Network Working Group
Request for Comments: 4680
Updates: 4346
Category: Standards Track

S. Santesson
Microsoft

September 2006

TLS Handshake Message for Supplemental Data

Status of This Memo

Copyright Notice

Abstract

This specification defines a TLS handshake message for exchange of supplemental application data. TLS hello message extensions are used to determine which supplemental data types are supported by both the TLS client and the TLS server. Then, the supplemental data handshake message is used to exchange the data. Other documents will define the syntax of these extensions and the syntax of the associated supplemental data types.

1. Introduction

Recent standards activities have proposed different mechanisms for transmitting supplemental application data in the TLS handshake message. For example, recent proposals transfer data that is not processed by the TLS protocol itself, but assist the TLS-protected application in the authentication and authorization decisions. One proposal transfers user name hints for locating credentials, and another proposal transfers attribute certificates and Security Assertions Markup Language (SAML) assertions for authorization checks.

In order to avoid definition of multiple handshake messages, one for each new type of application-specific supplemental data, this specification defines a new handshake message type that bundles together all data objects that are to be delivered to the TLS-protected application and sends them in a single handshake message.

1.1. 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 RFC 2119 [N1].

The syntax for the supplemental_data handshake message is defined using the TLS Presentation Language, which is specified in Section 4 of [N2].

2. Supplemental Data Handshake Message

The new supplemental_data handshake message type is defined to accommodate communication of supplemental data objects as agreed during the exchange of extensions in the client and server hello messages. See RFC 2246 (TLS 1.0) [N2] and RFC 4346 (TLS 1.1) [N3] for other handshake message types.

Information provided in a supplemental data object MUST be intended to be used exclusively by applications and protocols above the TLS protocol layer. Any such data MUST NOT need to be processed by the TLS protocol.

```
   enum {
          supplemental_data(23), (255)
      } HandshakeType;

   struct {
          HandshakeType msg_type;    /* handshake type */
          uint24 length;             /* octets in message */
          select (HandshakeType) {
                 case supplemental_data:   SupplementalData;
            } body;
       } Handshake;

   struct {
         SupplementalDataEntry supp_data<1..2^24-1>;
      } SupplementalData;

   struct {
         SupplementalDataType supp_data_type;
         uint16 supp_data_length;
         select(SupplementalDataType) { }
      } SupplementalDataEntry;

   enum {
         (65535)
     } SupplementalDataType;
```

supp_data_length
   This field is the length (in bytes) of the data selected by
   SupplementalDataType.

The client MUST NOT send more than one SupplementalData handshake message, and the server MUST NOT send more than one SupplementalData handshake message. Receiving more than one SupplementalData handshake message results in a fatal error, and the receiver MUST close the connection with a fatal unexpected_message alert.

