Transport Layer Security
Internet-Draft
Intended status: Informational
Expires: July 03, 2013

D. Thakore
CableLabs

January 2013

Transport Layer Security (TLS) Authorization Using DTCP Certificate draft-dthakore-tls-authz-02

Abstract

This document specifies the use of DTCP certificate as an authorization extension in the Transport Layer Security Handshake Protocol, according to guidelines in RFC 5878. Extensions carried in the client and server Hello messages confirm that both parties support the desired authorization data types. Then if supported by both the client and server, DTCP certificates are exchanged in the supplemental data handshake TLS handshake message as specified in RFC4680.

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 http://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 03, 2013.

Copyright Notice

Table of Contents

Transport Layer Security
Internet-Draft
Intended status: Informational
Expires: July 03, 2013

D. Thakore
CableLabs

January 2013

Transport Layer Security (TLS) Authorization Using DTCP Certificate
draft-dthakore-tls-authz-02

Abstract

This document specifies the use of DTCP certificate as an authorization extension in the Transport Layer Security Handshake Protocol, according to guidelines in RFC 5878. Extensions carried in the client and server Hello messages confirm that both parties support the desired authorization data types. Then if supported by both the client and server, DTCP certificates are exchanged in the supplemental data handshake TLS handshake message as specified in RFC4680.

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 http://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 03, 2013.

1. Introduction

The Transport Layer Security (TLS) protocol (TLS1.0 [RFC2246], TLS1.1 [RFC4346], TLS1.2 [RFC5246]) is being used in an increasing variety of operational environments, the most common among which is its use in securing HTTP traffic ([RFC2818]). RFC 5878 [AUTHZ] introduces extensions that enable TLS to operate in environments where authorization information needs to be exchanged between the client and the server before any protected data is exchanged. The use of these TLS authorization extensions is especially attractive since it can allow the client and server to determine the type of protected data to exchange based on the authorization information received in the extensions.

A number of consumer electronics devices such as TV's, tablets, game consoles, set-top boxes and other multimedia devices contain Digital Transmission Licensing Administrator [DTLA] issued Digital Transmission Content Protection [DTCP] certificates. These certificates are used for link protection over various types of links like DTCP over IP [DTCP-IP] to securely transmit premium audio-visual content between devices. These DTCP certificates can also be used to verify device functionality, besides being used to protect audio-visual content.

This document describes the format and necessary identifiers to exchange DTCP certificates within the supplemental data message (see [SuppData]) while negotiating a TLS exchange. The DTCP certificates are then used for authorization independent of its use for content protection and the corresponding DTCP Authentication and Key Exchange

(AKE) protocol. This authorization allows a client and/or server to perform certain actions or provide specific services. The actual semantics of the authorization decision by the client/server are beyond the scope of this document. The DTCP certificate can be cryptographically tied to the X.509 certificate being used during the TLS tunnel establishment by an EC-DSA [DTCP] signature.

1.1. Use Case for using DTCP Certificates

As mentioned above, the primary use of DTCP Certificates in consumer electronics devices (eg. TV's) is for secure transmission of audio-visual content. The Authentication and Key Exchange (AKE) protocol defined in [DTCP] is used to exchange DTCP Certificates and allows a device to be identified and authenticated based on the information in the DTCP Certificate. However these devices also have additional functionality (eg. HTML5 User Agents, web enabled apps) that is distinct from its capability to transmit and play audio-visual content. The mechanism outlined in this document allows a device to authenticate and authorize itself using its DTCP Certificate when accessing services over the internet (eg. web app on a TV accessing some service over HTTPS). This allows reusing an already provisioned credential for different services that are delivered over TLS.

1.2. Conventions

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 RFC 2119 [RFC2119].

2. Overview

2.1. Overview of DTCP Certificates

DTCP certificates are issued by [DTLA] for compliant devices and come in three general variations (see Section 4.2.3.1 of the DTCP Specification [DTCP])

Restricted Authentication device certificate format (Format 0): Typically issued to devices with limited computation resources.

