Abstract

This document describe how the Salsa20 stream cipher can be used in the Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) protocols.

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 May 31, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.
This document describe how the Salsa20 stream cipher can be used in the Transport Layer Security (TLS) version 1.0 [RFC2246], TLS version 1.1 [RFC4346], and TLS version 1.2 [RFC5246] protocols, as well as in the Datagram Transport Layer Security (DTLS) versions 1.0 [RFC4347] and 1.2 [RFC6347]. It can also be used with Secure Sockets Layer (SSL) version 3.0 [RFC6101].

Salsa20 [SALSA20SPEC] is a stream cipher that has been designed for high performance in software implementations. The cipher has compact implementation and uses few resources and inexpensive operations that makes it suitable for implementation on a wide range of architectures. It has been designed to prevent leakage of information through side channel analysis, has a simple and fast key setup and provides good overall performance. Salsa20 is one of the ciphers selected as part of the eSTREAM portfolio of stream ciphers [ESTREAM].

Recent attacks [CBC-ATTACK] have indicated problems with CBC-mode cipher suites in TLS and DTLS as well as issues with the only supported stream cipher (RC4) [RC4-ATTACK]. While the existing AEAD ciphersuites address these issues, concerns about their performance,
on general purpose CPUs, are sometimes raised [AEAD-PERFORMANCE]. Moreover, the DTLS protocol cannot take advantage of the fast RC4 stream cipher because it does not provide random access in the key stream.

Therefore, a new stream cipher to replace RC4 and address all the previous issues is needed. It is the purpose of this document to describe a secure stream cipher for both TLS and DTLS that is comparable to RC4 in speed on a wide range of platforms.

2. Salsa20 Cipher Suites

   The following variants of Salsa20 are specified. The variants provide a range of performance and security that can be selected as appropriate.

   ESTREAM_SALSA20: Salsa20 with 12 rounds and a 256 bit key. This cipher is the high performant eSTREAM Salsa20 with 256 bit key.

   SALSA20: Salsa20 with 20 rounds and a 256 bit key. This is the original (conservative with respect to security) variant of Salsa20.

In the next sections different ciphersuites are defined that utilize the Salsa20 cipher combined with various MAC methods

In all cases, the pseudorandom function (PRF) for TLS 1.2 is the TLS PRF with SHA-256 as the hash function. When used with TLS versions prior to 1.2, the PRF is calculated as specified in the appropriate version of the TLS specification.

2.1. Salsa20 Cipher Suites with HMAC-SHA1

   The following CipherSuites are defined: (note that the third column contains the suggested to IANA ciphersuite numbers)
Note that Salsa20 requires a 64-bit nonce. That nonce is updated on the encryption of every TLS record, and is set to be the 64-bit TLS record sequence number. In case of DTLS the 64-bit nonce is formed as the concatenation of the 16-bit epoch with the 48-bit sequence number.

The RSA, DHE_RSA, ECDHE_RSA, ECDHE_ECDSA, PSK, DHE_PSK, RSA_PSK, ECDHE_PSK key exchanges are performed as defined in [RFC5246], [RFC4492], and [RFC5489].

The MAC algorithm used in the ciphersuites above is HMAC-SHA1 [RFC6234].

3. The TLS GenericStreamCipher

The ciphersuites defined in this document differ from the TLS RC4 ciphersuites that have been the basis for the definition of GenericStreamCipher. Unlike RC4, Salsa20 requires a nonce per record. This however, does not affect the description of the GenericStreamCipher if one assumes that a nonce is optional and depends on the cipher’s characteristics (in that case RC4 uses a 0 byte nonce, and Salsa20 an 8-byte nonce).

As specified in TLS [RFC5246] the MAC is computed before encryption and the stream cipher encrypts the entire block, including the MAC.
4. Acknowledgements

The authors would like to thank D. J. Bernstein, David McGrew, Wan-Teh Chang, and Adam Langley for discussion and suggestions.

5. IANA Considerations

IANA is requested to allocate the following numbers in the TLS Cipher Suite Registry (note that the third column contains the suggested ciphersuite numbers):

