Transport Layer Security   Y. Poeluev
Internet Draft   TrustPoint Innovation Technologies
Intended status: Standards Track   W. Ford
Expires: September 2015   TrustPoint Innovation Technologies
March 23, 2015

Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Authentication Using M2M Certificate
draft-ypoeluev-tls-m2mcertificate-00.txt

Status of this Memo

Copyright Notice

Abstract

This memo defines Transport Layer Security (TLS) extensions and associated semantics that allow clients and servers to negotiate the use of M2M certificates for a TLS/DTLS session, and specifies how to transport M2M certificates via TLS/DTLS. It also defines the registry for non-X.509 certificate types.

The X.509 public key certificate format is overly verbose for Internet-of-Things (IoT) constrained environments, where nodes with limited memory and networks with limited bandwidth are not uncommon. The Machine-to-Machine (M2M) certificate format is a pruned down and encoding-optimized replacement for X.509, which reuses much of the X.509 semantics but reduces certificate sizes by typically 40%.

Table of Contents

1. Introduction

This document specifies a way to negotiate the use of M2M certificates [M2M-CER] for a TLS/DTLS session, and specifies how to transport M2M certificates via TLS/DTLS. The proposed extensions are backward compatible with the current TLS/DTLS specification, so that existing client and server implementations that make use of X.509 certificates are not affected.

The predominant public key certificate format has for many years been the X.509 format [RFC5280]. X.509 was designed to be extremely flexible and open-ended, in an environment of RSA and DSA signature technologies. X.509 is not, however, a good certificate format for Internet-of-Things constrained environments, where nodes with limited memory and networks with limited bandwidth are not uncommon. With RSA and DSA technologies, overheads in the certificate format were

comparatively inconsequential because the large key and signature fields were the dominant certificate components size-wise. However, with the much more efficient ECC technology used today, the certificate format overheads become a very important factor in making certificates efficient enough for low bandwidth constrained applications. In essence, the X.509 certificate format is too verbose for these applications.

The Machine-to-Machine (M2M) certificate format was designed to satisfy the above objectives. Essentially what was done was to strip down the X.509 format to eliminate features that are not needed today, while optimizing the encoding. The result is a certificate format that typically reduces certificate size by about 40% compared with X.509.

The M2M certificate format has been adopted by the NFC Forum for Near Field Communications (NFC) signatures, and published by that organization [NFC-SIG]. However it is a general-purpose design, which is equally applicable to Internet-of-Things (IoT) applications.

We are proposing that IETF recognize the M2M format as an optional replacement for X.509 in TLS Protocol [RFC5246] and DTLS Protocol [RFC6347] specifications. A companion Internet-Draft defines the M2M format [M2M-CER].

2. Terminology

This document uses the same notation and terminology used in the TLS Protocol specification [RFC5246] and DTLS Protocol specification [RFC6347].

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. Changes to the Handshake Message Contents

This section describes the changes to the TLS/DTLS handshake message contents when M2M certificates are to be used for authentication. When the document refers to the TLS extensions [RFC5246], the same extensions may be used for the DTLS handshake.

3.1. Client Hello

In order to indicate the support of multiple certificate types, clients MUST include an extension of type "cert_type" to the extended client hello message. The "cert_type" TLS extension is assigned the

value of 9 from the TLS ExtensionType registry. This value is used as the extension number for the extensions in both the client hello message and the server hello message. The hello extension mechanism is described in [RFC5246].

This extension carries a list of supported certificate types the client can use, sorted by client preference. This extension MUST be omitted if the client only supports X.509 certificates. The "extension_data" field of this extension contains a CertificateTypeExtension structure. Note that the CertificateTypeExtension structure is being used both by the client and the server, even though the structure is only specified once in this document. Reusing a single specification for both client and server is common in other specifications, such as the TLS protocol itself [RFC5246].

```
      enum { client, server } ClientOrServerExtension;

      enum { X.509(0), OpenPGP(1), RawPublicKey(2), M2M(3), (255) }

         CertificateType;

      struct {

         select(ClientOrServerExtension) {

            case client:

               CertificateType certificate_types<1..2^8-1>;

            case server:

               CertificateType certificate_type;

         }

      } CertificateTypeExtension;
```

No new cipher suites are required to use M2M certificates. All existing cipher suites that support a key exchange method compatible with the key in the certificate can be used in combination with M2M certificates.

