Stream: Internet Engineering Task Force (IETF)
RFC: 9191
Category: Informational
Published: February 2022
ISSN: 2070-1721
Authors: M. Sethi, *Ericsson*; J. Preuß Mattsson, *Ericsson*; S. Turner, *sn3rd*

RFC 9191
Handling Large Certificates and Long Certificate Chains in TLS-Based EAP Methods

Abstract

The Extensible Authentication Protocol (EAP), defined in RFC 3748, provides a standard mechanism for support of multiple authentication methods. EAP-TLS and other TLS-based EAP methods are widely deployed and used for network access authentication. Large certificates and long certificate chains combined with authenticators that drop an EAP session after only 40 - 50 round trips is a major deployment problem. This document looks at this problem in detail and describes the potential solutions available.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9191.

Copyright Notice

Table of Contents

1. Introduction 3
2. Terminology 4
3. Experience with Deployments 4
4. Handling of Large Certificates and Long Certificate Chains 5
4.1. Updating Certificates and Certificate Chains 5
4.1.1. Guidelines for Certificates 5
4.1.2. Pre-distributing and Omitting CA Certificates 6
4.1.3. Using Fewer Intermediate Certificates 6
4.2. Updating TLS and EAP-TLS Code 7
4.2.1. URLs for Client Certificates 7
4.2.2. Caching Certificates 7
4.2.3. Compressing Certificates 7
4.2.4. Compact TLS 1.3 8
4.2.5. Suppressing Intermediate Certificates 8
4.2.6. Raw Public Keys 8
4.2.7. New Certificate Types and Compression Algorithms 8
4.3. Updating Authenticators 9
5. IANA Considerations 9
6. Security Considerations 9
7. References 9
7.1. Normative References 9
7.2. Informative References 10
Acknowledgements 12
Authors' Addresses 12

1. Introduction

The Extensible Authentication Protocol (EAP), defined in [RFC3748], provides a standard mechanism for support of multiple authentication methods. EAP-TLS [RFC5216] [RFC9190] relies on TLS [RFC8446] to provide strong mutual authentication with certificates [RFC5280] and is widely deployed and often used for network access authentication. There are also many other standardized TLS-based EAP methods such as Flexible Authentication via Secure Tunneling (EAP-FAST) [RFC4851], Tunneled Transport Layer Security (EAP-TTLS) [RFC5281], the Tunnel Extensible Authentication Protocol (TEAP) [RFC7170], as well as several vendor-specific EAP methods such as the Protected Extensible Authentication Protocol (PEAP) [PEAP].

Certificates in EAP deployments can be relatively large, and the certificate chains can be long. Unlike the use of TLS on the web, where typically only the TLS server is authenticated, EAP-TLS deployments typically authenticate both the EAP peer and the EAP server. Also, from deployment experience, EAP peers typically have longer certificate chains than servers. This is because EAP peers often follow organizational hierarchies and tend to have many intermediate certificates. Thus, EAP-TLS authentication usually involves exchange of significantly more octets than when TLS is used as part of HTTPS.

Section 3.1 of [RFC3748] states that EAP implementations can assume a Maximum Transmission Unit (MTU) of at least 1020 octets from lower layers. The EAP fragment size in typical deployments is just 1020 - 1500 octets (since the maximum Ethernet frame size is ~ 1500 bytes). Thus, EAP-TLS authentication needs to be fragmented into many smaller packets for transportation over the lower layers. Such fragmentation not only can negatively affect the latency, but also results in other challenges. For example, some EAP authenticator (e.g., an access point) implementations will drop an EAP session if it has not finished after 40 - 50 round trips. This is a major problem and means that, in many situations, the EAP peer cannot perform network access authentication even though both the sides have valid credentials for successful authentication and key derivation.

Not all EAP deployments are constrained by the MTU of the lower layer. For example, some implementations support EAP over Ethernet "jumbo" frames that can easily allow very large EAP packets. Larger packets will naturally help lower the number of round trips required for successful EAP-TLS authentication. However, deployment experience has shown that these jumbo frames are not always implemented correctly. Additionally, EAP fragment size is also restricted by protocols such as RADIUS [RFC2865], which are responsible for transporting EAP messages between an authenticator and an EAP server. RADIUS can generally transport only about 4000 octets of EAP in a single message (the maximum length of a RADIUS packet is restricted to 4096 octets in [RFC2865]).

This document looks at related work and potential tools available for overcoming the deployment challenges induced by large certificates and long certificate chains. It then discusses the solutions available to overcome these challenges. Many of the solutions require TLS 1.3 [RFC8446]. The IETF has standardized EAP-TLS 1.3 [RFC9190] and is working on specifications such as [TLS-EAP-TYPES] for how other TLS-based EAP methods use TLS 1.3.