<table>
<thead>
<tr>
<th>Ciphersuite Description</th>
<th>First Number</th>
<th>Second Number</th>
<th>Third Number</th>
<th>Fourth Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLS_RSA_WITH_ESTREAM_SALSA20_SHA1</td>
<td>0xTBD</td>
<td>0xTBD</td>
<td>0xE4</td>
<td>0x10</td>
</tr>
<tr>
<td>TLS_RSA_WITH_SALSA20_SHA1</td>
<td>0xTBD</td>
<td>0xTBD</td>
<td>0xE4</td>
<td>0x11</td>
</tr>
<tr>
<td>TLS_ECDHE_RSA_WITH_ESTREAM_SALSA20_SHA1</td>
<td>0xTBD</td>
<td>0xTBD</td>
<td>0xE4</td>
<td>0x12</td>
</tr>
<tr>
<td>TLS_ECDHE_RSA_WITH_SALSA20_SHA1</td>
<td>0xTBD</td>
<td>0xTBD</td>
<td>0xE4</td>
<td>0x13</td>
</tr>
<tr>
<td>TLS_ECDHE_ECDSA_WITH_ESTREAM_SALSA20_SHA1</td>
<td>0xTBD</td>
<td>0xTBD</td>
<td>0xE4</td>
<td>0x14</td>
</tr>
<tr>
<td>TLS_ECDHE_ECDSA_WITH_SALSA20_SHA1</td>
<td>0xTBD</td>
<td>0xTBD</td>
<td>0xE4</td>
<td>0x15</td>
</tr>
<tr>
<td>TLS_PSK_WITH_ESTREAM_SALSA20_SHA1</td>
<td>0xTBD</td>
<td>0xTBD</td>
<td>0xE4</td>
<td>0x16</td>
</tr>
<tr>
<td>TLS_PSK_WITH_SALSA20_SHA1</td>
<td>0xTBD</td>
<td>0xTBD</td>
<td>0xE4</td>
<td>0x17</td>
</tr>
<tr>
<td>TLS_ECDHE_PSK_WITH_ESTREAM_SALSA20_SHA1</td>
<td>0xTBD</td>
<td>0xTBD</td>
<td>0xE4</td>
<td>0x18</td>
</tr>
<tr>
<td>TLS_ECDHE_PSK_WITH_SALSA20_SHA1</td>
<td>0xTBD</td>
<td>0xTBD</td>
<td>0xE4</td>
<td>0x19</td>
</tr>
<tr>
<td>TLS_RSA_PSK_WITH_ESTREAM_SALSA20_SHA1</td>
<td>0xTBD</td>
<td>0xTBD</td>
<td>0xE4</td>
<td>0x1A</td>
</tr>
<tr>
<td>TLS_RSA_PSK_WITH_SALSA20_SHA1</td>
<td>0xTBD</td>
<td>0xTBD</td>
<td>0xE4</td>
<td>0x1B</td>
</tr>
<tr>
<td>TLS_DHE_PSK_WITH_ESTREAM_SALSA20_SHA1</td>
<td>0xTBD</td>
<td>0xTBD</td>
<td>0xE4</td>
<td>0x1C</td>
</tr>
<tr>
<td>TLS_DHE_PSK_WITH_SALSA20_SHA1</td>
<td>0xTBD</td>
<td>0xTBD</td>
<td>0xE4</td>
<td>0x1D</td>
</tr>
<tr>
<td>TLS_DHE_RSA_WITH_ESTREAM_SALSA20_SHA1</td>
<td>0xTBD</td>
<td>0xTBD</td>
<td>0xE4</td>
<td>0x1E</td>
</tr>
<tr>
<td>TLS_DHE_RSA_WITH_SALSA20_SHA1</td>
<td>0xTBD</td>
<td>0xTBD</td>
<td>0xE4</td>
<td>0x1F</td>
</tr>
</tbody>
</table>

6. Security Considerations

The security of Salsa20 is discussed in the Salsa20 security [SALSA20-SECURITY] paper. At the time of writing this document, there are no known significant security problems with the eSTREAM variant of Salsa20, nor with the original 20 round variant. As of early 2013, the best cryptanalysis breaks 8 out of 20 rounds to recover the 256-bit secret key in $2^{251}$ operations, using $2^{31}$ keystream pairs (see [SALSA20-ATTACK]). For more background, see the eSTREAM report [ESTREAM].
There are no ciphersuites defined in this document that utilize the variant of Salsa20 with 128-bit key material, because (due to the design of Salsa20) they provide no performance advantage over the 256-bit variant.

This document should not introduce any other security considerations than those that directly follow from any use of the stream cipher Salsa20 and those that directly follow from introducing any set of stream cipher suites into TLS and DTLS.

7. Algorithm Selection Background

This draft uses Salsa20, a winner of an international competion of stream ciphers (eStream), which is easily implementable without leaking information through side-channels, i.e. timing and power attacks.

Suggestions has been made to instead use Chacha [CHACHASPEC], a derivative of Salsa20 that has been shown to be 7% faster in hardware and occupy 10% less space [VLSI-IMPL]. In our opinion the performance benefits don’t justify switching from a winner of an international competition to another algorithm (even if it is a derivative of it).

This draft adds a new cipher to existing TLS and DTLS implementations which is combined with the existing MAC algorithms in TLS (i.e., HMAC-SHA1). That allows the new cipher to replace the, currently known to be broken, RC4 ciphersuites, in all TLS versions.

8. References

8.1. Normative References


8.2. Informative References


Authors’ Addresses

Simon Josefsson
SJD AB
Email: simon@josefsson.org
URI: http://josefsson.org/

Joachim Strombergson
Secworks Sweden AB
Email: joachim@secworks.se
URI: http://secworks.se/

Nikos Mavrogiannopoulos
Red Hat
Email: nmav@redhat.com