TLS Working Group
Internet Draft
Intended status: Informational
Expires: July 2008

Mohamad Badra
LIMOS Laboratory
February 1, 2008

ECDHE_PSK Ciphersuites for Transport Layer Security (TLS)
draft-badra-ecdhe-tls-psk-03.txt

Status of this Memo

Copyright Notice

Abstract

This document extends RFC 4279 and RFC 4785 and specifies a set of ciphersuites that use an Elliptic Curve Diffie-Hellman exchange authenticated with a pre-shared key. These ciphersuites provide Perfect Forward Secrecy. It also specifies one authentication-only ciphersuites (with no encryption). This ciphersuite is useful when authentication and integrity protection is desired, but confidentiality is not needed or not permitted.

The reader is expected to become familiar with RFC 4279 and RFC 4785 prior to studying this document.

1. Introduction

RFC 4279 specifies ciphersuites for supporting TLS using pre-shared symmetric keys and they (a) use only symmetric key operations for authentication, (b) use a Diffie-Hellman exchange authenticated with a pre-shared key, or (c) combines public key authentication of the server with pre-shared key authentication of the client.

RFC 4785 specifies authentication-only ciphersuites (with no encryption).

This document specifies a set of ciphersuites that use an Elliptic Curve Diffie-Hellman exchange authenticated with a pre-shared key. These ciphersuites provide Perfect Forward Secrecy. This document also specifies one authentication-only ciphersuites (with no encryption). This ciphersuite is useful when authentication and integrity protection is desired, but confidentiality is not needed or not permitted.

1.1. Conventions used in this document

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. ECDHE_PSK Key Exchange Algorithm

The ciphersuites in this section match the ciphersuites defined in [RFC4279], except that they use an Elliptic Curve Diffie-Hellman exchange [RFC4492] authenticated with a pre-shared key. They are defined as follow:

CipherSuite	Key Exchange	Cipher	Hash
TLS_ECDHE_PSK_WITH_RC4_128_SHA	ECDHE_PSK	RC4_128	SHA
TLS_ECDHE_PSK_WITH_3DES_EDE_CBC_SHA	ECDHE_PSK	3DES_EDE_CBC	SHA
TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA	ECDHE_PSK	AES_128_CBC	SHA
TLS_ECDHE_PSK_WITH_AES_256_CBC_SHA	ECDHE_PSK	AES_256_CBC	SHA

When the ciphersuites defined in this document are used, the 'ec_diffie_hellman_psk' case inside the ServerKeyExchange and ClientKeyExchange structure is used instead of the 'psk' case defined in [RFC4279] (i.e. The ServerKeyExchange and ClientKeyExchange messages include the Diffie-Hellman parameters). The PSK identity and

identity hint fields MUST have the same meaning specified in [RFC4279] (note that the ServerKeyExchange message is always sent, even if no PSK identity hint is provided).

The format of the ServerKeyExchange and ClientKeyExchange messages is shown below.

```
struct {
    select (KeyExchangeAlgorithm) {
        /* other cases for rsa, diffie_hellman, etc. */
        case ec_diffie_hellman_psk:  /* NEW */
            opaque psk_identity_hint<0..2^16-1>;
            ServerECDHParams params;
    };
} ServerKeyExchange;

struct {
    select (KeyExchangeAlgorithm) {
        /* other cases for rsa, diffie_hellman, etc. */
        case ec_diffie_hellman_psk:   /* NEW */
            opaque psk_identity<0..2^16-1>;
            ClientECDiffieHellmanPublic public;
    } exchange_keys;
} ClientKeyExchange;
```

The premaster secret is formed as follows. First, perform an ECDH operation (See section 5.10 of [RFC4492]) to compute the shared secret. Next, concatenate a uint16 containing the length of the shared secret (in octets), the shared secret itself, a uint16 containing the length of the PSK (in octets), and the PSK itself.

This corresponds to the general structure for the premaster secrets (see Note 1 in Section 2 of [RFC4279]), with "other_secret" containing the shared secret:

```
struct {
     opaque other_secret<0..2^16-1>;
     opaque psk<0..2^16-1>;
};
```

3. 2. ECDHE_PSK Key Exchange Algorithm with NULL Encryption

The ciphersuite in this section matches the ciphersuites defined in [RFC4785], except that it uses an Elliptic Curve Diffie-Hellman exchange authenticated with a pre-shared key.

CipherSuite	Key Exchange	Cipher	Hash
TLS_ECDHE_PSK_WITH_NULL_SHA	ECDHE_PSK	NULL	SHA

4. Security Considerations

The security considerations described throughout [RFC4346], [RFC4785] and [RFC4279] apply here as well.

5. IANA Considerations

This document defines the following new ciphersuites, whose values are to be assigned from the TLS Cipher Suite registry defined in [RFC4346].

```
CipherSuite TLS_ECDHE_PSK_WITH_RC4_128_SHA        = { 0xXX, 0xXX };
CipherSuite TLS_ECDHE_PSK_WITH_3DES_EDE_CBC_SHA   = { 0xXX, 0xXX };
CipherSuite TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA    = { 0xXX, 0xXX };
CipherSuite TLS_ECDHE_PSK_WITH_AES_256_CBC_SHA    = { 0xXX, 0xXX };
CipherSuite TLS_ECDHE_PSK_WITH_NULL_SHA           = { 0xXX, 0xXX };
```

6. Acknowledgments

The author would like to thank Bodo Moeller, Simon Josefsson, Uri Blumenthal, Pasi Eronen, and the TLS mailing list members for their comments on the 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.

[RFC4346] Dierks, T., Rescorla, E., "The TLS Protocol Version 1.1", RFC 4346, April 200P.

[RFC4279] Eronen, P. and H. Tschofenig, "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)", RFC 4279, December 2005.

[RFC4785] Blumenthal, U., Goel, P., "Pre-Shared Key (PSK) Ciphersuites with NULL Encryption for Transport Layer Security (TLS)", RFC 4785, January 2007.

[RFC4492] Blake-Wilson, S., Bolyard, N., Gupta, V., Hawk, C., Moeller, B., "Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)", RFC 4492, May 2006.

Author's Addresses

Mohamad Badra
LIMOS Laboratory - UMR6158, CNRS
France

Email: badra@isima.fr

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.