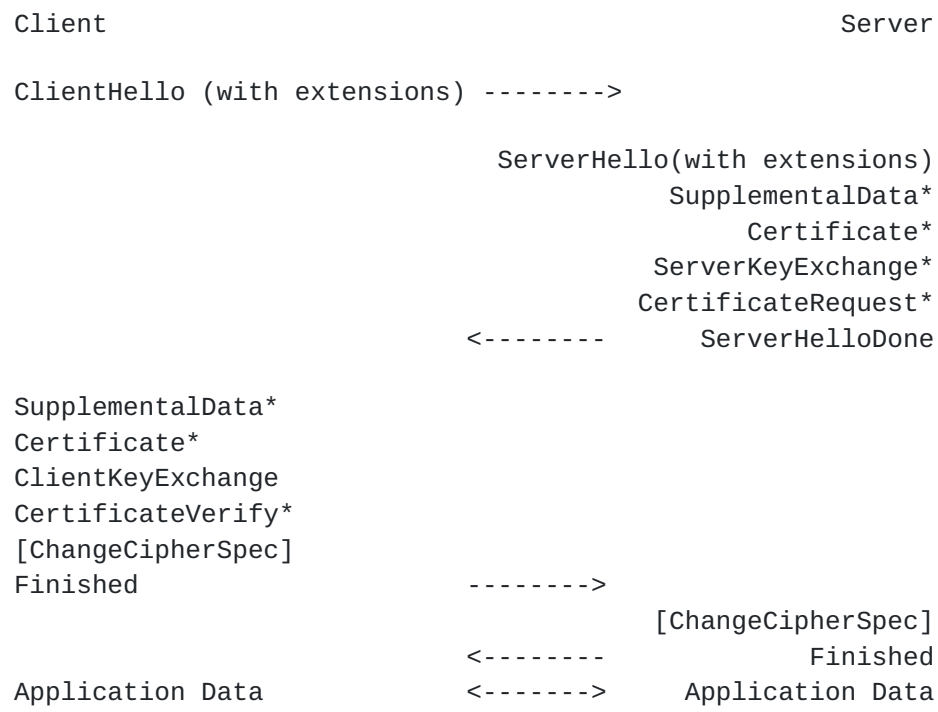
If present, the SupplementalData handshake message MUST contain a non-empty SupplementalDataEntry structure carrying data associated with at least one defined SupplementalDataType. An explicit agreement that governs presence of any supplemental data MUST be concluded between client and server for each SupplementalDataType using the TLS extensions [N4] in the client and server hello messages. Receiving an unexpected SupplementalData handshake message results in a fatal error, and the receiver MUST close the connection with a fatal unexpected_message alert.

Other documents will define specific SupplementalDataTypes and their associated data syntax and processing. These same specifications must also specify the client and server hello message extensions that are used to negotiate the support for the specified supplemental data type. This document simply specifies the TLS Handshake Protocol message that will carry the supplemental data objects.

Different situations require the transfer of supplemental data from the client to the server, require the transfer of supplemental data from the server to the client, or both ways. All three situations are fully supported.

3. Message Flow

The SupplementalData handshake message, if exchanged, MUST be sent as the first handshake message as illustrated in Figure 1 below.



* Indicates optional or situation-dependent messages.

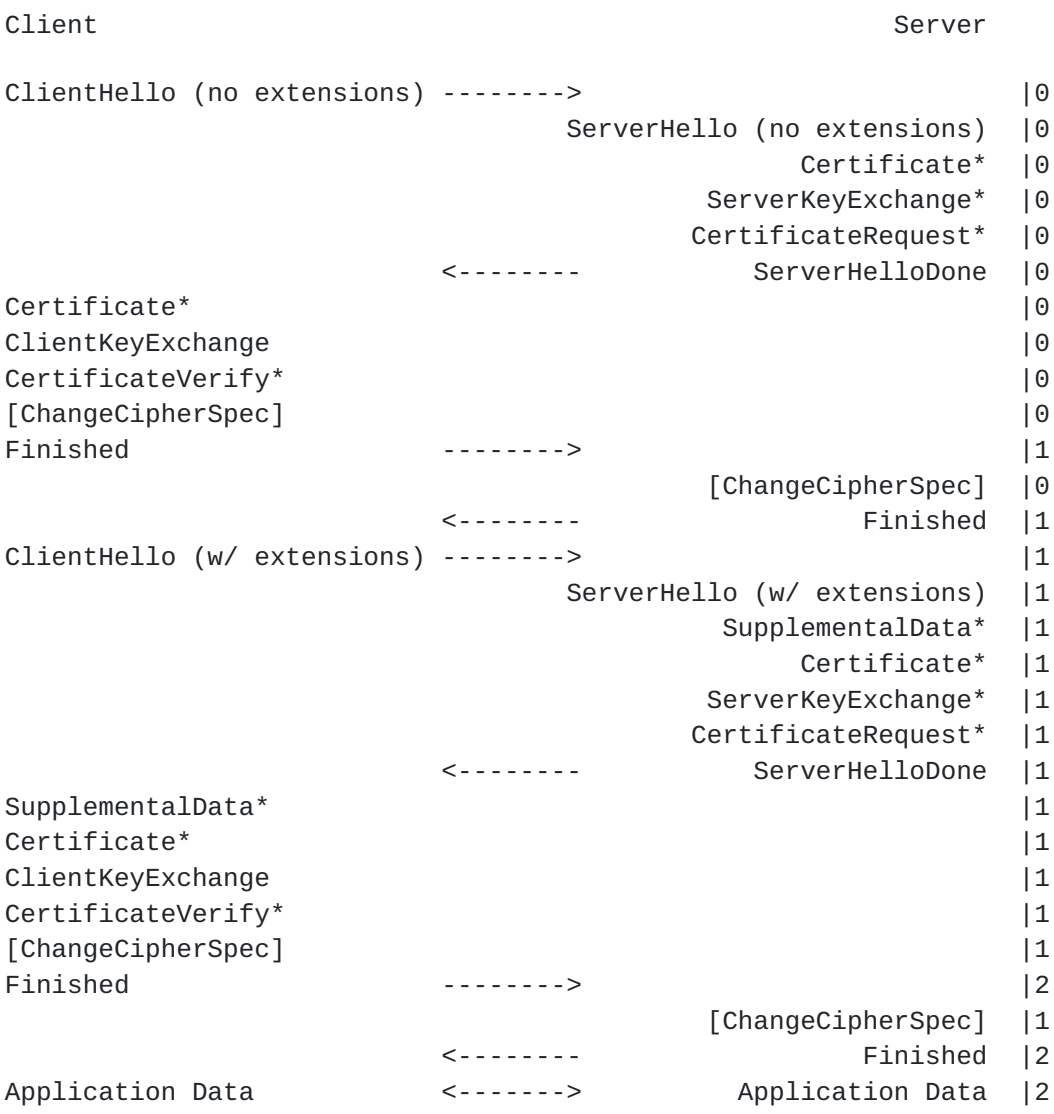
Figure 1. Message Flow with SupplementalData

