Network Working Group
Internet-Draft
Updates: 5216 (if approved)
Intended status: Standards Track
Expires: May 18, 2019

J. Mattsson
M. Sethi
Ericsson

November 14, 2018

Using EAP-TLS with TLS 1.3
draft-ietf-emu-eap-tls13-03

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. EAP-TLS with TLS 1.3 provides significantly improved protection against pervasive monitoring by mandating use of privacy. This document updates RFC 5216.

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 May 18, 2019.

Table of Contents

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 it's the default mechanism for

certificate based authentication in MulteFire [MulteFire] and 3GPP 5G [TS.33.501] 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 [RFC8446], 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] are not applicable to EAP-TLS with TLS 1.3 (or higher). Therefore, aspects such as resumption, privacy handling, and key derivation need to be appropriately addressed for EAP-TLS with TLS 1.3 (or higher).

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 other significant benefits of using EAP-TLS with TLS 1.3. Privacy is achieved without any additional round-trips, and TLS 1.3 introduces more possibilities to reduce fragmentation 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", "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.

Readers are expected to be familiar with the terms and concepts used in EAP-TLS [RFC5216] and TLS 1.3 [RFC8446].

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 RFC5216 [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 [RFC8446] 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 and all Post-Handshake messages have been sent, 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. The EAP server commits to not send any more handshake messages by sending an empty TLS record, see Section 2.5.

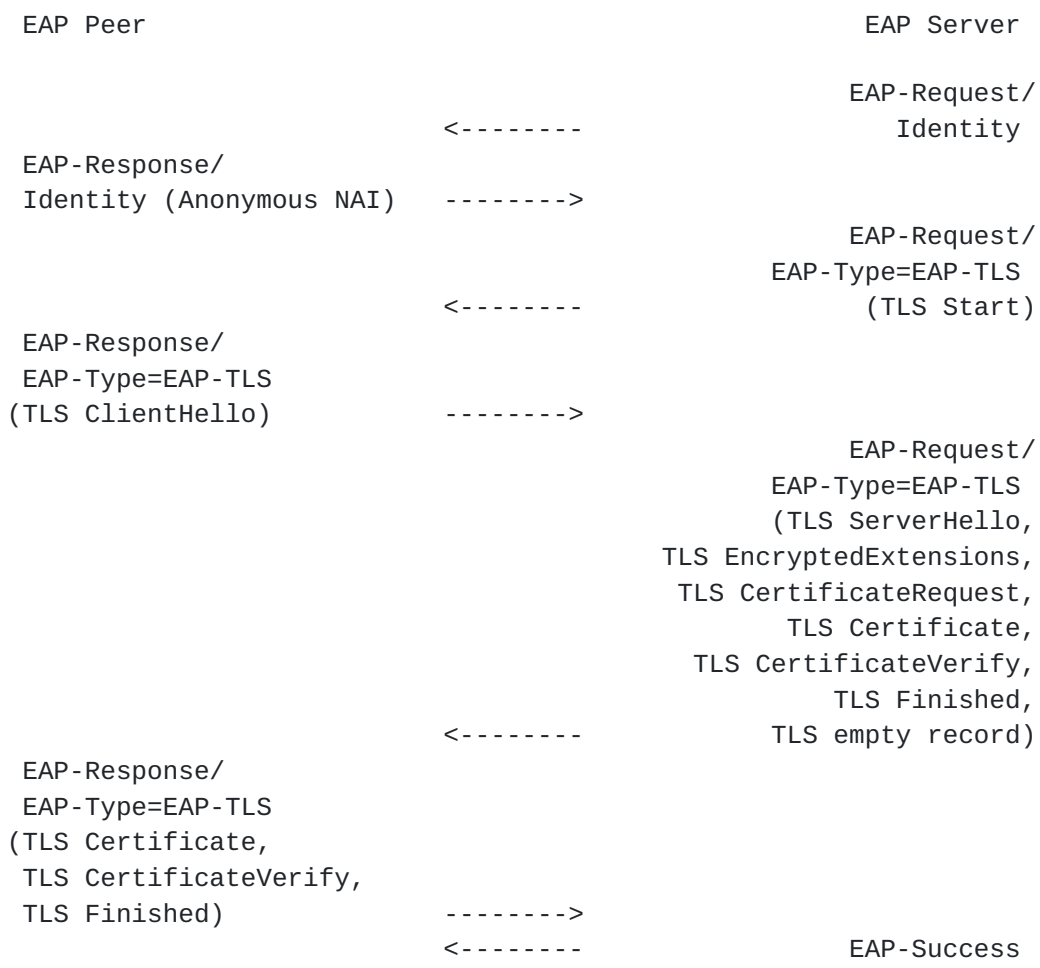

Figure 1: EAP-TLS mutual authentication

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.