3.2. Server Hello

If the server receives a client hello that contains the "cert_type" extension and chooses a cipher suite that requires a certificate,

then two outcomes are possible. The server MUST either select a certificate type from the certificate_types field in the extended client hello or terminate the session with a fatal alert of type "unsupported_certificate".

The certificate type selected by the server is encoded in a CertificateTypeExtension structure, which is included in the extended server hello message using an extension of type "cert_type". Servers that only support X.509 certificates MAY omit including the "cert_type" extension in the extended server hello.

3.3. Server and Client Certificates

The contents of the certificate message sent from server to client and vice versa are determined by the negotiated certificate type and the selected cipher suite's key exchange algorithm.

When present in the TLS/DTLS handshake, M2M certificates must be encoded using Distinguished Encoding Rules (DER) [X.680].

To carry the M2M certificate within the TLS/DTLS handshake, the Certificate payload is used as a container, as shown in Figure 1. The shown Certificate structure is an adaptation of its original form [RFC5246].

```
      opaque ASN.1Cert<1..2^24-1>;

      struct {

         select(certificate_type){

            // M2M certificate type defined [M2M-CER]

            case M2M:

               ASN.1Cert m2m_certificate_list<0..2^24-1>;


            // X.509 certificate defined in [RFC5246]

            case X.509:

               ASN.1Cert certificate_list<0..2^24-1>;

            // Additional certificate type based on
```

```
            // "TLS Certificate Types" subregistry

          };

       } Certificate;
```

Figure 1: Certificate Payload as a Container for the M2M certificate

3.4. Other Handshake Messages

All the other handshake messages are identical to the TLS/DTLS specifications, [RFC5246]/[RFC6347].

4. Security Considerations

All security considerations discussed in [RFC5246], [RFC6066], and [RFC4880] apply to this document. Considerations about the use of the web of trust or identity and certificate verification procedures are outside the scope of this document. These are considered issues to be handled by the application layer protocols.

The protocol for certificate type negotiation is identical in operation to cipher suite negotiation as described in the TLS specification [RFC5246], with the addition of default values when the extension is omitted. Since those omissions have a unique meaning and the same protection is applied to the values as with cipher suites, it is believed that the security properties of this negotiation are the same as with cipher suite negotiation.

5. IANA Considerations

This document uses a registry and the "cert_type" extension originally defined in [RFC6091].

In order to support M2M certificates the "TLS Certificate Types" registry established by [RFC6091] and [RFC7250] will need to be updated in the following ways:

1. Value 0 (X.509), value 1 (OpenPGP), value 2 (RawPublicKey), and value 3 (M2M) are defined in this document.

2. Values from 4 through 223 decimal inclusive are assigned via "RFC Required" [RFC5226].

3. Values from 224 decimal through 255 decimal inclusive are reserved for Private Use [RFC5226].

6. Conclusions

The IETF, Transport Layer Security and other applicable Working Groups are encouraged to adopt the M2M certificate format as an optional alternative to the X.509 format in all applications in the Internet-of-Things space. There are significant size and bandwidth savings and no significant loss of features of practical importance.

7. References

7.1. Normative References

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5280] Cooper, D., et al, "Internet Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[RFC6091] N. Mavrogiannopoulos and D. Gillmor, "Using OpenPGP Keys for Transport Layer Security (TLS) Authentication", RFC 6091, February 2011.

[RFC6347] E. Rescorla and N. Modadugu, "Datagram Transport Layer Security Version 1.2", January 2012.

[RFC7250] P. Wouters, Ed., et al, "Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Security (DTLS)", RFC 7250, June 2014.

[NFC-SIG] NFC Forum, Signature Record Type Definition, Technical Specification, V2.0, 2014. http://nfc-forum.org/our-work/specifications-and-application-documents/specifications/nfc-forum-technical-specifications/

[X.690] ITU-T Recommendation X.690: ISO/IEC 8825-1:2002, Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER), 2002.

[M2M-CER] W. Ford and Y. Poeluev, "The Machine-to-Machine (M2M) Public Key Certificate Format", draft-ford-m2mcertificate-00, February 2014.

8. Acknowledgments

Recognition is due to Rob Lambert for his critical reviews of the specification.

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Yuri Poeluev
TrustPoint Innovation Technologies, Ltd.
450 Phillip St., Suite 101
Waterloo, ON, Canada, N2L 5J2

Email: ypoeluev@trustpointinnovation.com

Warwick Ford
TrustPoint Innovation Technologies, Ltd.
700 S Monarch St Unit 203,
Aspen, CO 81611

Email: wford@wyltan.com