2. 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 [RFC3748], EAP-TLS [RFC5216], and TLS [RFC8446]. In particular, this document frequently uses the following terms as they have been defined in [RFC5216]:

Authenticator: The entity initiating EAP authentication. Typically implemented as part of a network switch or a wireless access point.

EAP peer: The entity that responds to the authenticator. In [IEEE-802.1X], this entity is known as the supplicant. In EAP-TLS, the EAP peer implements the TLS client role.

EAP server: The entity that terminates the EAP authentication method with the peer. In the case where no backend authentication server is used, the EAP server is part of the authenticator. In the case where the authenticator operates in pass-through mode, the EAP server is located on the backend authentication server. In EAP-TLS, the EAP server implements the TLS server role.

The document additionally uses the terms "trust anchor" and "certification path" defined in [RFC5280].

3. Experience with Deployments

As stated earlier, the EAP fragment size in typical deployments is just 1020 - 1500 octets. A certificate can, however, be large for a number of reasons:

- It can have a long Subject Alternative Name field.
- It can have long Public Key and Signature fields.
- It can contain multiple object identifiers (OIDs) that indicate the permitted uses of the certificate as noted in Section 5.3 of [RFC5216]. Most implementations verify the presence of these OIDs for successful authentication.
- It can contain multiple organization naming fields to reflect the multiple group memberships of a user (in a client certificate).

A certificate chain (called a certification path in [RFC5280]) in EAP-TLS can commonly have 2 - 6 intermediate certificates between the end-entity certificate and the trust anchor.

The size of certificates (and certificate chains) may also increase manyfold in the future with the introduction of post-quantum cryptography. For example, lattice-based cryptography would have public keys of approximately 1000 bytes and signatures of approximately 2000 bytes.

Many access point implementations drop EAP sessions that do not complete within 40 - 50 round trips. This means that if the chain is larger than ~ 60 kilobytes, EAP-TLS authentication cannot complete successfully in most deployments.

4. Handling of Large Certificates and Long Certificate Chains

This section discusses some possible alternatives for overcoming the challenge of large certificates and long certificate chains in EAP-TLS authentication. Section 4.1 considers recommendations that require an update of the certificates or certificate chains used for EAP-TLS authentication without requiring changes to the existing EAP-TLS code base. It also provides some guidelines that should be followed when issuing certificates for use with EAP-TLS. Section 4.2 considers recommendations that rely on updates to the EAP-TLS implementations and can be deployed with existing certificates. Finally, Section 4.3 briefly discusses what could be done to update or reconfigure authenticators when it is infeasible to replace deployed components giving a solution that can be deployed without changes to existing certificates or code.

4.1. Updating Certificates and Certificate Chains

Many IETF protocols now use elliptic curve cryptography (ECC) [RFC6090] for the underlying cryptographic operations. The use of ECC can reduce the size of certificates and signatures. For example, at a 128-bit security level, the size of a public key with traditional RSA is about 384 bytes, while the size of a public key with ECC is only 32-64 bytes. Similarly, the size of a digital signature with traditional RSA is 384 bytes, while the size is only 64 bytes with the elliptic curve digital signature algorithm (ECDSA) and the Edwards-curve digital signature algorithm (EdDSA) [RFC8032]. Using certificates that use ECC can reduce the number of messages in EAP-TLS authentication, which can alleviate the problem of authenticators dropping an EAP session because of too many round trips. In the absence of a standard application profile specifying otherwise, TLS 1.3 [RFC8446] requires implementations to support ECC. New cipher suites that use ECC are also specified for TLS 1.2 [RFC8422]. Using ECC-based cipher suites with existing code can significantly reduce the number of messages in a single EAP session.

4.1.1. Guidelines for Certificates

The general guideline of keeping the certificate size small by not populating fields with excessive information can help avert the problems of failed EAP-TLS authentication. More specific recommendations for certificates used with EAP-TLS are as follows:

- Object Identifier (OID) is an ASN.1 data type that defines unique identifiers for objects. The OID's ASN.1 value, which is a string of integers, is then used to name objects to which they relate. The Distinguished Encoding Rules (DER) specify that the first two integers always occupy one octet and subsequent integers are base-128 encoded in the fewest possible octets. OIDs are used lavishly in X.509 certificates [RFC5280] and while not all can be avoided, e.g., OIDs for extensions or algorithms and their associate parameters, some are well within the certificate issuer's control:
  - Each naming attribute in a DN (Distinguished Name) has one. DNs are used in the issuer and subject fields as well as numerous extensions. A shallower name will be smaller, e.g.,

C=FI, O=Example, SN=B0A123499EFC as against C=FI, O=Example, OU=Division 1, SOPN=Southern Finland, CN=Coolest IoT Gadget Ever, and SN=B0A123499EFC.

