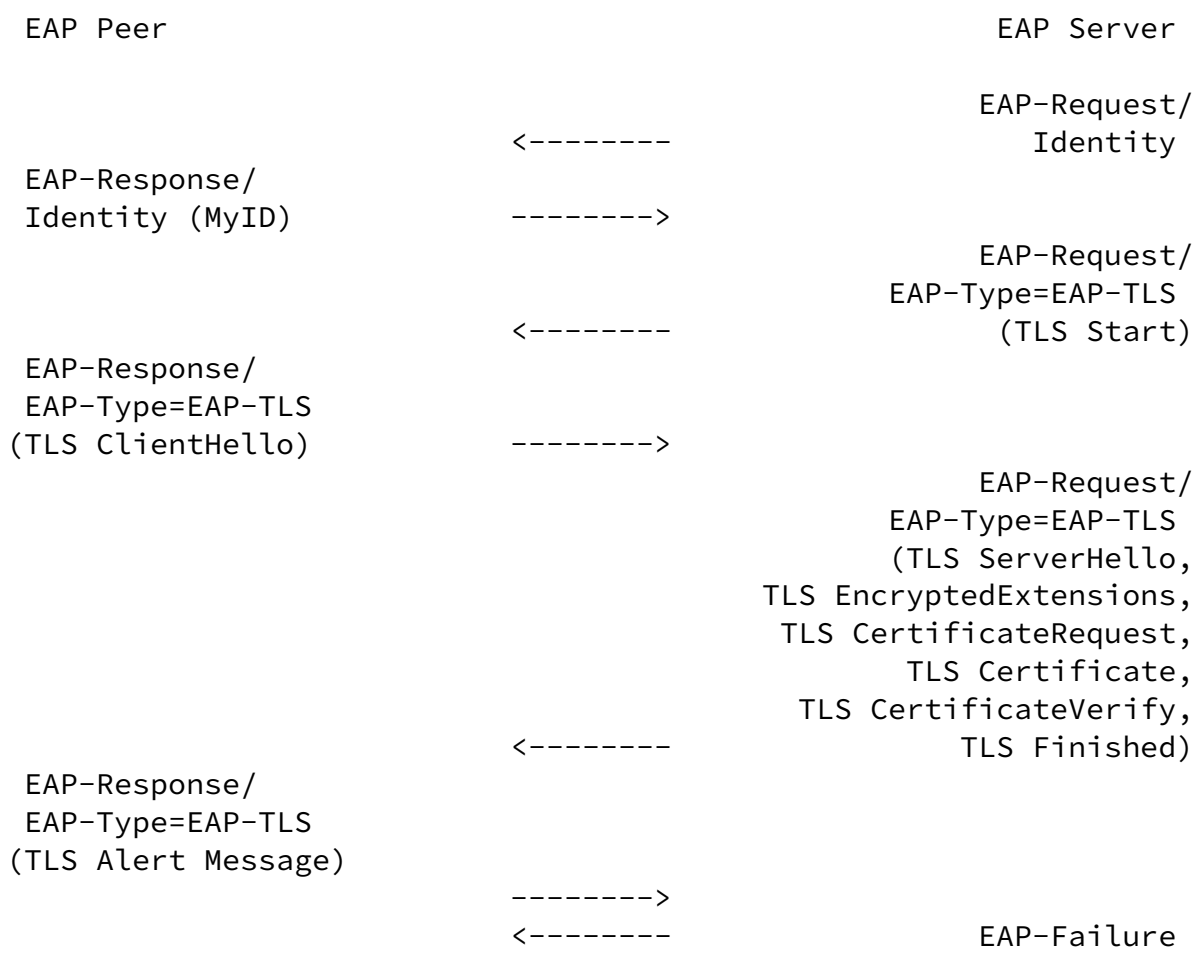


Figure 4: EAP-TLS unsuccessful server authentication

In the case where the server authenticates to the peer successfully, but the peer fails to authenticate to the server, the conversation will appear as follows:

