TLS Working Group
Internet-Draft
Intended status: Informational
Expires: February 19, 2019

P. Kampanakis, Ed.
Cisco
M. Msahli, Ed.
Telecom ParisTech
August 18, 2018

TLS 1.3 Authentication using ETSI TS 103 097 and IEEE 1609.2 certificates
draft-tls-certieee1609-01.txt

Abstract

This document specifies the use of two new certificate types to authenticate TLS entities. The first type enables the use of a certificate specified by the Institute of Electrical and Electronics Engineers (IEEE) [IEEE1609.2] and the second by the European Telecommunications Standards Institute (ETSI) [TS103097].

Table of Contents

1. Introduction

At present, TLS 1.3 protocol [RFC8446] uses X509 and Raw Public Key in order to authenticate servers and clients. This document describes the use of certificates specified either by the Institute of Electrical and Electronics Engineers (IEEE) [IEEE1609.2] or the European Telecommunications Standards Institute (ETSI) [TS103097]. These standards are defined in order to secure communications in vehicular environments. Existing authentication methods, such as X509 and Raw Public Key, are designed for Internet use, particularly for flexibility and extensibility, and are not optimized for bandwidth and processing time to support delay-sensitive applications. This is why size-optimized certificates that meet the ITS requirements were designed and standardized.

Two new values referring the previously mentioned certificates will be added to the "client_certificate_type" and the "server_certificate_type" extensions defined in [RFC7250].

2. Requirements 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].

3. Extension Overview

The extension format for extended Client Hello and Server Hello uses the "extension_data" field of the "Certificate_Type_Extension" structue defined in RFC7250. The CertificateType structure is an enum with values taken from the TLS 1.3 Certificate Types. In order to negotiate the support of IEEE 1609.2 or ETSI TS 103097 certificate-based authentication, the clients and the servers MAY include the extension of type "client_certificate_type" and "server_certificate_type" in the extended client hello and EncryptedExtensions. The extension_data" field of this extension SHALL contain a list of supported certificate types proposed by the client as provided in figure below:

```
     /* Managed by IANA */
      enum {
          X509(0),
          RawPublicKey(2),
          1609Dot2(?), /* Number 3 will be requested for 1609.2 */
          (255)
          103097(?), /* Number 4 will be requested for 103097 */
          (255)
      } CertificateType;

      struct {
          select (certificate_type) {

               /*  certificate type defined in this document.*/
               case 103097:
               opaque cert_data<1..2^24-1>;

              /* certificate type defined in this document.*/
               case 1609Dot2:
               opaque cert_data<1..2^24-1>;

               /* RawPublicKey defined in RFC 7250*/
              case RawPublicKey:
              opaque ASN.1_subjectPublicKeyInfo<1..2^24-1>;

              /* X.509 certificate defined in RFC 5246*/
              case X.509:
              opaque cert_data<1..2^24-1>;

               };

             Extension extensions<0..2^16-1>;
        } CertificateEntry;
```