- Every certificate policy (and qualifier) and any mappings to another policy uses identifiers. Consider carefully what policies apply.

- DirectoryString and GeneralName types are used extensively to name things, e.g., the DN naming attribute O= (the organizational naming attribute) DirectoryString includes "Example" for the Example organization and uniformResourceIdentifier can be used to indicate the location of the Certificate Revocation List (CRL), e.g., "http://crl.example.com/sfig2s1-128.crl", in the CRL Distribution Point extension. For these particular examples, each character is a single byte. For some non-ASCII character strings, characters can be several bytes. Obviously, the names need to be unique, but there is more than one way to accomplish this without long strings. This is especially true if the names are not meant to be meaningful to users.
- Extensions are necessary to comply with [RFC5280], but the vast majority are optional. Include only those that are necessary to operate.
- As stated earlier, certificate chains of the EAP peer often follow organizational hierarchies. In such cases, information in intermediate certificates (such as postal addresses) do not provide any additional value and they can be shortened (for example, only including the department name instead of the full postal address).

4.1.2. Pre-distributing and Omitting CA Certificates

The TLS Certificate message conveys the sending endpoint's certificate chain. TLS allows endpoints to reduce the size of the Certificate message by omitting certificates that the other endpoint is known to possess. When using TLS 1.3, all certificates that specify a trust anchor known by the other endpoint 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]). Therefore, updating TLS implementations to version 1.3 can help to significantly reduce the number of messages exchanged for EAP-TLS authentication. The omitted certificates need to be pre-distributed independently of TLS, and the TLS implementations need to be configured to omit these pre-distributed certificates.

4.1.3. Using Fewer Intermediate Certificates

The EAP peer certificate chain does not have to mirror the organizational hierarchy. For successful EAP-TLS authentication, certificate chains **SHOULD NOT** contain more than 4 intermediate certificates.

Administrators responsible for deployments using TLS-based EAP methods can examine the certificate chains and make rough calculations about the number of round trips required for successful authentication. For example, dividing the total size of all the certificates in the peer and server certificate chain (in bytes) by 1020 bytes will indicate the number of round trips required. If this number exceeds 50, then administrators can expect failures with many common authenticator implementations.

4.2. Updating TLS and EAP-TLS Code

This section discusses how the fragmentation problem can be avoided by updating the underlying TLS or EAP-TLS implementation. Note that in some cases, the new feature may already be implemented in the underlying library and simply needs to be enabled.

4.2.1. URLs for Client Certificates

[RFC6066] defines the "client_certificate_url" extension, which allows TLS clients to send a sequence of Uniform Resource Locators (URLs) instead of the client certificate chain. URLs can refer to a single certificate or a certificate chain. Using this extension can curtail the amount of fragmentation in EAP deployments thereby allowing EAP sessions to successfully complete.

4.2.2. Caching Certificates

The TLS Cached Information Extension [RFC7924] specifies an extension where a server can exclude transmission of certificate information cached in an earlier TLS handshake. The client and the server would first execute the full TLS handshake. The client would then cache the certificate provided by the server. When the TLS client later connects to the same TLS server without using session resumption, it can attach the "cached_info" extension to the ClientHello message. This would allow the client to indicate that it has cached the certificate. The client would also include a fingerprint of the server certificate chain. If the server's certificate has not changed, then the server does not need to send its certificate and the corresponding certificate chain again. In case information has changed, which can be seen from the fingerprint provided by the client, the certificate payload is transmitted to the client to allow the client to update the cache. The extension, however, necessitates a successful full handshake before any caching. This extension can be useful when, for example, a successful authentication between an EAP peer and EAP server has occurred in the home network. If authenticators in a roaming network are stricter at dropping long EAP sessions, an EAP peer can use the Cached Information Extension to reduce the total number of messages.

However, if all authenticators drop the EAP session for a given EAP peer and EAP server combination, a successful full handshake is not possible. An option in such a scenario would be to cache validated certificate chains even if the EAP-TLS exchange fails, but such caching is currently not specified in [RFC7924].

4.2.3. Compressing Certificates

The TLS Working Group has standardized an extension for TLS 1.3 [RFC8879] that allows compression of certificates and certificate chains during full handshakes. The client can indicate support for compressed server certificates by including this extension in the ClientHello message. Similarly, the server can indicate support for compression of client certificates by including this extension in the CertificateRequest message. While such an extension can alleviate the problem of excessive fragmentation in EAP-TLS, it can only be used with TLS version 1.3 and higher. Deployments that rely on older versions of TLS cannot benefit from this extension.