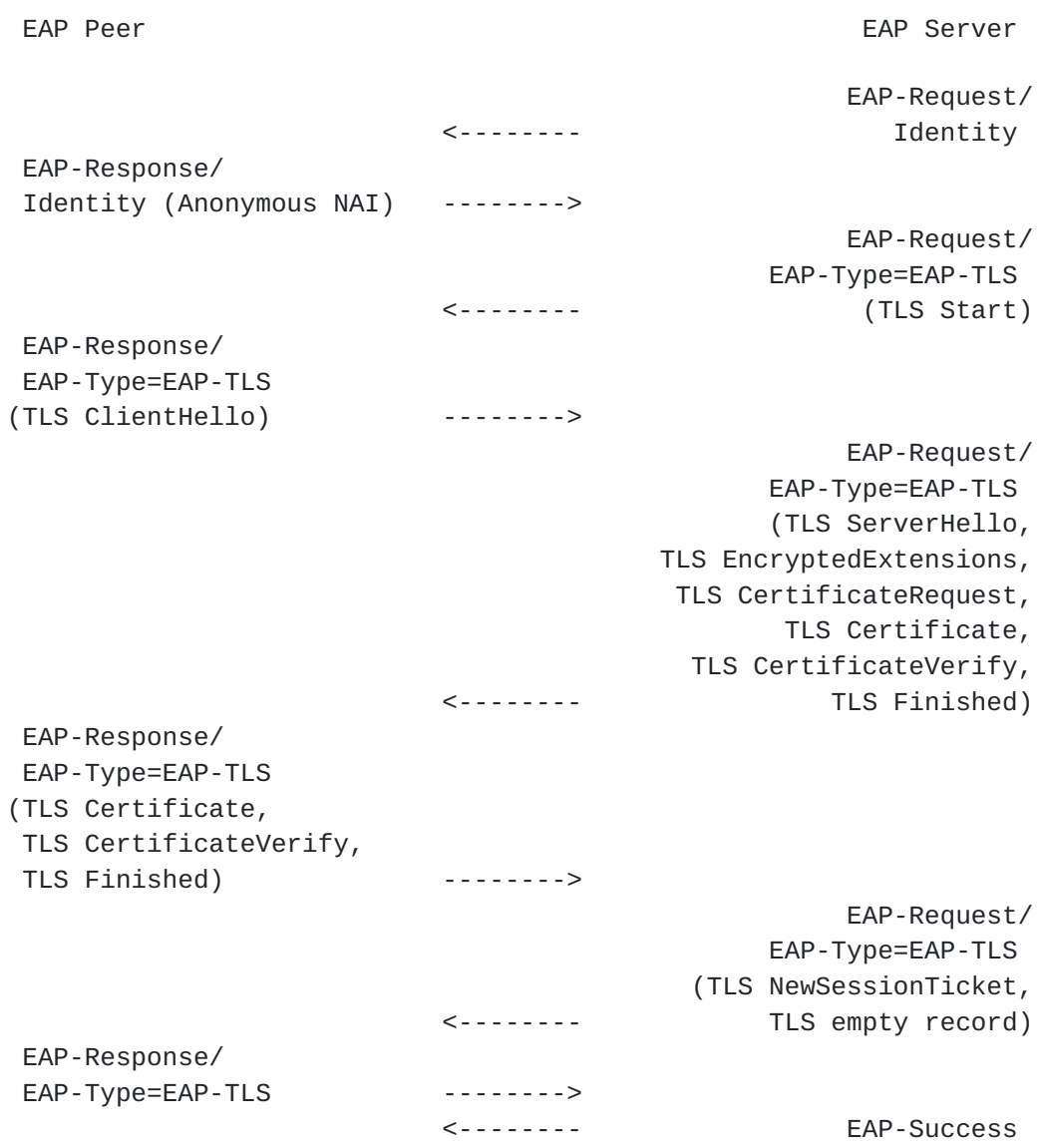

Figure 2: EAP-TLS ticket establishment

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 [RFC8446]. 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 an EAP peer SHOULD use resumption as long as it has a valid ticket cached. It is RECOMMENDED that the EAP server accept resumption as long as the ticket is valid. However, the server MAY choose to require a full authentication.