4. Security Considerations

Each SupplementalDataType included in the handshake message defined in this specification introduces its own unique set of security properties and related considerations. Security considerations must therefore be defined in each document that defines a supplemental data type.

In some cases, the SupplementalData information may be sensitive. The double handshake technique can be used to provide protection for the SupplementalData information. Figure 2 illustrates the double handshake, where the initial handshake does not include any extensions, but it does result in protected communications. Then, a second handshake that includes the SupplementalData information is performed using the protected communications. In Figure 2, the number on the right side indicates the amount of protection for the TLS message on that line. A zero (0) indicates that there is no communication protection; a one (1) indicates that protection is provided by the first TLS session; and a two (2) indicates that protection is provided by both TLS sessions.

The placement of the SupplementalData message in the TLS Handshake results in the server providing its SupplementalData information before the client is authenticated. In many situations, servers will not want to provide authorization information until the client is authenticated. The double handshake illustrated in Figure 2 provides a technique to ensure that the parties are mutually authenticated before either party provides SupplementalData information.



* Indicates optional or situation-dependent messages.

Figure 2. Double Handshake to Protect Supplemental Data

5. IANA Considerations

IANA has taken the following actions:

1) Created an entry, supplemental_data(23), in the existing registry for HandshakeType (defined in RFC 2246 [N2]).

2) Established a registry for TLS Supplemental Data Formats (SupplementalDataType). Values in the inclusive range 0-16385 (decimal) are assigned via RFC 2434 [N5] Standards Action. Values from the inclusive range 16386-65279 (decimal) are assigned via RFC 2434 IETF Consensus. Values from the inclusive range 65280-65535 (decimal) are reserved for RFC 2434 Private Use.

6. Normative References

[N1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[N2] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[N3] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006.

[N4] Blake-Wilson, S., Nystrom, M., Hopwood, D., Mikkelsen, J., and T. Wright, "Transport Layer Security (TLS) Extensions", RFC 4366, April 2006.

[N5] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

7. Acknowledgements

The fundamental architectural idea for the supplemental data handshake message was provided by Russ Housley and Eric Rescorla.

Author's Address

Stefan Santesson
Microsoft
Finlandsgatan 30
164 93 KISTA
Sweden

EMail: stefans@microsoft.com

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).