4.2.4. Compact TLS 1.3

[cTLS] defines a "compact" version of TLS 1.3 and reduces the message size of the protocol by removing obsolete material and using more efficient encoding. It also defines a compression profile with which either side can define a dictionary of "known certificates". Thus, cTLS could provide another mechanism for EAP-TLS deployments to reduce the size of messages and avoid excessive fragmentation.

4.2.5. Suppressing Intermediate Certificates

For a client that has all intermediate certificates in the certificate chain, having the server send intermediates in the TLS handshake increases the size of the handshake unnecessarily. [TLS-SIC] proposes an extension for TLS 1.3 that allows a TLS client that has access to the complete set of published intermediate certificates to inform servers of this fact so that the server can avoid sending intermediates, reducing the size of the TLS handshake. The mechanism is intended to be complementary with certificate compression.

The Authority Information Access (AIA) extension specified in [RFC5280] can be used with end-entity and CA certificates to access information about the issuer of the certificate in which the extension appears. For example, it can be used to provide the address of the Online Certificate Status Protocol (OCSP) responder from where revocation status of the certificate (in which the extension appears) can be checked. It can also be used to obtain the issuer certificate. Thus, the AIA extension can reduce the size of the certificate chain by only including a pointer to the issuer certificate instead of including the entire issuer certificate. However, it requires the side receiving the certificate containing the extension to have network connectivity (unless the information is already cached locally). Naturally, such indirection cannot be used for the server certificate (since EAP peers in most deployments do not have network connectivity before authentication and typically do not maintain an up-to-date local cache of issuer certificates).

4.2.6. Raw Public Keys

[RFC7250] defines a new certificate type and TLS extensions to enable the use of raw public keys for authentication. Raw public keys use only a subset of information found in typical certificates and are therefore much smaller in size. However, raw public keys require an out-of-band mechanism to bind the public key with the entity presenting the key. Using raw public keys will obviously avoid the fragmentation problems resulting from large certificates and long certificate chains. Deployments can consider their use as long as an appropriate out-of-band mechanism for binding public keys with identifiers is in place. Naturally, deployments will also need to consider the challenges of revocation and key rotation with the use of raw public keys.

4.2.7. New Certificate Types and Compression Algorithms

There is ongoing work to specify new certificate types that are smaller than traditional X.509 certificates. For example, [CBOR-CERT] defines a Concise Binary Object Representation (CBOR) [RFC8949] encoding of X.509 Certificates. The CBOR encoding can be used to compress existing X.509 certificates or for natively signed CBOR certificates. [TLS-CWT] registers a new TLS Certificate type that would enable TLS implementations to use CBOR Web Tokens (CWTs)

[RFC8392] as certificates. While these are early initiatives, future EAP-TLS deployments can consider the use of these new certificate types and compression algorithms to avoid large message sizes.

4.3. Updating Authenticators

There are several legitimate reasons that authenticators may want to limit the number of packets / octets / round trips that can be sent. The main reason has been to work around issues where the EAP peer and EAP server end up in an infinite loop ACKing their messages. Another reason is that unlimited communication from an unauthenticated device using EAP could provide a channel for inappropriate bulk data transfer. A third reason is to prevent denial-of-service attacks.

Updating the millions of already deployed access points and switches is in many cases not realistic. Vendors may be out of business or no longer supporting the products and administrators may have lost the login information to the devices. For practical purposes, the EAP infrastructure is ossified for the time being.

Vendors making new authenticators should consider increasing the number of round trips allowed to 100 before denying the EAP authentication to complete. Based on the size of the certificates and certificate chains currently deployed, such an increase would likely ensure that peers and servers can complete EAP-TLS authentication. At the same time, administrators responsible for EAP deployments should ensure that this 100 round-trip limit is not exceeded in practice.

5. IANA Considerations

This document has no IANA actions.

6. Security Considerations

Updating implementations to TLS version 1.3 allows omitting all certificates with a trust anchor known by the other endpoint. TLS 1.3 additionally provides improved security, privacy, and reduced latency for EAP-TLS [RFC9190].

Security considerations when compressing certificates are specified in [RFC8879].

Specific security considerations of the referenced documents apply when they are taken into use.

7. References

7.1. 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>.

[RFC4851] Cam-Winget, N., McGrew, D., Salowey, J., and H. Zhou, "The Flexible Authentication via Secure Tunneling Extensible Authentication Protocol Method (EAP-FAST)", RFC 4851, DOI 10.17487/RFC4851, May 2007, <https://www.rfc-editor.org/info/rfc4851>.