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

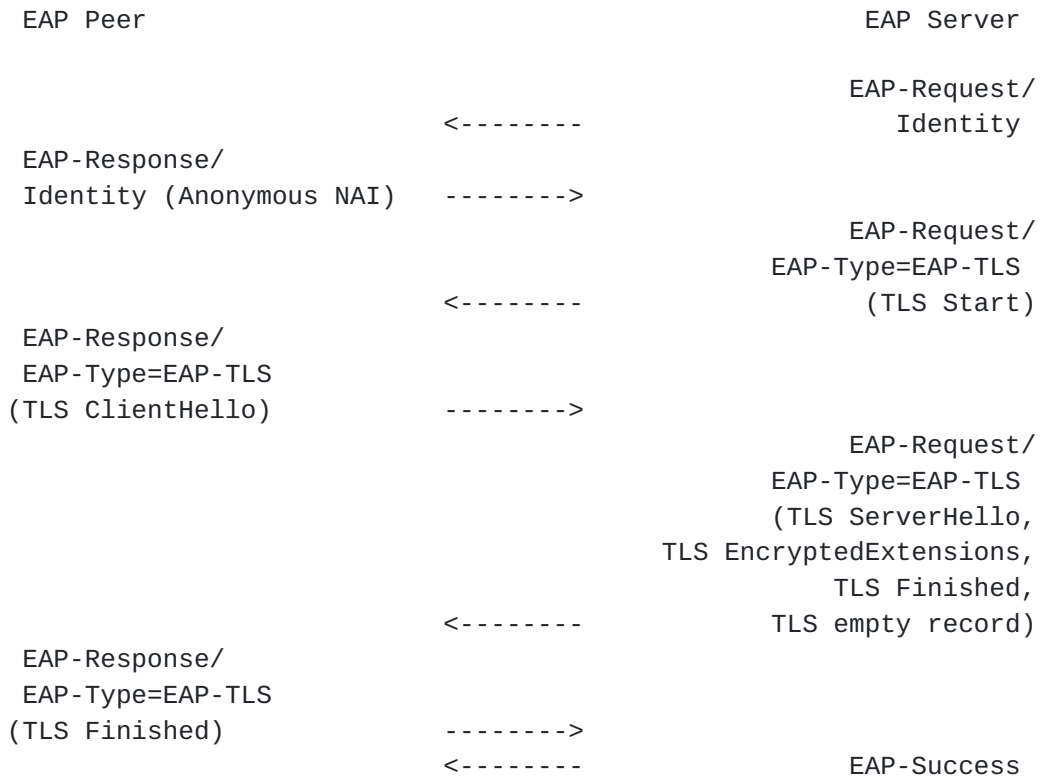

Figure 3: EAP-TLS resumption

As specified in Section 2.2 of [RFC8446], the EAP peer SHOULD supply a "key_share" extension when offering resumption, which allows the EAP server to decline resumption and continue the handshake as a full handshake. The message flow in this case is given by Figure 1 or Figure 2. If the EAP peer did not supply a "key_share" extension when offering resumption, the EAP server needs to reject the ClientHello and the EAP peer needs to restart a full handshake. The message flow in this case is given by Figure 4 followed by Figure 1 or Figure 2.

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 RFC 5216 [RFC5216] does not apply for TLS 1.3 or higher. The two paragraphs below replaces the corresponding paragraphs in Section 2.1.3 of RFC 5216 [RFC5216] when EAP-TLS is used with TLS 1.3 or higher. The other paragraphs in Section 2.1.3 of RFC 5216 [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 conforming to 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 conforming to the version of TLS used. The message flow ends with the EAP server sending an EAP-Success message.

Figures 4, 5, 6, and 7 illustrate message flows in several cases where the EAP peer or EAP server sends a TLS fatal alert message. TLS warning alerts generally mean that the connection can continue normally and does not change the message flow. Note that the party receiving a TLS warning alert may choose to terminate the connection by sending a TLS fatal alert, which may add an extra round-trip, see [RFC8446].