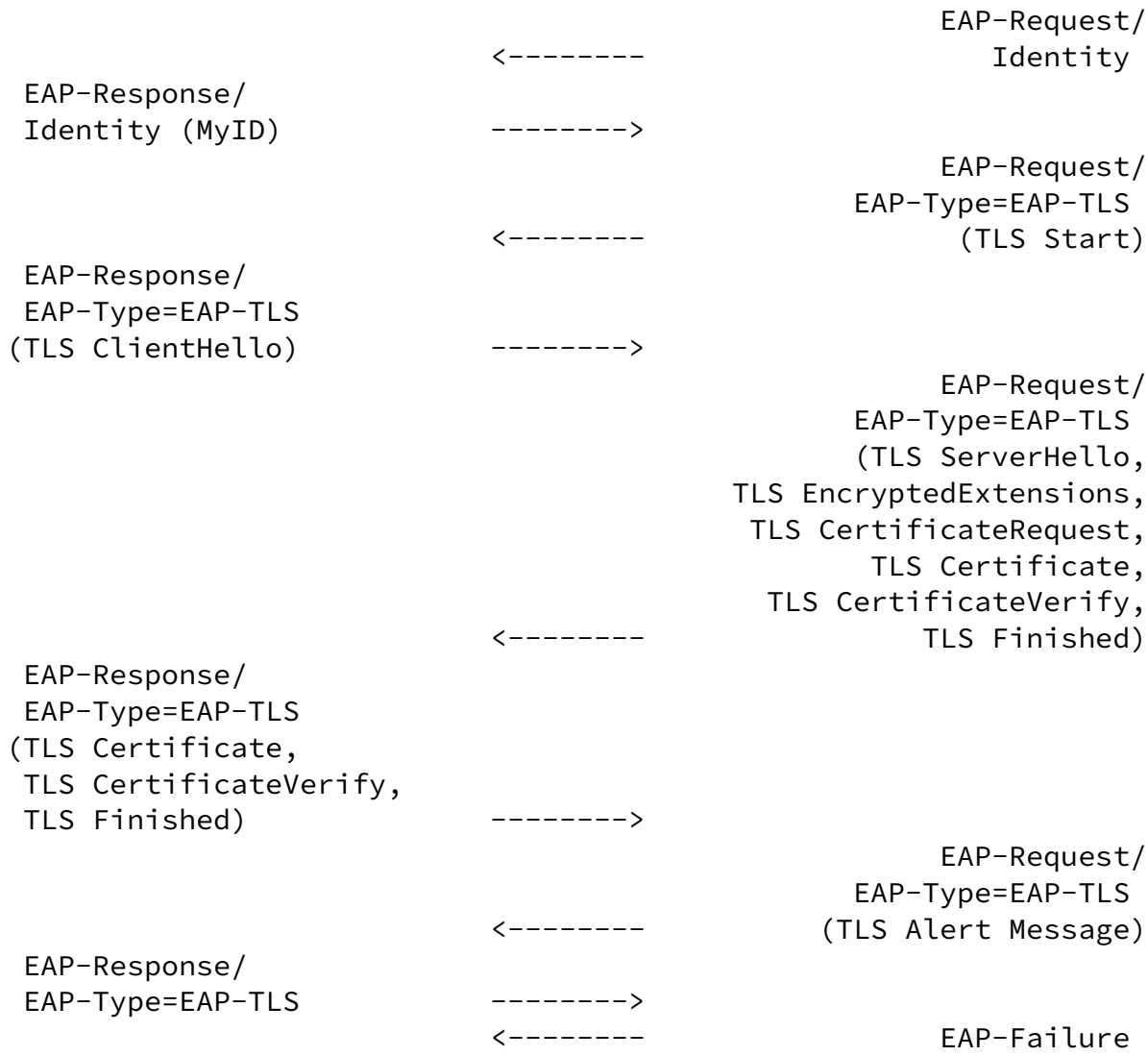


Figure 5: EAP-TLS unsuccessful client authentication

2.1.4. Privacy

TLS 1.3 significantly increases privacy when compared to earlier version of TLS by forbidding cipher suites without confidentiality and encrypting large parts of the TLS handshake including the certificate messages.

EAP-TLS peer and server implementations supporting TLS 1.3 or higher MUST support anonymous NAIs (Network Access Identifiers) ([RFC7542], Section 2.4) and MUST confidentiality protect its identity (e.g. using Anonymous NAIs) when the EAP-TLS server is known to support TLS 1.3 or higher.