Baseline Full Authentication device certificate format (Format 1): This is is the most commonly issued certificate format. Format 1 certificates include a unique Device ID and device EC-DSA public/private key pair generated by the DTLA. (See Section 4.3 of DTCP [DTCP])

Extended Full Authentication device certificate format (Format 2): This is is issued to devices that possess additional functions (eg. additional channel ciphers, specific device properties). The presence of these additional functions is indicated by the device capability mask as specified in Section 4.2.3.2 of the DTCP specification [DTCP]. Format 2 certificates also include a unique Device ID and device EC-DSA public/private key pair

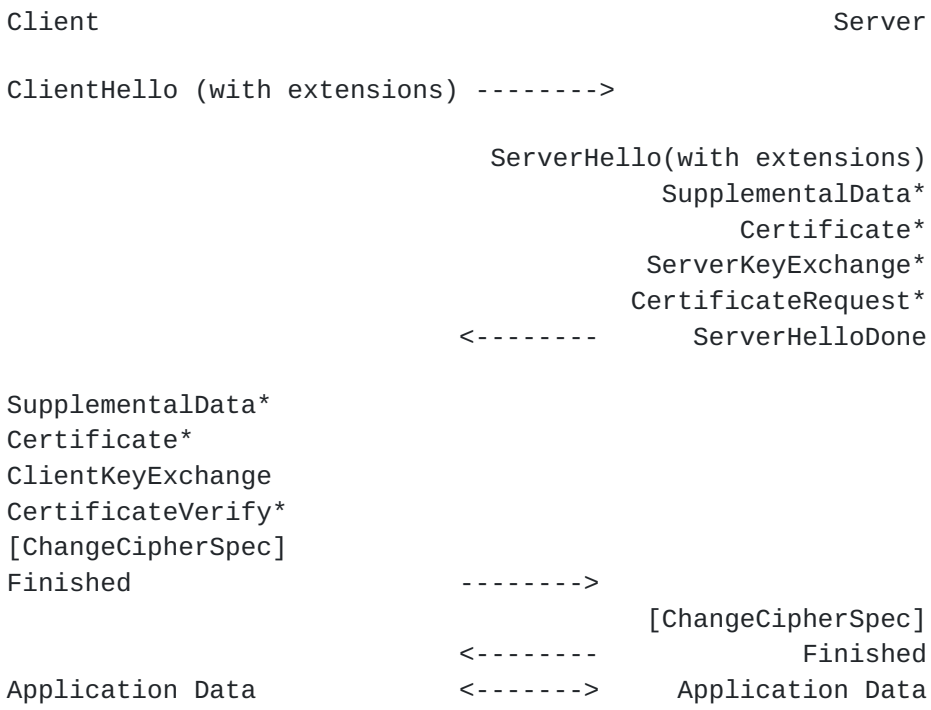
generated by the DTLA. (See Section 4.3 of DTCP [DTCP])

The mechanism specified in this document allows only Format 1 and Format 2 type DTCP certificates to be exchanged in the supplemental data message since it requires the use of the EC-DSA private key associated with the certificate.

2.2. Overview of Supplemental Data handshake

Figure 1 (Figure 1) illustrates the exchange of SupplementalData message during the TLS handshake as specified in RFC4680 [SuppData] and is repeated here for convenience.

TLS handshake message exchange with SupplementalData [SuppData]



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

[] Indicates that ChangeCipherSpec is an independent TLS protocol content type; it is not a TLS handshake message.

Figure 1

2.3. Overview of authorization extensions

RFC5878 [AUTHZ] defines two authorization extension types that are

used in the ClientHello and ServerHello messages and are repeated below for convenience.

```
enum {
  client_authz(7), server_authz(8), (65535)
} ExtensionType;
```

A client uses the client_authz and server_authz extensions in the ClientHello message to indicate that it will send client authorization data and receive server authorization data respectively in the SupplementalData messages. A server uses the extensions in a similar manner in its ServerHello message. RFC5878 [AUTHZ] also establishes a registry that is maintained by IANA for registering authorization data formats. This document defines a new authorization data type that is used in both the client_authz and server_authz extensions and allows the client and server to exchange DTCP certificates in the SupplementalData message.