In the case where the server rejects the ClientHello, the conversation will appear as shown in Figure 4.

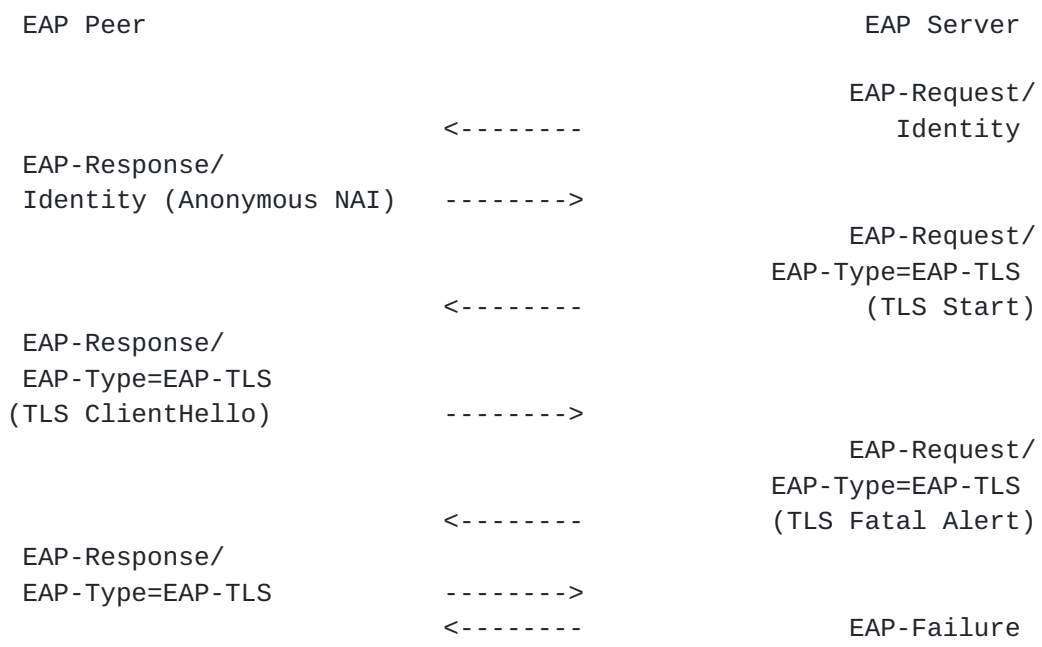

Figure 4: EAP-TLS server rejection of ClientHello

In the case where server authentication is unsuccessful, the conversation will appear as shown in Figure 5.

