Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 10 January 2022

N. Aviram

9 July 2021

Deprecating Obsolete Key Exchange Methods in TLS
draft-aviram-tls-deprecate-obsolete-kex-00

Abstract

This document deprecates the use of RSA key exchange in TLS, and limits the use of Diffie Hellman key exchange over a finite field such as to avoid known vulnerabilities or improper security properties.

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 https://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 10 January 2022.

Table of Contents

1. Introduction

TLS supports a variety of key exchange algorithms, including RSA and Diffie Hellman over a finite field, as well as elliptic curve Diffie Hellman (ECDH). Diffie Hellman key exchange, over any group, may use either long-lived or ephemeral secrets. Diffie Hellman key exchange with long-lived secrets over a finite field is already deprecated in [deprecate-ffdh]. This document focuses on Diffie Hellman over a finite field with ephemeral secrets (FFDHE), as well as RSA key exchange.

Recent years have brought to light several security concerns regarding FFDHE key exchange that stem from implementation choices. Additionally, RSA key exchange suffers from security problems that are independent of implementation choices, as well as problems that stem purely from the difficulty of implementing security countermeasures correctly.

At a rough glance, the problems affecting FFDHE are as follows:

1. FFDHE suffers from interoperability problems, because there is no mechanism for negotiating the group size, and some implementations only support small group sizes; see [RFC7919], Section 1.

2. In practice, some operators use 1024 bit FFDHE groups, since this is the maximum size that ensures wide support; see [RFC7919], Section 1. This size leaves only a small security margin vs. the current discrete log record, which stands at 795 bits [DLOG795].

3. Expanding on the previous point, a handful of very large computations would allow cheaply decrypting a relatively large fraction of FFDHE traffic [weak-dh].

4. When secrets are not fully ephemeral, FFDHE suffers from the [Raccoon] side channel attack.

5. FFDHE groups may have small subgroups, which may enable several attacks [subgroups].

And the problems affecting RSA key exchange are as follows:

1. RSA key exchange offers no forward secrecy, by construction.

2. RSA key exchange may be vulnerable to Bleichenbacher's attack [BLEI]. Experience shows that variants of this attack arise every few years, because implementing the relevant countermeasure correctly is difficult; see [ROBOT], [NEW-BLEI], [DROWN].

3. In addition to the above point, there is no convenient mechanism in TLS for domain separation of keys. Therefore, a single endpoint that is vulnerable to Bleichenbacher's attack would affect all endpoints sharing the same RSA key; see [XPROT], [DROWN].

Given these problems, this document updates [RFC4346], [RFC5246], [RFC4162], [RFC6347], [RFC5932], [RFC5288], [RFC6209], [RFC6367], [RFC8422], [RFC5289], and [RFC5469] to deprecate RSA key exchange in TLS, and limit use of FFDHE such that it provides acceptable security properties.

1.1. Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. RSA

Clients and servers MUST NOT offer RSA cipher suites in TLS 1.0, 1.1, and 1.2 connections. This includes all cipher suites listed in the following table. Note that these cipher suites are already marked as not recommended in the "TLS Cipher Suites" registry.

Ciphersuite	Reference
TLS_RSA_WITH_NULL_MD5	[RFC5246]
TLS_RSA_WITH_NULL_SHA	[RFC5246]
TLS_RSA_EXPORT_WITH_RC4_40_MD5	[RFC4346] [RFC6347]

TLS_RSA_WITH_RC4_128_MD5	[RFC5246] [RFC6347]
TLS_RSA_WITH_RC4_128_SHA	[RFC5246] [RFC6347]
TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5	[RFC4346]
TLS_RSA_WITH_IDEA_CBC_SHA	[RFC5469][SC-tls-des-idea-ciphers-to-historic]
TLS_RSA_EXPORT_WITH_DES40_CBC_SHA	[RFC4346]
TLS_RSA_WITH_DES_CBC_SHA	[RFC5469][SC-tls-des-idea-ciphers-to-historic]
TLS_RSA_WITH_3DES_EDE_CBC_SHA	[RFC5246]
TLS_RSA_PSK_WITH_NULL_SHA	[RFC4785]
TLS_RSA_WITH_AES_128_CBC_SHA	[RFC5246]
TLS_RSA_WITH_AES_256_CBC_SHA	[RFC5246]
TLS_RSA_WITH_NULL_SHA256	[RFC5246]
TLS_RSA_WITH_AES_128_CBC_SHA256	[RFC5246]
TLS_RSA_WITH_AES_256_CBC_SHA256	[RFC5246]

