Network Working Group
Internet-Draft
Intended status: Informational
Expires: June 9, 2014

N. Mavrogiannopoulos
Red Hat
J. Strombergson
Secworks Sweden AB
S. Josefsson
SJD AB

December 6, 2013

The ChaCha Stream Cipher for Transport Layer Security
draft-mavrogiannopoulos-chacha-tls-00

Abstract

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

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

This document describe how the Chacha 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].

Chacha [CHACHASPEC] 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. It is a variant of Salsa20 [SALSA20SPEC] which is one of the selected ciphers in the eSTREAM portfolio [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. Chacha Cipher Suites

The variant of Chacha used in this draft is Chacha with 20 rounds and a 256 bit key. This is the conservative with respect to security variant of the Chacha family. Test vectors for this cipher can be found at [I-D.strombergson-chacha-test-vectors].

In the next sections different ciphersuites are defined that utilize the Chacha 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. Chacha Cipher Suites with HMAC-SHA1

The following CipherSuites are defined: (note that the third column contains the suggested to IANA ciphersuite numbers)

TLS_RSA_WITH_CHACHA_SHA1	= {0xTBD, 0xTBD}	{0xE5, 0x00}
TLS_ECDHE_RSA_WITH_CHACHA_SHA1	= {0xTBD, 0xTBD}	{0xE5, 0x01}
TLS_ECDHE_ECDSA_WITH_CHACHA_SHA1	= {0xTBD, 0xTBD}	{0xE5, 0x02}
TLS_PSK_WITH_CHACHA_SHA1	= {0xTBD, 0xTBD}	{0xE5, 0x03}
TLS_ECDHE_PSK_WITH_CHACHA_SHA1	= {0xTBD, 0xTBD}	{0xE5, 0x04}
TLS_RSA_PSK_WITH_CHACHA_SHA1	= {0xTBD, 0xTBD}	{0xE5, 0x05}
TLS_DHE_PSK_WITH_CHACHA_SHA1	= {0xTBD, 0xTBD}	{0xE5, 0x06}
TLS_DHE_RSA_WITH_CHACHA_SHA1	= {0xTBD, 0xTBD}	{0xE5, 0x07}

Note that Chacha 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, Chacha 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 Chacha 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 Zooko Wilcox-OHearn and Samuel Neves for suggestions that led to this draft.

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):

TLS_RSA_WITH_CHACHA_SHA1	= {0xTBD, 0xTBD}	{0xE5, 0x00}
TLS_ECDHE_RSA_WITH_CHACHA_SHA1	= {0xTBD, 0xTBD}	{0xE5, 0x01}
TLS_ECDHE_ECDSA_WITH_CHACHA_SHA1	= {0xTBD, 0xTBD}	{0xE5, 0x02}
TLS_PSK_WITH_CHACHA_SHA1	= {0xTBD, 0xTBD}	{0xE5, 0x03}
TLS_ECDHE_PSK_WITH_CHACHA_SHA1	= {0xTBD, 0xTBD}	{0xE5, 0x04}
TLS_RSA_PSK_WITH_CHACHA_SHA1	= {0xTBD, 0xTBD}	{0xE5, 0x05}
TLS_DHE_PSK_WITH_CHACHA_SHA1	= {0xTBD, 0xTBD}	{0xE5, 0x06}
TLS_DHE_RSA_WITH_CHACHA_SHA1	= {0xTBD, 0xTBD}	{0xE5, 0x07}

6. Security Considerations

Chacha follows the same basic principle as Salsa20, a cipher with significant security review [SALSA20-SECURITY][ESTREAM]. At the time of writing this document, there are no known significant security problems with either cipher, and Chacha is shown to be more resistant in certain attacks than Salsa20 [SALSA20-ATTACK]. Furthermore Chacha was used as the core of the BLAKE hash function, a SHA3 finalist, that had received considerable cryptanalytic attention [NIST-SHA3].

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

7. References

7.1. Normative References

[RFC2246] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006.

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

[RFC4347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", RFC 4347, April 2006.

[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.

[RFC5489] Badra, M. and I. Hajjeh, "ECDHE_PSK Cipher Suites for Transport Layer Security (TLS)", RFC 5489, March 2009.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, January 2012.

[RFC6234] Eastlake, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, May 2011.

[CHACHASPEC]
Bernstein, D., "Chacha, a variant of Salsa20", WWW http://cr.yp.to/chacha/chacha-20080128.pdf, January 2008.

7.2. Informative References

[I-D.strombergson-chacha-test-vectors]
Strombergson, J., "Test Vectors for the Stream Cipher ChaCha", draft-strombergson-chacha-test-vectors-00 (work in progress), October 2013.

[SALSA20SPEC]
Bernstein, D., "Salsa20 specification", WWW http://cr.yp.to/snuffle/spec.pdf, April 2005.

[RFC6101] Freier, A., Karlton, P., and P. Kocher, "The Secure Sockets Layer (SSL) Protocol Version 3.0", RFC 6101, August 2011.

[SALSA20-SECURITY]
Bernstein, D., "Salsa20 security",
WWW http://cr.yp.to/snuffle/security.pdf, April 2005.

[ESTREAM] Babbage, S., DeCanniere, C., Cantenaut, A., Cid, C., Gilbert, H., Johansson, T., Parker, M., Preneel, B., Rijmen, V., and M. Robshaw, "The eSTREAM Portfolio (rev. 1)", WWW http://www.ecrypt.eu.org/stream/finallist.html, September 2008.

[CBC-ATTACK]
AlFardan, N. and K. Paterson, "Lucky Thirteen: Breaking the TLS and DTLS Record Protocols", IEEE Symposium on Security and Privacy , 2013.

[RC4-ATTACK]
Isobe, T., Ohigashi, T., Watanabe, Y., and M. Morii, "Full Plaintext Recovery Attack on Broadcast RC4", International Workshop on Fast Software Encryption , 2013.

[AEAD-PERFORMANCE]
Krovetz, T. and P. Rogaway, "The Software Performance of Authenticated-Encryption Modes", International Workshop on Fast Software Encryption , 2011.

[SALSA20-ATTACK]
Aumasson, J-P., Fischer, S., Khazaei, S., Meier, W., and C. Rechberger, "New Features of Latin Dances: Analysis of Salsa, ChaCha, and Rumba",
WWW http://eprint.iacr.org/2007/472.pdf, 2007.

[VLSI-IMPL]
Henzen, L., Carbognani, F., and W. Fichtner, "VLSI hardware evaluation of the stream ciphers Salsa20 and ChaCha, and the compression function Rumba", 2008.

[NIST-SHA3]
Chang, S., Burr, W., Kelsey, J., Paul, S., and L. Bassham, "Third-Round Report of the SHA-3 Cryptographic Hash Algorithm Competition",
WWW http://dx.doi.org/10.6028/NIST.IR.7896, 2012.

Authors' Addresses

Nikos Mavrogiannopoulos
Red Hat

Email: nmav@redhat.com

Joachim Strombergson
Secworks Sweden AB

Email: joachim@secworks.se
URI: http://secworks.se/

Simon Josefsson
SJD AB

Email: simon@josefsson.org
URI: http://josefsson.org/