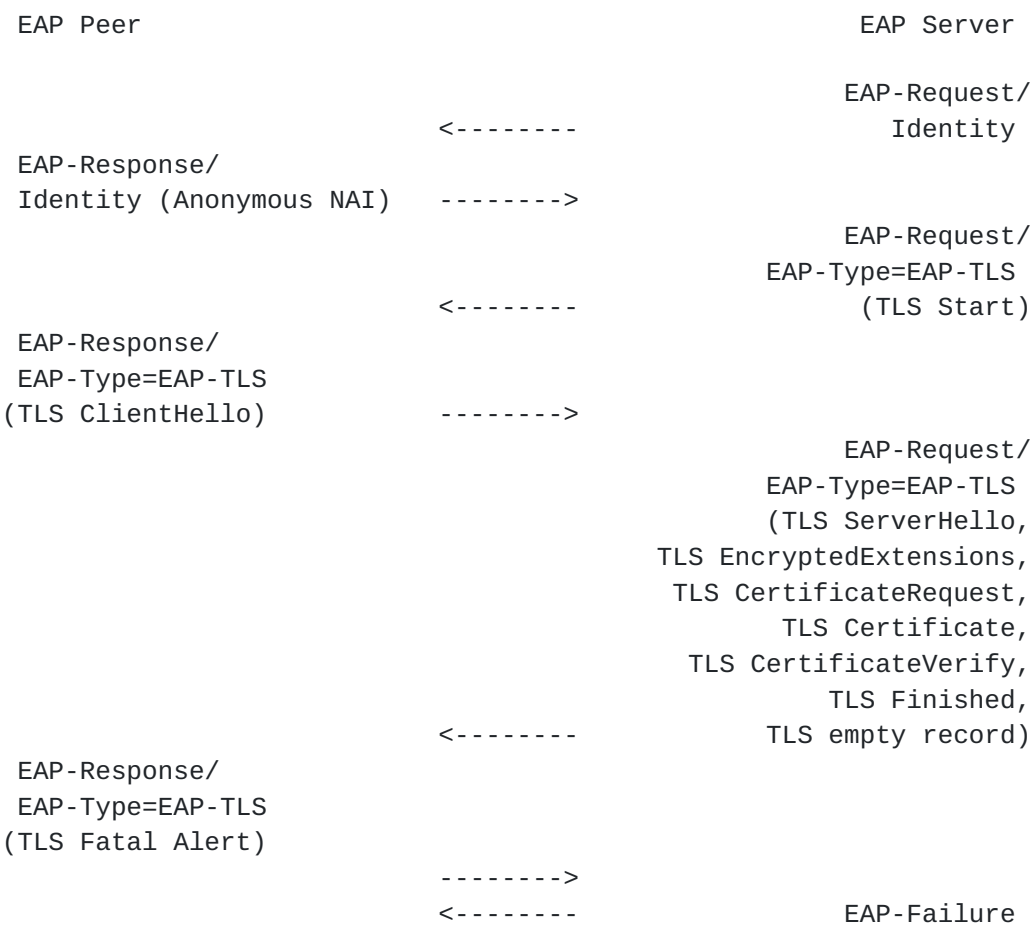

Figure 5: EAP-TLS unsuccessful server authentication

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 shown in Figure 6.