2.4. Overview of supplemental data usage for authorization

Section 3 of RFC5878 [AUTHZ] specifies the syntax of the supplemental data message when carrying the authz_data message that is negotiated in the client_authz and/or server_authz types. The syntax is repeated here for convenience.

```
enum {
   authz_data(16386), (65535)
} SupplementalDataType;

struct {
   SupplementalDataType supplemental_data_type;
   select(SupplementalDataType) {
      case authz_data:  AuthorizationData;
   }
} SupplementalData;
```

This document defines a new authorization data format that is used in the authz_data message when sending DTCP Authorization data.

3. DTCP Authorization Data Format

3.1. DTCP Authorization Type

The DTCP Authorization type definition in the TLS Authorization Data Formats registry is:

```
dtcp_authorization(TBA);
```

3.2. DTCP Authorization Data

The DTCP Authorization data SHALL be sent in the authz_data message when the authorization data type is dtcp_authorization. The syntax of the authorization data is:

```
struct {
    opaque random_bytes[32];
} RandomNonce;

struct {
    opaque DTCPCert<1..2^24-1>;
    [[opaque ASN.1Cert<1..2^24-1>]];
    opaque signature<1..2^16-1>;
} DigitallySigned;

struct {
    RandomNonce nonce;
    [[DigitallySigned certs]];
} dtcp_authz_data;
```

RandomNonce - consists of 32 bytes generated by a secure random number generator. The dtcp_authz_data message MUST always contain a RandomNonce to avoid replay attacks.

If the ASN.1 Certificate is being sent in the structure above, it MUST be the same as the sender's certificate that will be sent in the Certificate or ClientCertificate message.

DigitallySigned - contains the DTCP Certificate and the optional ASN.1 Certificate. This is then followed by a digital signature covering the DTCP Certificate and the ASN.1 certificate if it is present. The signature is generated using the private key associated with the DTCP certificate using an Elliptic Curve Digital Signature Algorithm (EC-DSA) as specified in [DTCP]. This signature provides proof of the possession of the private key by the sender. A sender sending its own DTCP Certificate MUST populate the certs field.

3.3. Usage rules for clients to exchange DTCP Authorization data

A client MUST include both the client_authz and server_authz extensions in the extended client hello message when indicating its

desire to exchange DTCP authorization data with the server. Additionally the client MUST use the authorization data type specified in Section 3.1 in the extension_data field to specify the format of the authorization data. A client will receive the server's dtcp_authz_data before it sends its own dtcp_authz_data. When sending its own dtcp_authz_data message, the client MUST use the same RandomNonce that it received in the server's dtcp_authz_data message. A client MAY include its ASN.1 Certificate in the certs field to cryptographically tie its dtcp_authz_data with the TLS session being established.

3.4. Usage rules for servers to exchange DTCP Authorization data

A server MUST respond with both the client_authz and server_authz extensions in the extended server hello message when indicating its desire to exchange dtcp_authorization data with the client. Additionally the server MUST use the authorization data type specified in Section 3.1 in the extension_data field to specify the format of the dtcp_authorization data. A server MUST generate and populate the RandomNonce in the dtcp_authz_data message. If the client's hello message does not contain both the client_authz and server_authz extensions with dtcp_authorization type, the server SHALL NOT include support for dtcp_authorization data in its hello message. A server MAY include its ASN.1 Certificate in the certs field to cryptographically tie its dtcp_authz_data with the TLS session being established.

3.5. Alert Messages

This document reuses TLS Alert messages for any errors that arise during authorization processing, while preserving the AlertLevels as specified in [AUTHZ]. Additionally the following AlertDescription values SHALL be used to report errors in dtcp_authorization processing:

unsupported_extension:
In dtcp_authorization processing a client uses this when it receives a server hello message that indicates support for only one of client_authz or server_authz extension. This message is always fatal.