[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>.

[RFC5281] Funk, P. and S. Blake-Wilson, "Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0)", RFC 5281, DOI 10.17487/RFC5281, August 2008, <https://www.rfc-editor.org/info/rfc5281>.

[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/info/rfc7170>.

[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>.

[RFC9190] Preuß Mattsson, J. and M. Sethi, "EAP-TLS 1.3: Using the Extensible Authentication Protocol with TLS 1.3", RFC 9190, DOI 10.17487/RFC9190, February 2022, <https://www.rfc-editor.org/rfc/rfc9190>.

7.2. Informative References

[CBOR-CERT] Raza, S., Höglund, J., Selander, G., Preuß Mattsson, J., and M. Furuhed, "CBOR Encoded X.509 Certificates (C509 Certificates)", Work in Progress, Internet-Draft, draft-mattsson-cose-cbor-cert-compress-08, 22 February 2021, <https://datatracker.ietf.org/doc/html/draft-mattsson-cose-cbor-cert-compress-08>.

[cTLS] Rescorla, E., Barnes, R., and H. Tschofenig, "Compact TLS 1.3", Work in Progress, Internet-Draft, draft-ietf-tls-ctls-04, 25 October 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-tls-ctls-04>.

[IEEE-802.1X] IEEE, "IEEE Standard for Local and Metropolitan Area NNetworks--Port-Based Network Access Control", DOI 10.1109/IEEESTD.2020.9018454, IEEE Standard 802.1X-2020, February 2020, <https://doi.org/10.1109/IEEESTD.2020.9018454>.

[PEAP] Microsoft Corporation, "[MS-PEAP]: Protected Extensible Authentication Protocol (PEAP)", June 2021.

[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/info/rfc2865>.

[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>.

[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>.

[RFC6090] McGrew, D., Igoe, K., and M. Salter, "Fundamental Elliptic Curve Cryptography Algorithms", RFC 6090, DOI 10.17487/RFC6090, February 2011, <https://www.rfc-editor.org/info/rfc6090>.

[RFC7250] Wouters, P., Ed., Tschofenig, H., Ed., Gilmore, J., Weiler, S., and T. Kivinen, "Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", RFC 7250, DOI 10.17487/RFC7250, June 2014, <https://www.rfc-editor.org/info/rfc7250>.

[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>.

[RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, DOI 10.17487/RFC8032, January 2017, <https://www.rfc-editor.org/info/rfc8032>.

[RFC8392] Jones, M., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "CBOR Web Token (CWT)", RFC 8392, DOI 10.17487/RFC8392, May 2018, <https://www.rfc-editor.org/info/rfc8392>.

[RFC8422] Nir, Y., Josefsson, S., and M. Pegourie-Gonnard, "Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS) Versions 1.2 and Earlier", RFC 8422, DOI 10.17487/RFC8422, August 2018, <https://www.rfc-editor.org/info/rfc8422>.

[RFC8879] Ghedini, A. and V. Vasiliev, "TLS Certificate Compression", RFC 8879, DOI 10.17487/RFC8879, December 2020, <https://www.rfc-editor.org/info/rfc8879>.

[RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <https://www.rfc-editor.org/info/rfc8949>.

[TLS-CWT] Tschofenig, H. and M. Brossard, "Using CBOR Web Tokens (CWTs) in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", Work in Progress, Internet-Draft, draft-tschofenig-tls-cwt-02, 13 July 2020, <https://datatracker.ietf.org/doc/html/draft-tschofenig-tls-cwt-02>.

[TLS-EAP-TYPES] DeKok, A., "TLS-based EAP types and TLS 1.3", Work in Progress, Internet-Draft, draft-ietf-emu-tls-eap-types-04, 22 January 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-emu-tls-eap-types-04>.

[TLS-SIC] Thomson, M., "Suppressing Intermediate Certificates in TLS", Work in Progress, Internet-Draft, draft-thomson-tls-sic-00, 27 March 2019, <https://datatracker.ietf.org/doc/html/draft-thomson-tls-sic-00>.

Acknowledgements

This document is a result of several useful discussions with Alan DeKok, Bernard Aboba, Jari Arkko, Jouni Malinen, Darshak Thakore, and Hannes Tschofening.

Authors' Addresses

Mohit Sethi
Ericsson
FI-02420 Jorvas
Finland
Email: mohit@iki.fi

John Preuß Mattsson
Ericsson
Kista
Sweden
Email: john.mattsson@ericsson.com

Sean Turner
sn3rd
Email: sean@sn3rd.com