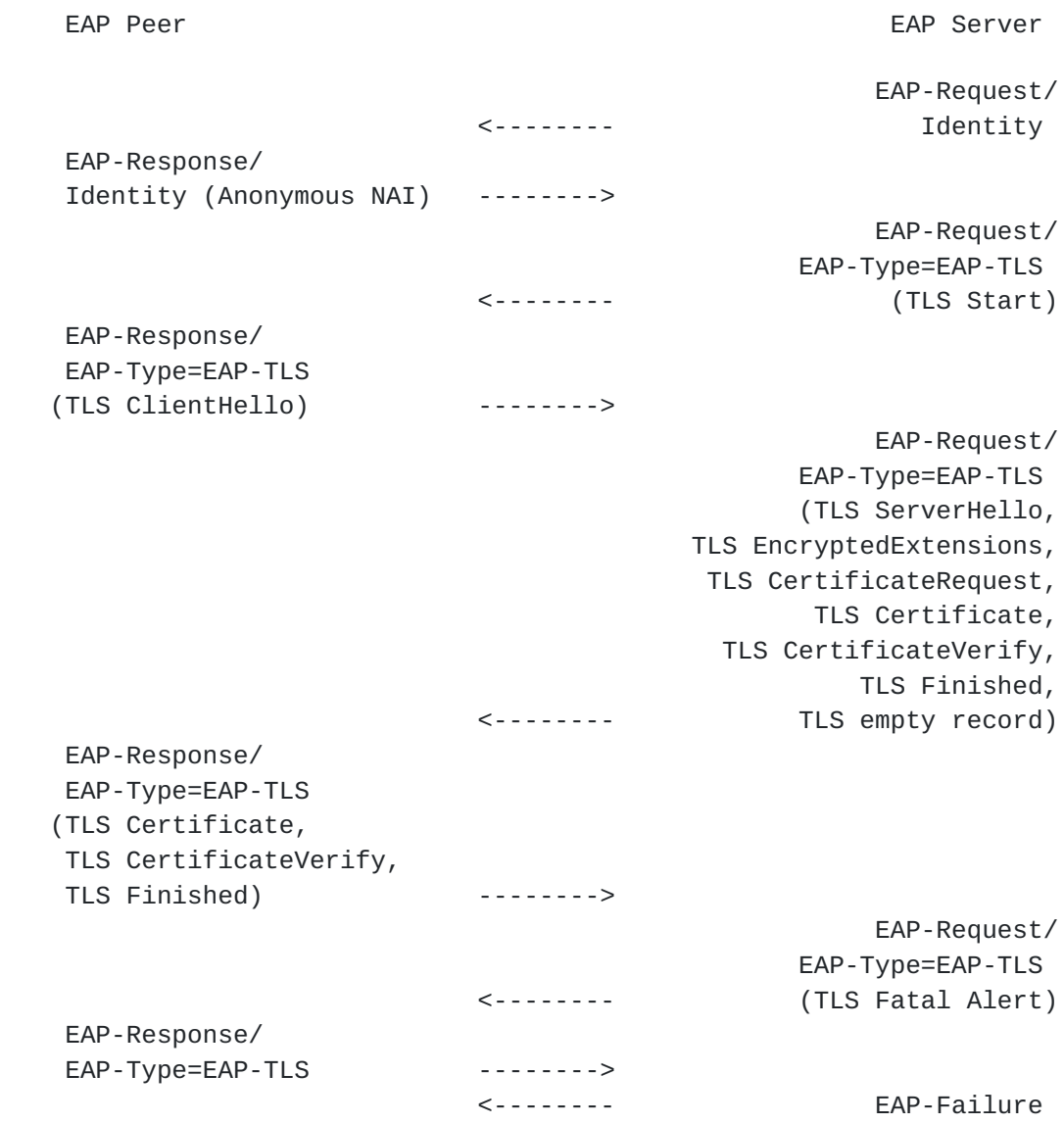

Figure 6: EAP-TLS unsuccessful client authentication

In the case where the client rejects a NewSessionTicket, the conversation will appear as shown in Figure 7.