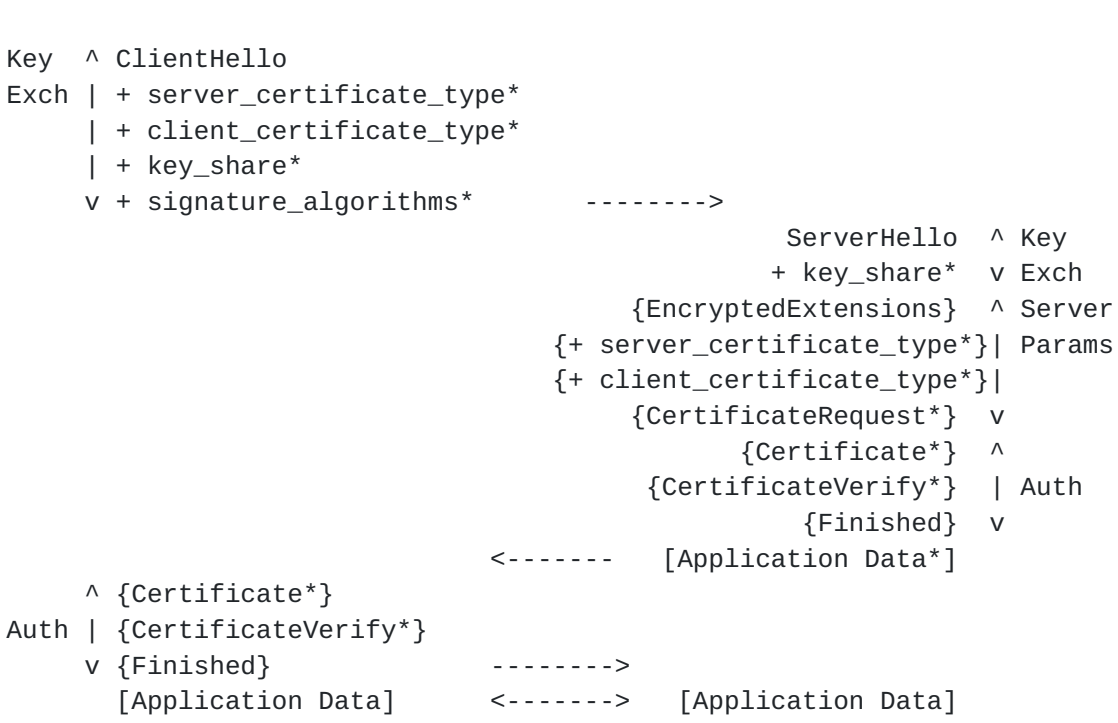
In case where TLS server accepts the described extension, it selects one of the certificate types in the extension described here. Note that a server MAY authenticate the client using other authentication methods. The client MAY at its discretion either continue the handshake, or respond with a fatal message alert.

The end-entity certificate's public key has to be compatible with one of the certificate types listed in extension described here.

Servers aware of the extension described here but not wishing to use it, SHOULD gracefully not proceed with the negotiation.

4. Message Flow

The "client_certificate_type" and "server_certificate_type" messages MUST be sent in handshake phase as illustrated in Figure 1 below. The reply of the server MUST be sent in EncryptedExtensions.



\+ Indicates noteworthy extensions sent in the previously noted message.

* Indicates optional or situation-dependent messages/extensions that are not always sent.

{} Indicates messages protected using keys derived from a [sender]_handshake_traffic_secret.

[] Indicates messages protected using keys derived from [sender]_application_traffic_secret_N.

Figure 1: Message Flow with certificate type extension for Full TLS 1.3 Handshake

4.1. Client Hello

In order to indicate the support of IEEE 1609.2 or ETSI TS 103097 certificates, client MUST include an extension of type "client_certificate_type" and "server_certificate_type"to the extended client hello message. The hello extension mechanism is described in Section 4.1.2 of TLS 1.3 [RFC8446].

The extension 'client_certificate_type' sent in the client hello MAY carry a list of supported certificate types, sorted by client preference. It is a list in the case where the client supports multiple certificate types.

Client MAY respond along with supported certificate by sending a "Certificate" message immediately followed by the "CetificateVerify" message.

4.2. Server Hello

When the server receives the client hello containing the client_certificate_type extension and/or the server_certificate_type extension. The following outcomes are possible:

- The server supports the extension described in this document. It selects a certificate type from the client_certificate_type field in the extended Client Hello and must take into account the client authentication list priority.

- The server does not support the proposed certificate type and terminates the session with a fatal alert of type "unsupported_certificate".

5. Certificate Verification

5.1. IEEE 1609.2 certificates

Verification of an IEEE 1609.2 certificate or certificate chain is described in section 5.5.2 of [IEEE1609.2].

5.2. ETSI TS 103 097 certificates

Verification of ETSI TS 103 097 certificate or certificate chain is described in [TS103097].

6. Examples

Some of exchanged messages examples are illustrated in Figures 2 and 3.

6.1. TLS Server and TLS Client use the 1609Dot2 Certificate

This section shows an example where the TLS client as well as the TLS server use the IEEE 1609.2 certificate. In consequence, both the server and the client populate the client_certificate_type and server_certificate_type with extension IEEE 1609.2 certificates as mentioned in figure 2.

Figure 2: TLS Client and TLS Server use the IEEE 1609.2 certificate

6.2. TLS Server uses the IEEE 1609.2 certificate and TLS Client uses the X 509 certificate

This example shows the TLS authentication, where the TLS client indicates its ability to receive and to validate an IEEE 1609.2 certificate from the server. Therefore, the client populates the server_certificate_type extension with the IEEE 1609.2 certificate type as presented in figure 3.

