Network Working Group
Request for Comments: 5489
Category: Informational

M. Badra
CNRS/LIMOS Laboratory
I. Hajjeh
INEOVATION

March 2009

ECDHE_PSK Cipher Suites for Transport Layer Security (TLS)

Status of This Memo

Copyright Notice

Abstract

This document extends RFC 4279, RFC 4492, and RFC 4785 and specifies a set of cipher suites that use a pre-shared key (PSK) to authenticate an Elliptic Curve Diffie-Hellman exchange with Ephemeral keys (ECDHE). These cipher suites provide Perfect Forward Secrecy (PFS).

Table of Contents

1. Introduction

RFC 4279 specifies cipher suites for supporting TLS using pre-shared symmetric keys that (a) use only symmetric key operations for authentication, (b) use a Diffie-Hellman exchange authenticated with a pre-shared key (PSK), or (c) combine public key authentication of the server with pre-shared key authentication of the client.

RFC 4785 specifies authentication-only cipher suites (with no encryption). These cipher suites are useful when authentication and integrity protection is desired, but confidentiality is not needed or not permitted.

RFC 4492 defines a set of Elliptic Curve Cryptography (ECC)-based cipher suites for TLS and describes the use of ECC certificates for client authentication. In particular, it specifies the use of Elliptic Curve Diffie-Hellman (ECDH) key agreement in a TLS handshake and the use of the Elliptic Curve Digital Signature Algorithm (ECDSA) as a new authentication mechanism.

This document specifies a set of cipher suites that use a PSK to authenticate an ECDH exchange. These cipher suites provide Perfect Forward Secrecy. Some of these cipher suites provide authentication only.

The reader is expected to become familiar with RFC 4279, RFC 4492, and RFC 4785 prior to studying this document.

1.1. Applicability Statement

The cipher suites defined in this document can be negotiated, whatever the negotiated TLS version is.

The applicability statement in [RFC4279] applies to this document as well.

1.2. 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 cipher suites described in this document make use of the elliptic curve (EC) parameter negotiation mechanism defined in RFC 4492. When the cipher suites defined in this document are used, the 'ec_diffie_hellman_psk' case inside the ServerKeyExchange and ClientKeyExchange structure MUST be used instead of the 'psk' case defined in [RFC4279] (i.e., the ServerKeyExchange and ClientKeyExchange messages include the EC Diffie-Hellman parameters in the form specified in Sections 5.4 and 5.7 of [RFC4492]). The PSK identity and identity hint fields have the same meaning and encoding as 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 the ECDH computation as described in Section 5.10 of [RFC4492]. Let Z be the octet string produced by this computation. Next, concatenate a uint16 containing the length of Z (in octets), Z 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 Z.

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

3. ECDHE_PSK-Based Cipher Suites

3.1. ECDHE_PSK Cipher Suites Using the SHA-1 Hash

```
CipherSuite TLS_ECDHE_PSK_WITH_RC4_128_SHA          = {0xC0,0x33};
CipherSuite TLS_ECDHE_PSK_WITH_3DES_EDE_CBC_SHA     = {0xC0,0x34};
CipherSuite TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA      = {0xC0,0x35};
CipherSuite TLS_ECDHE_PSK_WITH_AES_256_CBC_SHA      = {0xC0,0x36};
```

The above four cipher suites match the cipher suites defined in [RFC4279], except that they use an Elliptic Curve Diffie-Hellman exchange [RFC4492] authenticated with a PSK, and:

- The Message Authentication Code (MAC) is the Hashed Message Authentication Code (HMAC) [RFC2104] with SHA-1 as the hash function.

- When negotiated in a version of TLS prior to 1.2, the Pseudo-Random Function (PRF) from that version is used; otherwise, the PRF is the TLS PRF [RFC5246] with SHA-256 as the hash function.

