Internet Engineering Task Force (IETF)
Request for Comments: 6091
Obsoletes: 5081
Category: Informational
ISSN: 2070-1721

N. Mavrogiannopoulos
KUL
D. Gillmor
Independent

February 2011

# Using OpenPGP Keys for Transport Layer Security (TLS) Authentication

## Abstract

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

## Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6091.

Table of Contents



## 1. Introduction

The IETF has two sets of standards for public key certificates: one set for the use of X.509 certificates [RFC5280], and one for OpenPGP certificates [RFC4880]. At the time of this writing, TLS [RFC5246] standards are defined to use X.509 certificates. This document specifies a way to negotiate the use of OpenPGP certificates for a TLS session, and specifies how to transport OpenPGP certificates via TLS. The proposed extensions are backward-compatible with the current TLS specification, so that existing client and server implementations that make use of X.509 certificates are not affected.

These extensions are not backward-compatible with [RFC5081], and the major differences are summarized in Appendix A. Although the OpenPGP CertificateType value is being reused by this memo with the same number as that specified in [RFC5081] but with different semantics, we believe that this causes no interoperability issues because the latter was not widely deployed.

## 2. Terminology

The term "OpenPGP key" is used in this document as in the OpenPGP specification [RFC4880]. We use the term "OpenPGP certificate" to refer to OpenPGP keys that are enabled for authentication.

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

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 handshake message contents when OpenPGP certificates are to be used for authentication.

### 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), (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 OpenPGP certificates. All existing cipher suites that support a key exchange method compatible with the key in the certificate can be used in combination with OpenPGP 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 Certificate

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.

If the OpenPGP certificate type is negotiated, then it is required to present an OpenPGP certificate in the certificate message. The certificate must contain a public key that matches the selected key exchange algorithm, as shown below.

| Key Exchange Algorithm | OpenPGP Certificate Type |
|---|---|
| RSA | RSA public key that can be used for encryption. |
| DHE_DSS | DSA public key that can be used for authentication. |
| DHE_RSA | RSA public key that can be used for authentication. |

An OpenPGP certificate appearing in the certificate message is sent using the binary OpenPGP format. The certificate MUST contain all the elements required by Section 11.1 of [RFC4880].

OpenPGP certificates to be transferred are placed in the Certificate structure and tagged with the OpenPGPCertDescriptorType "subkey_cert". Since those certificates might contain several subkeys, the subkey ID to be used for this session is explicitly

specified in the OpenPGPKeyID field. The key ID must be specified even if the certificate has only a primary key. The peer, upon receiving this type, has to either use the specified subkey or terminate the session with a fatal alert of "unsupported_certificate".

The option is also available to send an OpenPGP fingerprint, instead of sending the entire certificate, by using the "subkey_cert_fingerprint" tag. This tag uses the OpenPGPSubKeyFingerprint structure and requires the primary key fingerprint to be specified, as well as the subkey ID to be used for this session. The peer shall respond with a "certificate_unobtainable" fatal alert if the certificate with the given fingerprint cannot be found. The "certificate_unobtainable" fatal alert is defined in Section 5 of [RFC6066].

Implementations of this protocol MUST ensure that the sizes of key IDs and fingerprints in the OpenPGPSubKeyCert and OpenPGPSubKeyFingerprint structures comply with [RFC4880]. Moreover, it is RECOMMENDED that the keys to be used with this protocol have the authentication flag (0x20) set.

The process of fingerprint generation is described in Section 12.2 of [RFC4880].