As the certificate messages in TLS 1.3 are encrypted, there is no need to send an empty certificate_list or perform a second handshake (as needed by EAP-TLS when with earlier versions of TLS). When EAP-TLS is used with TLS version 1.3 or higher the EAP-TLS peer and EAP-TLS server SHALL follow the processing specified by the used version of TLS. For TLS 1.3 this means that the EAP-TLS peer only sends on empty certificate_list if it does not have an appropriate certificate to send and the EAP-TLS server MAY treat an empty certificate_list as a terminal condition.

When EAP-TLS is used with TLS 1.3 and privacy, no extra round-trips are added and the message flow looks just like a normal message flow with the only difference that an anonymous NAI is used. In the case where EAP-TLS with mutual authentication and privacy is successful, the conversation will appear as shown in Figure 6.

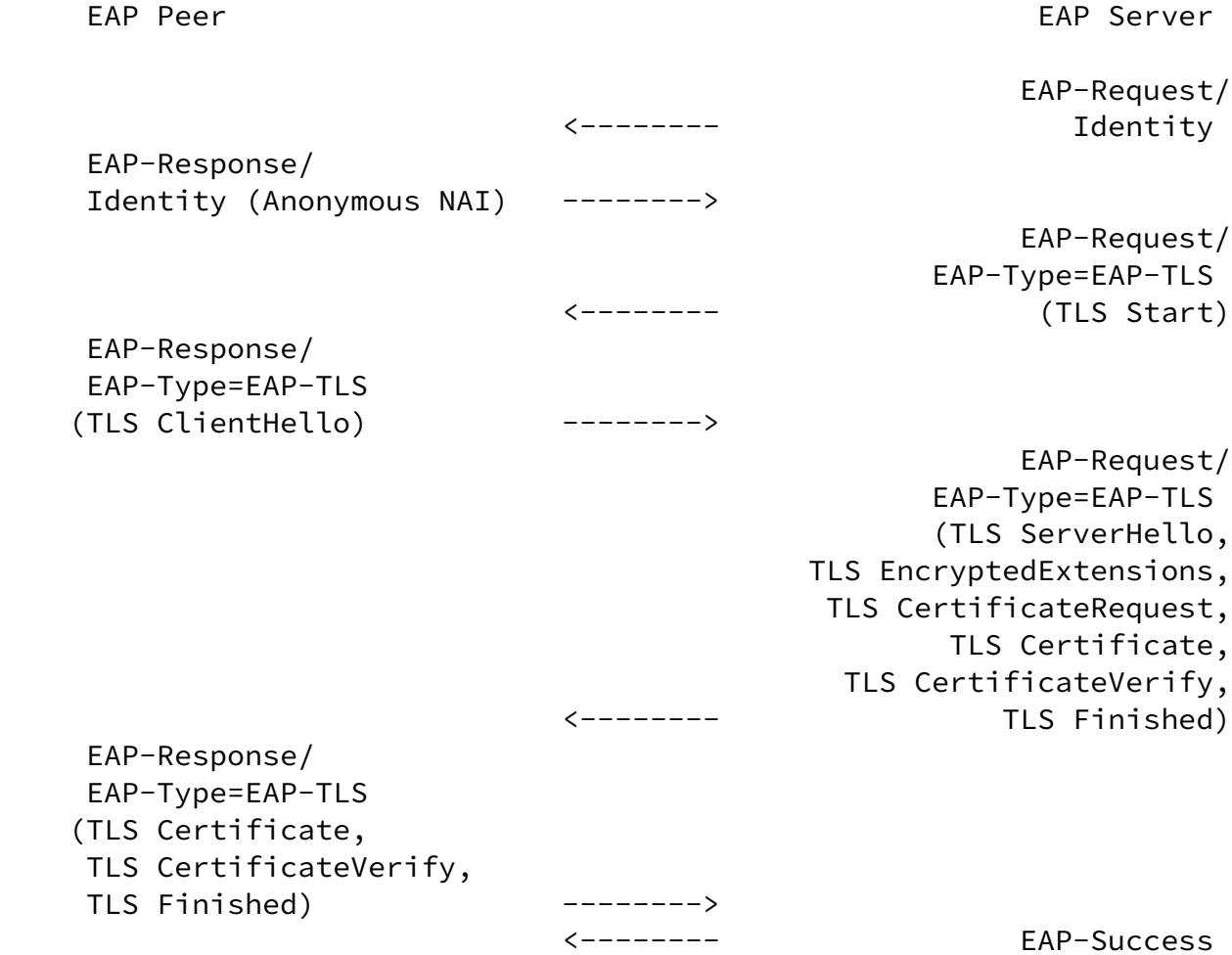


Figure 6: EAP-TLS privacy

2.1.5. Fragmentation

Including ContentType and ProtocolVersion a single TLS record may be up to 16387 octets in length. While Elliptic Curve Cryptography (ECC) was optional for earlier version of TLS, TLS 1.3 mandates support of ECC (see Section 9 of [I-D.ietf-tls-tls13]). To avoid fragmentation, the use of ECC in certificates, signature algorithms, and groups are RECOMMENDED when using EAP-TLS with TLS 1.3 or higher. At a 128-bit security level, this reduces public key sizes from 384 bytes (RSA and DHE) to 32 bytes (ECDHE) and signatures from 384 bytes (RSA) to 64 bytes (ECDSA and EdDSA).

2.2. Identity Verification

No updates to [RFC5216].

2.3. Key Hierarchy

TLS 1.3 replaces the TLS pseudorandom function (PRF) used in earlier versions of TLS with HKDF and completely changes the Key Schedule. The key hierarchies shown in Section 2.3 of [RFC5216] are therefore not correct when EAP-TLS is used with TLS version 1.3 or higher. For TLS 1.3 the key schedule is described in Section 7.1 of [I-D.ietf-tls-tls13].

When EAP-TLS is used with TLS version 1.3 or higher the Key_Material, IV, and Session-Id SHALL be derived from the exporter_master_secret using the TLS exporter interface (for TLS 1.3 this is defined in Section 7.5 of [I-D.ietf-tls-tls13]).

```
Key_Material = TLS-Exporter("client EAP encryption KM", "", 128)
IV           = TLS-Exporter("client EAP encryption IV", "", 64)