Figure 3: TLS Server uses the IEEE 1609.2 certificate and TLS Client uses the X 509 certificate

6.3. TLS Server uses the IEEE 1609.2 certificate and TLS Client uses the ETSI TS 103097 certificate

This section shows an example combining an IEEE 1609.2 certificate and an ETSI TS 103097 certificate. The client uses the ETSI TS 103097 certificate for client authentication, and the server provides an IEEE 1609.2 certificate. This exchange starts with the client indicating its ability to process an IEEE 1609.2 certificate if provided by the server. For client authentication, the server indicates that it has selected the ETSI TS 103097 format and requests the certificate from the client as presented in figure 4.

Figure 4: TLS Server uses the IEEE 1609.2 certificate and TLS Client uses the ETSI TS 103097 certificate

7. Security Considerations

This section provides an overview of the basic security considerations which need to be taken into account before implementing the necessary security mechanisms. The security considerations described throughout [RFC8446] apply here as well.

For security considerations in a vehicular environment, the minimal use of any TLS extensions is recommended such as :

- The "client_certificate_type" [IANA value 19] extension who's purpose was previously described in [RFC7250].
- The "server_certificate_type" [IANA value 20] extension who's purpose was previously described in [RFC7250].
- The "SessionTicket" [IANA value 35] extension for session resumption.

8. Privacy Considerations

For privacy considerations in a vehicular environment the use of ETSI TS 103097 and IEEE 1609.2 certificates is recommended for many reasons:

In order to address the risk of a personal data leakage, messages exchanged for V2V communications are signed using IEEE 1609.2 and ETSI TS 103097 pseudonym certificates

The purpose of these certificates is to provide privacy relying on geographical and/or temporal validity criteria, and minimizing the exchange of private data

9. IANA Considerations

Existing IANA references have not been updated yet to point to this document.

IANA is asked to register two new values in the "TLS Certificate Types" registry of Transport Layer Security (TLS) Extensions [TLS-Certificate-Types-Registry], as follows:

- Value: TBD Description: 1609Dot2 Reference: [THIS RFC]

- Value: TBD Description: 103097 Reference: [THIS RFC]

10. Acknowledgements

This document borrows a lot from [draft-serhrouchni-tls-certieee1609-00]. The authors wish to thank Eric Rescola and Ilari Liusvaara and William Whyte for their feedback and suggestions on improving this document. Thanks are due to Sean Turner for his valuable and detailed comments.

11. References

11.1. Normative References

[IEEE1609.2] IEEE, "IEEE Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages", 2016.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", March 1997.

[RFC4492] Blake-Wilson, S., Bolyard, N., Gupta, V., Hawk, C., and B. Moeller, "Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)", May 2006.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", August 2008.

[RFC5746] Rescorla, E., Ray, M., Dispensa, S., and N. Oskov, "Transport Layer Security (TLS) Renegotiation Indication Extension"", February 2010.

[RFC7250] Wouters, P., Tschofenig, H., Weiler, S., and T. Kivinen, "Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", June 2014.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", August 2018.

[TS103097] ETSI, "ETSI TS 103 097 v1.3.1 (2017-10): Intelligent Transport Systems (ITS); Security; Security header and certificate formats", October 2017.

11.2. Informative References

[draft-serhrouchni-tls-certieee1609-00] KAISER, A., LABIOD, H., LONC, B., MSAHLI, M., and A. SERHROUCHNI, "Transport Layer Security (TLS) Authentication using ITS ETSI and IEEE certificates", august 2017.

Appendix A. Co-Authors

- Nancy Cam-Winget
  CISCO, USA
  ncamwing@cisco.com

- Maik Seewald
  CISCO, USA
  maseewal@cisco.com

- Houda Labiod
  Telecom Paristech, France
  houda.labiod@telecom-paristech.fr

- Ahmed Serhrouchni
  Telecom ParisTech
  ahmed.serhrouchni@telecom-paristech.fr

Authors' Addresses

Panos Kampanakis (editor)
Cisco
USA

EMail: EMail: pkampana@cisco.com

Mounira Msahli (editor)
Telecom ParisTech
France

EMail: mounira.msahli@telecom-paristech.fr