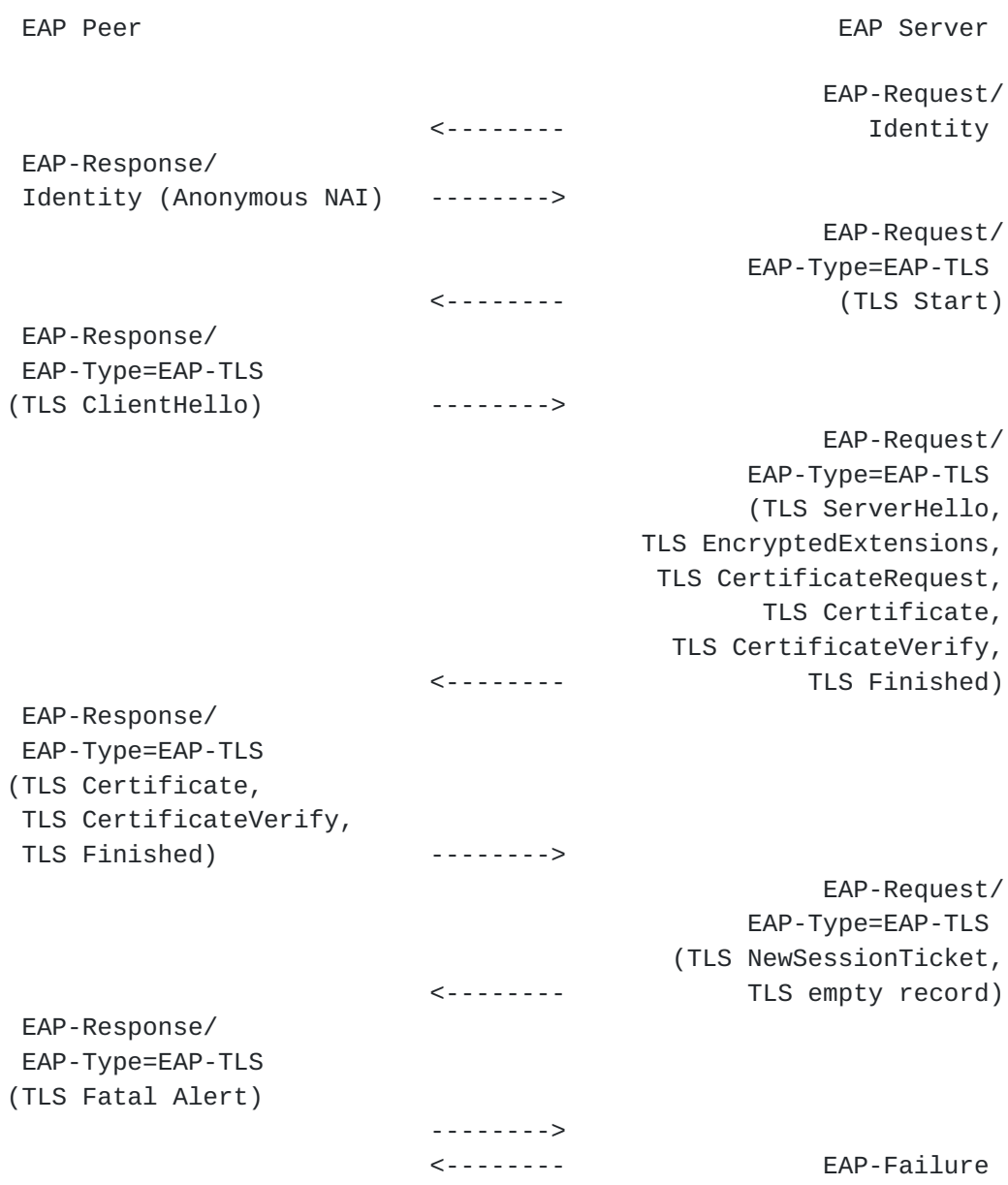

Figure 7: EAP-TLS client rejection of NewSessionTicket

2.1.4. Privacy

TLS 1.3 significantly improves privacy when compared to earlier versions 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) (Section 2.4 in [RFC7542]) and a client supporting TLS 1.3 MUST NOT send its username in cleartext in the Identity Response. It is RECOMMENDED to

use anonymous NAIs, but other solutions where the username is encrypted MAY be used.

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 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 an 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.

EAP-TLS with TLS 1.3 is always used with privacy since this does not add any extra round-trips and the message flow with privacy is just the normal message flow as shown in Figure 1.

2.1.5. Fragmentation

Including ContentType and ProtocolVersion a single TLS record may be up to 16387 octets in length. Some EAP implementations and access networks may limit the number of EAP packet exchanges that can be handled. To avoid fragmentation, it is RECOMMENDED to keep the sizes of client, server, and trust anchor certificates small and the length of the certificate chains short. In addition, it is RECOMMENDED to use mechanisms that reduce the sizes of Certificate messages.

While Elliptic Curve Cryptography (ECC) was optional for earlier version of TLS, TLS 1.3 mandates support of ECC (see Section 9 of [RFC8446]). 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). An EAP-TLS deployment MAY further reduce the certificate sizes by limiting the number of Subject Alternative Names.

Endpoints SHOULD reduce the sizes of Certificate messages by omitting certificates that the other endpoint is known to possess. When using TLS 1.3, all certificates that specifies a trust anchor may be omitted (see Section 4.4.2 of [RFC8446]). When using TLS 1.2 or earlier, only the self-signed certificate that specifies the root certificate authority may be omitted (see Section 7.4.2 of [RFC5246]). EAP-TLS peers and servers SHOULD support and use the Cached Information Extension as specified in [RFC7924]. EAP-TLS peers and servers MAY use other extensions for reducing the sizes of Certificate messages, e.g. certificate compression [I-D.ietf-tls-certificate-compression].

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 [RFC8446].

When EAP-TLS is used with TLS version 1.3 or higher the Key_Material, IV, and Method-Id SHALL be derived from the exporter_master_secret using the TLS exporter interface [RFC5705] (for TLS 1.3 this is defined in Section 7.5 of [RFC8446]).

```
Key_Material = TLS-Exporter("EXPORTER_EAP_TLS_Key_Material", "", 128)
IV           = TLS-Exporter("EXPORTER_EAP_TLS_IV", "", 64)
Method-Id    = TLS-Exporter("EXPORTER_EAP_TLS_Method-Id", "", 64)
Session-Id   = 0x0D || Method-Id
```

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 [RFC8446]), 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 [RFC8446].

2.5. EAP State Machines

TLS 1.3 [RFC8446] introduces Post-Handshake messages. These Post-Handshake messages use the handshake content type and can be sent after the main handshake. One such Post-Handshake message is NewSessionTicket. The NewSessionTicket can be used for resumption. After sending TLS Finished, the EAP server may send any number of Post-Handshake messages in separate EAP-Requests. To decrease the uncertainty for the EAP peer, the following procedure MUST be followed:

When an EAP server has sent its last handshake message (Finished or a Post-Handshake), it commits to not sending any more handshake messages by appending an empty application data record (i.e. a TLS record with TLSPlaintext.type = application_data and TLSPlaintext.length = 0) to the last handshake record. After sending an empty application data record, the EAP server may only send an EAP-Success, an EAP-Failure, or an EAP-Request with a TLS Alert Message.

Note that the use of an empty application data record does not violate the requirement that the TLS cipher suite shall not be used to protect application data, as the application data is the empty string, no application data is protected.