Session-Id   = TLS-Exporter("client EAP encryption ID", "", 64)
```

By using the TLS exporter, EAP-TLS can use any TLS 1.3 implementation without having to extract the Master Secret, ClientHello.random, and ServerHello.random in a non-standard way.

All other parameters such as MSK and EMSK are derived as specified in

EAP-TLS [RFC5216], Section 2.3. The use of these keys is specific to the lower layer, as described [RFC5247].

2.4. Parameter Negotiation and Compliance Requirements

TLS 1.3 cipher suites are defined differently than in earlier versions of TLS (see Section B.4 of [I-D.ietf-tls-tls13]), and the cipher suites discussed in Section 2.4 of [RFC5216] can therefore not be used when EAP-TLS is used with TLS version 1.3 or higher. The

requirements on protocol version and compression given in Section 2.4 of [RFC5216] still apply.

When EAP-TLS is used with TLS version 1.3 or higher, the EAP-TLS peers and servers MUST comply with the requirements for the TLS version used. For TLS 1.3 the compliance requirements are defined in Section 9 of [I-D.ietf-tls-tls13].

3. Detailed Description of the EAP-TLS Protocol

No updates to [RFC5216].

4. IANA considerations

There are no IANA impacts in this memo.

5. Security Considerations

5.1. Security Claims

Using EAP-TLS with TLS 1.3 does not change the security claims for EAP-TLS as given in Section 4.1 of [RFC5216]. However, it strengthens several of the claims as described in the following updates to the notes given in Section 4.1 of [RFC5216].

[2] Confidentiality: The TLS 1.3 handshake offers much better confidentiality than earlier versions of TLS by mandating cipher suites with confidentiality and encrypting certificates and some of the extensions, see [I-D.ietf-tls-tls13]. When using EAP-TLS with TLS 1.3, the use of privacy does not cause any additional round-trips.

[3] Key strength: TLS 1.3 forbids all algorithms with known weaknesses including 3DES, CBC mode, RC4, SHA-1, and MD5. TLS 1.3 only supports cryptographic algorithms offering at least 112-bit security, see [I-D.ietf-tls-tls13].

[4] Cryptographic Negotiation: TLS 1.3 increases the number of cryptographic parameters that are negotiated in the handshake. When EAP-TLS is used with TLS 1.3, EAP-TLS inherits the cryptographic negotiation of AEAD algorithm, HKDF hash algorithm, key exchange groups, and signature algorithm, see Section 4.1.1 of [I-D.ietf-tls-tls13].

