Network Working Group	A. Langley
Internet-Draft	Google Inc
Expires: April 26, 2012	October 24, 2011

Transport Layer Security (TLS) Encrypted Client Certificates
draft-agl-tls-encryptedclientcerts-00

Abstract

This document describes a Transport Layer Security (TLS) extension that allows client certificates to be encrypted in the initial TLS handshake.

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 April 26, 2012.

Copyright Notice

Table of Contents

1. Introduction

TLS *[RFC5246]* defines a handshake in which both the server's and client's certificates (if any) are sent in the clear during the initial handshake. Although the server's certificates are usually non-sensitive, client certificates may include email address or even full legal names. Even client certificates that contain nothing but a serial number provide a unique identifier that can be correlated across connections by an eavesdropper.

This motivates encrypting the client's certificates. One existing solution is to perform an initial handshake without client authentication and then to renegotiate with it. This solves the disclosure issue but at a significant cost in handshake overhead and latency. The solution presented below simply moves the client's certificates after the client's ChangeCipherSpec. This is fundamentally incompatable with DH or ECDH certificates but we note that such certificates are rarely used in our experience. This solution is also weak as it only defends against eavesdroppers, not active attackers. We still consider it worthwhile given the very low cost.

2. Requirements Notation

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]*.

3. Encrypted client certificates extension

A new extension type (encrypted_client_certificates(provisionally 13180)) is defined and MAY be included by the client in its ClientHello message. If, and only if, the server sees this extension in the ClientHello, it MAY choose to include the extension in its ServerHello. The extension_data MUST be empty in each case.

```
enum {
  encrypted_client_certificates(provisionally 13180), (65535)
} ExtensionType;
```

If the extension is echoed by the server then encrypted client certificates are in effect for the handshake. This causes the client's second flow to be reordered so that the Certificate and CertificateVerify messages occur after the ChangeCipherSpec.

Here is an example of the client's second flow without encrypted client certificates (taken from RFC 5246 *[RFC5246]*):

```
Certificate*
ClientKeyExchange
CertificateVerify*
[ChangeCipherSpec]
Finished
```

When client encrypted certificates are in effect, this becomes:

```
ClientKeyExchange
[ChangeCipherSpec]
Certificate*
CertificateVerify*
Finished
```

The handshake_messages value of the CertificateVerify is constructed using the new message order.

This extension does not imply that a CertificateRequest handshake message will be sent by the server, nor that a Certificate or CertificateVerify message will be sent by the client. It only affects the message ordering when a client certificate would have normally been sent in the clear.

4. Security considerations

In the course of a normal handshake, the use of this extension will protect the client certificate from eavesdroppers. An active attacker can perform a downgrade attack and expose the client's certificates at the cost of dooming the connection. In order to defend against the active attack, a strict client may refuse to send certificates is the server doesn't support this extension in the initial handshake.

5. IANA Considerations

This document requires IANA to update its registry of TLS extensions to assign an entry, referred herein as encrypted_client_certificates.

6. Acknowledgements

Thanks to Wan-Teh Chang, Diana Smetters, Brian Smith, Adam Barth, Dirk Balfanz and Mayank Upadhyay for discussions around this design.

7. References

[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
[RFC5246]	Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

Appendix A. Changes

To be removed by RFC Editor before publication

Author's Address

Adam Langley Langley Google Inc EMail: agl@google.com