4. Acknowledgements

This document derives its structure and much of its content from [SuppData], [AUTHZ] and [RFC6042].

5. IANA Considerations

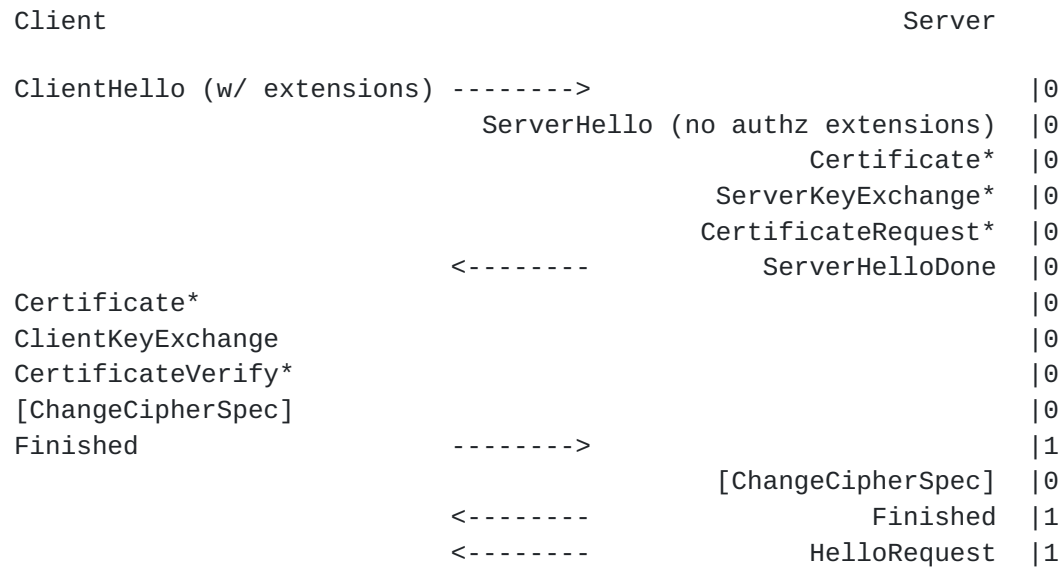
This document requires a new entry in the IANA-maintained TLS Authorization Data Formats registry, dtcp_authorization(TBA). This registry is defined in [AUTHZ].