5.2. Peer and Server Identities

No updates to [RFC5216].

5.3. Certificate Validation

No updates to [RFC5216].

5.4. Certificate Revocation

The OCSP status handling in TLS 1.3 is different from earlier versions of TLS, see Section 4.4.2.1 of [I-D.ietf-tls-tls13]. In TLS 1.3 the OCSP information is carried in the CertificateEntry containing the associated certificate instead of a separate CertificateStatus message as in [RFC4366]. This enables sending OCSP information for all certificates in the certificate chain.

EAP-TLS peers and servers supporting TLS 1.3 SHOULD support Certificate Status Requests (OCSP stapling) as specified in [RFC6066] and Section 4.4.2.1 of [I-D.ietf-tls-tls13]. The use of Certificate Status Requests to determine the current status of the EAP server's certificate is RECOMMENDED.

5.5. Packet Modification Attacks

No updates to [RFC5216].

6. References

6.1. Normative References

[I-D.ietf-tls-tls13]
Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", draft-ietf-tls-tls13-22 (work in progress), November 2017.

[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>.

[RFC2246] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, DOI 10.17487/RFC2246, January 1999, <https://www.rfc-editor.org/info/rfc2246>.

[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/info/rfc3748>.

[RFC5216] Simon, D., Aboba, B., and R. Hurst, "The EAP-TLS Authentication Protocol", RFC 5216, DOI 10.17487/RFC5216, March 2008, <https://www.rfc-editor.org/info/rfc5216>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[RFC6066] Eastlake 3rd, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, DOI 10.17487/RFC6066, January 2011,

<https://www.rfc-editor.org/info/rfc6066>.

[RFC6960] Santesson, S., Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC 6960, DOI 10.17487/RFC6960, June 2013, <https://www.rfc-editor.org/info/rfc6960>.

[RFC7542] DeKok, A., "The Network Access Identifier", RFC 7542, DOI 10.17487/RFC7542, May 2015, <https://www.rfc-editor.org/info/rfc7542>.

6.2. Informative references

[IEEE-802.11]
Institute of Electrical and Electronics Engineers, "IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012) , December 2016.

[IEEE-802.1X]
Institute of Electrical and Electronics Engineers, "IEEE Standard for Local and metropolitan area networks -- Port-Based Network Access Control", IEEE Standard 802.1X-2010 , February 2010.

[RFC2560] Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC 2560, DOI 10.17487/RFC2560, June 1999, <https://www.rfc-editor.org/info/rfc2560>.

[RFC3280] Housley, R., Polk, W., Ford, W., and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, DOI 10.17487/RFC3280, April 2002, <https://www.rfc-editor.org/info/rfc3280>.

[RFC4282] Aboba, B., Beadles, M., Arkko, J., and P. Eronen, "The Network Access Identifier", RFC 4282, DOI 10.17487/RFC4282, December 2005, <https://www.rfc-editor.org/info/rfc4282>.

[RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, DOI 10.17487/RFC4346, April 2006, <https://www.rfc-editor.org/info/rfc4346>.

[RFC4366] Blake-Wilson, S., Nystrom, M., Hopwood, D., Mikkelsen, J., and T. Wright, "Transport Layer Security (TLS) Extensions", RFC 4366, DOI 10.17487/RFC4366, April 2006, <https://www.rfc-editor.org/info/rfc4366>.

[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>.

[RFC5247] Aboba, B., Simon, D., and P. Eronen, "Extensible Authentication Protocol (EAP) Key Management Framework", RFC 5247, DOI 10.17487/RFC5247, August 2008, <https://www.rfc-editor.org/info/rfc5247>.

Appendix A. Updated references

All the following references in [RFC5216] are updated as specified below when EAP-TLS is used with TLS 1.3 or higher.

All references to [RFC2560] are updated with [RFC6960].

All references to [RFC3280] are updated with [RFC5280].

All references to [RFC4282] are updated with [RFC7542].

Appendix B. Acknowledgements

The authors want to thank Alan DeKok, Ari Keraenen, Bernard Aboba, Jari Arkko, and Vesa Toivinen for comments and suggestions on the

draft.

Authors' Addresses

John Mattsson
Ericsson
164 40 Stockholm
Sweden

Email: john.mattsson@ericsson.com

Mohit Sethi
Ericsson
02420 Jorvas
Finland

Email: mohit@piuha.net