3. Detailed Description of the EAP-TLS Protocol

No updates to [RFC5216].

4. IANA considerations

This section provides guidance to the Internet Assigned Numbers Authority (IANA) regarding registration of values related to the EAP-TLS 1.3 protocol in accordance with [RFC8126].

This memo requires IANA to add the following labels to the TLS Exporter Label Registry defined by [RFC5705]. These labels are used in derivation of Key_Material, IV and Method-Id as defined in Section 2.3:

- "EXPORTER_EAP_TLS_Key_Material"
- "EXPORTER_EAP_TLS_IV"
- "EXPORTER_EAP_TLS_Method-Id"

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 [RFC8446]. When using EAP-TLS with TLS 1.3, the use of privacy is mandatory and 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 [RFC8446].

[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 [RFC8446].

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 [RFC8446]. 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 [RFC8446]. 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].

5.6. Privacy Considerations

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

TLS 1.3 offers much better privacy than earlier versions of TLS as discussed in Section 2.1.4. In this section, we only discuss the privacy properties of EAP-TLS with TLS 1.3. For privacy properties of TLS 1.3 itself, see [RFC8446].

EAP-TLS sends the standard TLS 1.3 handshake messages encapsulated in EAP packets. Additionally, the EAP peer sends an identity in the first EAP-Response. The other fields in the EAP-TLS Request and the EAP-TLS Response packets do not contain any cleartext privacy sensitive information.

When EAP-TLS is used with TLS 1.3, the username part of the identity response is always confidentiality protected (e.g. using Anonymous NAIs). However, as with other EAP methods, even when privacy-friendly identifiers or EAP tunneling is used, the domain name (i.e. the realm) in the NAI is still typically visible. How much privacy sensitive information the domain name leaks is highly dependent on how many other users are using the same domain name in the particular access network. If all EAP peers have the same domain, no additional information is leaked. If a domain name is used by a small subset of the EAP peers, it may aid an attacker in tracking or identifying the user.

An EAP peer with a policy allowing communication with EAP servers supporting only TLS 1.2 (or lower) without privacy and with RSA key exchange is vulnerable to disclosure of the peer username. An active attacker can in this case make the EAP peer believe that an EAP server supporting TLS 1.3 only supports TLS 1.2 (or lower) without privacy. The attacker can simply impersonate the EAP server and negotiate TLS 1.2 (or lower) with RSA key exchange and send an TLS alert message when the EAP peer tries to use privacy by sending an empty certificate message. Since the attacker (impersonating the EAP server) does not provide a proof-of-possession of the private key until the Finished message when RSA key exchange is used, an EAP peer may inadvertently disclose its identity (username) to an attacker.

Therefore, it is RECOMMENDED for EAP peers to not use EAP-TLS with TLS 1.2 (or lower) and RSA based cipher suites without privacy.

5.7. 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 and by mandating the use of privacy, TLS 1.3 offers much better protection against pervasive monitoring. 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 EAP-TLS together with information gathered from other protocols increases the risk of identifying individual users.

6. References

6.1. 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>.

[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>.

[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>.

[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>.

[RFC7924] Santesson, S. and H. Tschofenig, "Transport Layer Security (TLS) Cached Information Extension", RFC 7924, DOI 10.17487/RFC7924, July 2016, <https://www.rfc-editor.org/info/rfc7924>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[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>.

6.2. Informative references

[I-D.ietf-tls-certificate-compression]
Ghedini, A. and V. Vasiliev, "TLS Certificate Compression", draft-ietf-tls-certificate-compression-04 (work in progress), October 2018.

[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.

[MulteFire]
MulteFire, "MulteFire Release 1.0.1 specification", 2017.

[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>.

[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>.

[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>.

[TS.33.501] 3GPP, "Security architecture and procedures for 5G System", 3GPP TS 33.501 15.2.0, September 2018.

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].

Acknowledgments

The authors want to thank Alan DeKok, Ari Keraenen, Bernard Aboba, Eric Rescorla, Jari Arkko, Jim Schaad, Jouni Malinen, and Vesa Torvinen for comments and suggestions on the draft.

Authors' Addresses

John Mattsson
Ericsson
Stockholm 164 40
Sweden

Email: john.mattsson@ericsson.com

Mohit Sethi
Ericsson
Jorvas 02420
Finland

Email: mohit@piuha.net