3.2. ECDHE_PSK Cipher Suites Using SHA-2 Hashes

```
CipherSuite TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA256   = {0xC0,0x37};
CipherSuite TLS_ECDHE_PSK_WITH_AES_256_CBC_SHA384   = {0xC0,0x38};
```

The above two cipher suites are the same as the corresponding Advanced Encryption Standard (AES) cipher suites in Section 3.1 above, except for the hash and PRF algorithms, which SHALL be as follows:

- For the cipher suite TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA256:
  - The MAC is HMAC [RFC2104] with SHA-256 as the hash function.
  - When negotiated in a version of TLS prior to 1.2, the PRF from that version is used; otherwise, the PRF is the TLS PRF [RFC5246] with SHA-256 as the hash function.
- For the cipher suite TLS_ECDHE_PSK_WITH_AES_256_CBC_SHA384:
  - The MAC is HMAC [RFC2104] with SHA-384 as the hash function.
  - When negotiated in a version of TLS prior to 1.2, the PRF from that version is used; otherwise the PRF is the TLS PRF [RFC5246] with SHA-384 as the hash function.

4. ECDHE_PSK-Based Cipher Suites with NULL Encryption

4.1. ECDHE_PSK Cipher Suite Using the SHA-1 Hash with NULL Encryption

The following cipher suite matches the cipher suites defined in Section 3.1, except that we define a suite with NULL encryption.

```
CipherSuite TLS_ECDHE_PSK_WITH_NULL_SHA             = {0xC0,0x39};
```

4.2. ECDHE_PSK Cipher Suites Using SHA-2 Hashes with NULL Encryption

The following two cipher suites are the same as the corresponding cipher suites in Section 3.2, but with NULL encryption (instead of AES).

```
CipherSuite TLS_ECDHE_PSK_WITH_NULL_SHA256          = {0xC0,0x3A};
CipherSuite TLS_ECDHE_PSK_WITH_NULL_SHA384          = {0xC0,0x3B};
```

5. Security Considerations

The security considerations described throughout [RFC5246], [RFC4785], [RFC4492], and [RFC4279] apply here as well. In particular, as the authentication-only cipher suites (with no encryption) defined here do not support confidentiality, care should be taken not to send sensitive information (such as passwords) over connections protected with one of the cipher suites with NULL encryption defined in Section 4 of this document.

Implementers and administrators should monitor the general statements on recommended cryptographic algorithms (e.g., SHA-1 hash function) that are published from time to time by various forums, including the IETF, as a base for the portfolio they support and the policies for strength of function acceptable for the cipher suites they set.

6. IANA Considerations

This document defines the following new cipher suites, whose values have been assigned from the TLS Cipher Suite registry defined in [RFC5246].

```
CipherSuite TLS_ECDHE_PSK_WITH_RC4_128_SHA          = {0xC0,0x33};
CipherSuite TLS_ECDHE_PSK_WITH_3DES_EDE_CBC_SHA     = {0xC0,0x34};
CipherSuite TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA      = {0xC0,0x35};
CipherSuite TLS_ECDHE_PSK_WITH_AES_256_CBC_SHA      = {0xC0,0x36};
CipherSuite TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA256   = {0xC0,0x37};
CipherSuite TLS_ECDHE_PSK_WITH_AES_256_CBC_SHA384   = {0xC0,0x38};
CipherSuite TLS_ECDHE_PSK_WITH_NULL_SHA             = {0xC0,0x39};
CipherSuite TLS_ECDHE_PSK_WITH_NULL_SHA256          = {0xC0,0x3A};
CipherSuite TLS_ECDHE_PSK_WITH_NULL_SHA384          = {0xC0,0x3B};
```

7. Acknowledgments

The author appreciates Alfred Hoenes for his detailed review and effort on resolving issues in discussion. The author would like to acknowledge Bodo Moeller, Simon Josefsson, Uri Blumenthal, Pasi Eronen, Paul Hoffman, Joseph Salowey, Mark Tillinghast, and the TLS mailing list members for their comments on the document.

8. Normative References

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

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

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

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

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

Authors' Addresses

Mohamad Badra
CNRS/LIMOS Laboratory
Campus de cezeaux, Bat. ISIMA
Aubiere 63170
France

EMail: badra@isima.fr

Ibrahim Hajjeh
INEOVATION
France

EMail: ibrahim.hajjeh@ineovation.fr