Abstract

This document introduce the public-key signature algorithm EdDSA for use in Transport Layer Security (TLS). By defining new SignatureAlgorithm, NamedCurve and ECPointFormat enumerations, we describe how EdDSA and Ed25519 is used for digital signatures in the existing ECDSA cipher suites. This is intended to work with any version of TLS and Datagram TLS.

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 December 11, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

Josefsson Expires December 11, 2015 [Page 1]
the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

1. Introduction

TLS [RFC5246] and DTLS [RFC6347] support different key exchange algorithms and authentication mechanisms, and define the SignatureAlgorithm enumeration for different signature algorithms. In TLS-ECC [RFC4492], key exchange and authentication using ECC is specified, where the NamedCurve and ECPointFormat registries and associated TLS extensions are introduced.

This document describes how to use EdDSA and Ed25519 [I-D.josefsson-eddsa-ed25519] as a new authentication mechanism in TLS. It define new SignatureAlgorithm, NamedCurve and ECPointFormat enumeration values and describe how these are used to negotiate EdDSA-based signatures.

The goal is that all existing ECDSA cipher suites will, when the EdDSA SignatureAlgorithm is negotiated, use EdDSA instead of the traditional ECDSA signature algorithm.

This document is a self-contained alternative to draft-josefsson-tls-eddsa. This document reuse the ECDSA cipher suites for EdDSA, whereas draft-josefsson-tls-eddsa specify new cipher suites for EdDSA. It is an open issue which approach is to be prefered.

1.1. Requirements 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 [RFC2119].

2. EdDSA SignatureAlgorithm

Negotiation of the authentication mechanism is signalled by sending a SignatureAlgorithm value. Here we extend this enumeration for EdDSA.

```c
enum {
    eddsa(4)
} SignatureAlgorithm;
```

EdDSA is suitable for use with TLS [RFC5246] and DTLS [RFC6347].

Note that EdDSA merely signals which signature algorithm to use, it does not imply any curve or hash parameter choice.
3. EdDSA Public-key format

This section defines a new point format suitable to encode EdDSA public keys, as well as an identifier to negotiate this new format in TLS, and includes guidance on their use.

The curves defined in [RFC4492] define a public key as a point on the curve. In order to exchange public keys, the points are serialized as a string of bytes. Since EdDSA public keys already are string of bytes, no serialization is needed.

We define the following ECPointFormat value for raw binary EdDSA public keys.

```c
enum {
    eddsa_public_key(4),
} ECPointFormat;
```

When included in a ServerKeyExchange or ClientKeyExchange message, the public key is wrapped in an ECPoint structure as defined in [RFC4492], whose payload is as described above.

For example, a Ed25519 public key with value D75A980182B10AB7D54BFED3C964073A0EE172F3DAA62325AF021A68F707511A appears on the wire as follows (including the length byte of ECPoint.point).

```
  20 D7 5A 98 01 82 B1 0A B7 D5 4B FE D3 C9 64 07
  3A 0E E1 72 F3 DA A6 23 25 AF 02 1A 68 F7 07 51
  1A
```

4. Ed25519 NamedCurve

Negotiation the elliptic curve is signalled with the Supported Elliptic Curves Extension extension. Here we extend the NamedCurve enumeration for EdDSA with the Ed25519 parameter choice.

```c
enum {
    ed25519(5)
} NamedCurve;
```

5. Using EdDSA in a handshake

The following describe how EdDSA is used in a handshake. For ease of explanation, we assume a full handshake. [RFC4492] describes the semantics of ECC in TLS, including how resumed handshakes work, and this document refer to it for a complete discussion.
A client that wishes to negotiate use of EdDSA in a handshake MUST offer an ECDSA key exchange algorithm and send, in the same way as is done for other Signature Algorithms in ECDSA, a Signature Algorithm extension that include the "eddsa" SignatureAlgorithm value. The HashAlgorithm, NamedCurve and ECPointFormat types are also required, but depend on the actual EdDSA parameter choices.

The HashAlgorithm value to specify for Ed25519 MUST be "sha512", to indicate use of Ed25519-SHA-512 as defined in [I-D.josefsson-eddsa-ed25519]. The meaning of using a "eddsa" SignatureAlgorithm with other HashAlgorithms will require additional specification to specify the meaning.

The Supported Elliptic Curves Extension ("NamedCurve") must also contain the relevant curve. Currently the only curve applicable to EdDSA is the "ed25519" NamedCurve value defined in this document. This is used to indicate request for Ed25519.

The client MUST also include a Supported Point Formats Extension ("ECPointFormat") that contains the "eddsa_public_key" value.

6. IANA Considerations

IANA is requested to assign a number for EdDSA described in Section 2 to the Transport Layer Security (TLS) Parameters [IANA-TLS] registry under "SignatureAlgorithm" as follows.

```
+--------+-------------+---------+-----------+
| Value  | Description | DTLS-OK | Reference |
+--------+-------------+---------+-----------+
|   4    |    eddsa    |    Y    |  This doc |
```

IANA is requested to assign a number for Ed25519 described in Section 4 to the Transport Layer Security (TLS) Parameters registry EC Named Curve [IANA-TLS] as follows.

```
+--------+-------------+---------+-----------+
| Value  | Description | DTLS-OK | Reference |
+--------+-------------+---------+-----------+
|   5    |   ed25519   |    Y    |  This doc |
```

IANA is requested to assign a number for eddsa_public_key described in Section 3 to the Transport Layer Security (TLS) Parameters registry EC Point Format [IANA-TLS] as follows.

```
+--------+-------------+---------+-----------+
| Value  | Description | DTLS-OK | Reference |
+--------+-------------+---------+-----------+
```
### Security Considerations

The security considerations of TLS [RFC5246], DTLS [RFC6347], TLS-ECC [RFC4492], and EdDSA and Ed25519 [I-D.josefsson-eddsa-ed25519] are inherited.

As with all cryptographic algorithms and security protocols, the reader should stay informed about new research insights into the security of the algorithms and protocols involved.

While discussed in the EdDSA/Ed25519 specification and papers, we would like to stress the significance of EdDSA/Ed25519 implementation security. In particular, implementations must avoid side-channel attacks by, for example, being constant-time.

### Acknowledgements

Thanks to Yoav Nir for suggesting re-use of ECDSA cipher suites with EdDSA, to reduce the cartesian product cipher suite explosion. Thanks to Klaus Hartke and Nicolas Williams for numerous fixes to the document. Other people who contributed include Ilari Liusvaara and Martin Thomson.

### References

#### 9.1. Normative References


9.2. Informative References

[IANA-TLS]
Internet Assigned Numbers Authority, "Transport Layer Security (TLS) Parameters",

Author’s Address

Simon Josefsson
SJD AB

Email: simon@josefsson.org