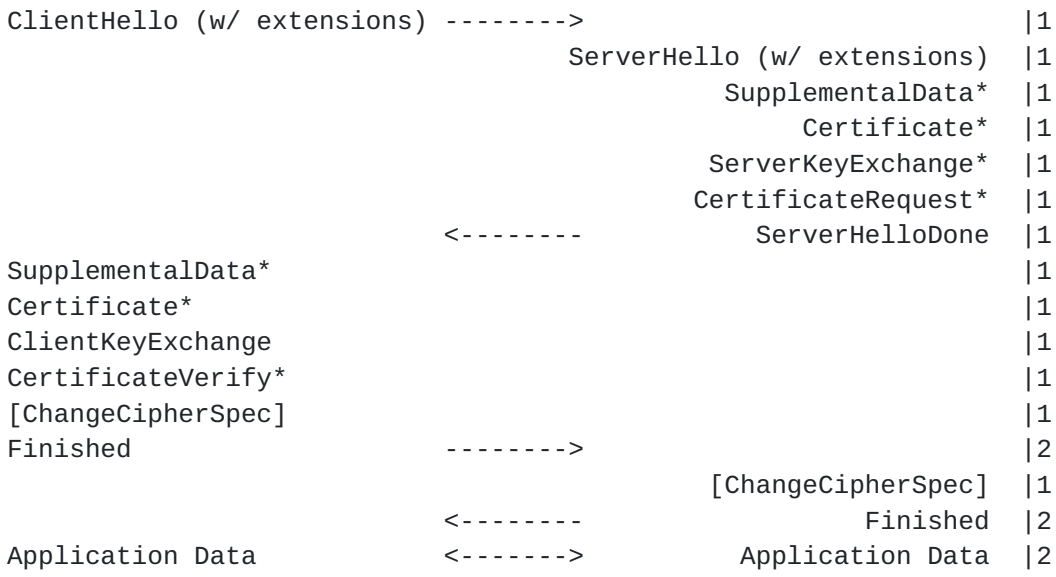
6. Security Considerations

In cases where the SupplementalData information is sensitive, the double handshake technique described in [SuppData] can be used to provide protection for the SupplementalData information. The double handshake specified in [SuppData] assumes that the client knows the context of the TLS session that is being set up and uses the authorization extensions as needed. Figure 2 illustrates a variation of the double handshake that addresses the case where the client may not have a priori knowledge that it will be communicating with a server capable of exchanging dtcp_authz_data (typical for https usages based on [RFC2818]). In Figure 2 (Figure 2) below client's Hello messages always include support for the client_authz and server_authz extensions. The server simply establishes an encrypted TLS tunnel with the client in the first handshake by not indicating support for any authz extensions. The server initiates a second handshake by sending a HelloRequest. The second handshake will include server's support for authz extensions which will result in SupplementalData being exchanged.

The double handshake mechanism is vulnerable to the TLS MITM Renegotiation exploit as explained in [RFC5746]. In order to address this vulnerability, clients and servers MUST use the secure_renegotiation extension as specified in [RFC5746] when performing a double handshake.

Double Handshake to protect Supplemental Data





* Indicates optional or situation-dependent messages.

Figure 2

There are no additional security considerations beyond those discussed in [DTCP], [DTCP-IP] and [AUTHZ].

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2246] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0 ", RFC 2246, January 1999, <http://tools.ietf.org/html/rfc2246>.

[RFC4346] Dierks, T. and E. Rescorla, "The TLS Protocol Version 1.1 ", RFC 4346, April 2006, <http://tools.ietf.org/html/rfc4346>.

[RFC5246] Dierks, T. and E. Rescorla, "The TLS Protocol Version 1.2 ", RFC 5246, August 2008, <http://tools.ietf.org/html/rfc5246>.

[RFC5746] Rescorla, E., Ray, M., Dispensa, S., and N. Oskov, "Transport Layer Security (TLS) Renegotiation Indication Extension ", RFC 5746, February 2010, <http://tools.ietf.org/html/rfc5746>.

[SuppData] Santesson, S., "TLS Handshake Message for Supplemental

Data", September 2006, <http://tools.ietf.org/html/rfc4680>.

[AUTHZ] Brown, M. and R. Housley, "Transport Layer Security (TLS) Authorization Extensions ", RFC 5878, May 2010, <http://tools.ietf.org/html/rfc5878>.

[DTCP] Digital Transmission Licensing Administrator, "Digital Transmission Content Protection", , <http://www.dtcp.com/documents/dtcp/info-20111214-dtcp-v1-rev-1-p-7.pdf>.

[DTCP-IP] Digital Transmission Licensing Administrator, "DTCP Volume 1 Supplement E", , <http://www.dtcp.com/documents/dtcp/info-20111214-dtcp-v1se-ip-rev-1-p-4-ed-1.pdf>.

7.2. Informative References

[RFC2629] Rose, M.T., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.

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

[DTLA] Digital Transmission Licensing Administrator, "DTLA", , <http://www.dtcp.com>.

[RFC2818] Rescorla, E., "HTTP over TLS", RFC 2818, May 2000, <http://tools.ietf.org/html/rfc2818>.

[RFC6042] Keromytis, A., "Transport Layer Security (TLS) Authorization Using KeyNote ", RFC 6042, October 2010, <http://tools.ietf.org/html/rfc6042>.

Appendix A. Additional Stuff

This becomes an Appendix.

Author's Address

D. Thakore
Cable Television Laboratories, Inc.
858 Coal Creek Circle
Louisville, CO 80023
USA

Email: d.thakore@cablelabs.com