The enumerated types "cert_fingerprint" and "cert" of OpenPGPCertDescriptorType that were defined in [RFC5081] are not used and are marked as obsolete by this document. The "empty_cert" type has replaced "cert" and is a backward-compatible way to specify an empty certificate; "cert_fingerprint" MUST NOT be used with this updated specification, and hence that old alternative has been removed from the Certificate struct description.

```
enum {
     empty_cert(1),
     subkey_cert(2),
     subkey_cert_fingerprint(3),
     (255)
} OpenPGPCertDescriptorType;

uint24 OpenPGPEmptyCert = 0;

struct {
    opaque OpenPGPKeyID<8..255>;
    opaque OpenPGPCert<0..2^24-1>;
} OpenPGPSubKeyCert;

struct {
    opaque OpenPGPKeyID<8..255>;
    opaque OpenPGPCertFingerprint<20..255>;
} OpenPGPSubKeyFingerprint;

struct {
     OpenPGPCertDescriptorType descriptorType;
     select (descriptorType) {
          case empty_cert: OpenPGPEmptyCert;
          case subkey_cert: OpenPGPSubKeyCert;
          case subkey_cert_fingerprint:
              OpenPGPSubKeyCertFingerprint;
     }
} Certificate;
```

### 3.4. Certificate Request

The semantics of this message remain the same as in the TLS specification. However, if this message is sent, and the negotiated certificate type is OpenPGP, the "certificate_authorities" list MUST be empty.

### 3.5. Client Certificate

This message is only sent in response to the certificate request message. The client certificate message is sent using the same formatting as the server certificate message, and it is also required to present a certificate that matches the negotiated certificate type. If OpenPGP certificates have been selected and no certificate is available from the client, then a certificate structure of type "empty_cert" that contains an OpenPGPEmptyCert value MUST be sent. The server SHOULD respond with a "handshake_failure" fatal alert if client authentication is required.

### 3.6. Other Handshake Messages

All the other handshake messages are identical to the TLS specification.

## 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.

When using OpenPGP fingerprints instead of the full certificates, the discussion in Section 5 of [RFC6066] for "Client Certificate URLs" applies, especially when external servers are used to retrieve keys. However, a major difference is that although the "client_certificate_url" extension allows identifying certificates without including the certificate hashes, this is not possible in the protocol proposed here. In this protocol, the certificates, when not sent, are always identified by their fingerprint, which serves as a cryptographic hash of the certificate (see Section 12.2 of [RFC4880]).

The information that is available to participating parties and eavesdroppers (when confidentiality is not available through a previous handshake) is the number and the types of certificates they hold, plus the contents of the certificates.

## 5. IANA Considerations

This document uses a registry and the "cert_type" extension originally defined in [RFC5081]. Existing IANA references have been updated to point to this document.

In addition, the "TLS Certificate Types" registry established by [RFC5081] has been updated in the following ways:

1. Values 0 (X.509) and 1 (OpenPGP) are defined in this document.

2. Values from 2 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. Acknowledgements

The authors wish to thank Alfred Hoenes and Ted Hardie for their suggestions on improving this document.

## 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.

[RFC4880] Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and R. Thayer, "OpenPGP Message Format", RFC 4880, November 2007.

[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.

[RFC6066] Eastlake 3rd, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, January 2011.

### 7.2. Informative References

[RFC5081] Mavrogiannopoulos, N., "Using OpenPGP Keys for Transport Layer Security (TLS) Authentication", RFC 5081, November 2007.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

## Appendix A. Changes from RFC 5081

This document incorporates a major change in the "Server Certificate" and "Client Certificate" TLS messages that will make implementations following this protocol incompatible with those following [RFC5081]. This change requires the subkey IDs used for TLS authentication to be marked explicitly in the handshake procedure. This was decided in order to place no limitation on the OpenPGP certificates' contents that can be used with this protocol.

[RFC5081] required that an OpenPGP key or subkey be marked with the authentication flag; thus, authentication would have failed if this flag was not set or if this flag was set in more than one subkey. The protocol in this memo has no such limitation.

## Authors' Addresses

Nikos Mavrogiannopoulos
ESAT/COSIC Katholieke Universiteit Leuven
Kasteelpark Arenberg 10, bus 2446
Leuven-Heverlee, B-3001
Belgium

EMail: nikos.mavrogiannopoulos@esat.kuleuven.be

Daniel Kahn Gillmor
Independent
119 Herkimer St.
Brooklyn, NY 11216-2801
US

EMail: dkg@fifthhorseman.net