TLS_RSA_WITH_CAMELLIA_128_CBC_SHA	[RFC5932]
TLS_RSA_WITH_CAMELLIA_256_CBC_SHA	[RFC5932]
TLS_RSA_PSK_WITH_RC4_128_SHA	[RFC4279] [RFC6347]
TLS_RSA_PSK_WITH_3DES_EDE_CBC_SHA	[RFC4279]
TLS_RSA_PSK_WITH_AES_128_CBC_SHA	[RFC4279]
TLS_RSA_PSK_WITH_AES_256_CBC_SHA	[RFC4279]
TLS_RSA_WITH_SEED_CBC_SHA	[RFC4162]
TLS_RSA_WITH_AES_128_GCM_SHA256	[RFC5288]
TLS_RSA_WITH_AES_256_GCM_SHA384	[RFC5288]
TLS_RSA_PSK_WITH_AES_128_GCM_SHA256	[RFC5487]
TLS_RSA_PSK_WITH_AES_256_GCM_SHA384	[RFC5487]
TLS_RSA_PSK_WITH_AES_128_CBC_SHA256	[RFC5487]
TLS_RSA_PSK_WITH_AES_256_CBC_SHA384	[RFC5487]

TLS_RSA_PSK_WITH_NULL_SHA256	[RFC5487]
TLS_RSA_PSK_WITH_NULL_SHA384	[RFC5487]
TLS_RSA_WITH_CAMELLIA_128_CBC_SHA256	[RFC5932]
TLS_RSA_WITH_CAMELLIA_256_CBC_SHA256	[RFC5932]
TLS_RSA_WITH_ARIA_128_CBC_SHA256	[RFC6209]
TLS_RSA_WITH_ARIA_256_CBC_SHA384	[RFC6209]
TLS_RSA_WITH_ARIA_128_GCM_SHA256	[RFC6209]
TLS_RSA_WITH_ARIA_256_GCM_SHA384	[RFC6209]
TLS_RSA_PSK_WITH_ARIA_128_CBC_SHA256	[RFC6209]
TLS_RSA_PSK_WITH_ARIA_256_CBC_SHA384	[RFC6209]
TLS_RSA_PSK_WITH_ARIA_128_GCM_SHA256	[RFC6209]
TLS_RSA_PSK_WITH_ARIA_256_GCM_SHA384	[RFC6209]
TLS_RSA_WITH_CAMELLIA_128_GCM_SHA256	[RFC6367]

TLS_RSA_WITH_CAMELLIA_256_GCM_SHA384		[RFC6367]
TLS_RSA_PSK_WITH_CAMELLIA_128_GCM_SHA256		[RFC6367]
TLS_RSA_PSK_WITH_CAMELLIA_256_GCM_SHA384		[RFC6367]
TLS_RSA_PSK_WITH_CAMELLIA_128_CBC_SHA256		[RFC6367]
TLS_RSA_PSK_WITH_CAMELLIA_256_CBC_SHA384		[RFC6367]
TLS_RSA_WITH_AES_128_CCM		[RFC6655]
TLS_RSA_WITH_AES_256_CCM		[RFC6655]
TLS_RSA_WITH_AES_128_CCM_8		[RFC6655]
TLS_RSA_WITH_AES_256_CCM_8		[RFC6655]
TLS_RSA_PSK_WITH_CHACHA20_POLY1305_SHA256		[RFC7905]

Table 1

3. Ephemeral Finite Field Diffie Hellman

Clients and servers MAY offer fully ephemeral FFDHE cipher suites in TLS 1.0, 1.1, and 1.2 connections, under the following conditions:

1. The secret DH key is fully ephemeral, that is, a fresh DH exponent is generated for each TLS connection. Note that this requirement is also specified in [deprecate-ffdh].

2. The group is one of the following well-known groups described in [RFC7919]: ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192.

We note that previously, supporting the broadest range of clients would have required supporting either RSA key exchange, or 1024-bit FFDHE. This is no longer the case, and it is possible to support most clients released since circa 2015 using 2048-bit FFDHE, or more modern key exchange methods, and without RSA key exchange [server_side_tls].

The above requirements apply to all cipher suites listed in the following table.

Ciphersuite	Reference
TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA	[RFC4346]
TLS_DHE_DSS_WITH_DES_CBC_SHA	[RFC5469][SC-tls-des-idea-ciphers-to-historic]
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA	[RFC5246]
TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA	[RFC4346]
TLS_DHE_RSA_WITH_DES_CBC_SHA	[RFC5469][SC-tls-des-idea-ciphers-to-historic]
TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA	[RFC5246]
TLS_DHE_PSK_WITH_NULL_SHA	[RFC4785]

| TLS_DHE_DSS_WITH_AES_128_CBC_SHA | [RFC5246]
|
+---
+---+
| TLS_DHE_RSA_WITH_AES_128_CBC_SHA | [RFC5246]
|
+---
+---+
| TLS_DHE_DSS_WITH_AES_256_CBC_SHA | [RFC5246]
|
+---
+---+
| TLS_DHE_RSA_WITH_AES_256_CBC_SHA | [RFC5246]
|
+---
+---+
| TLS_DHE_DSS_WITH_AES_128_CBC_SHA256 | [RFC5246]
|
+---
+---+
| TLS_DHE_DSS_WITH_CAMELLIA_128_CBC_SHA | [RFC5932]
|
+---
+---+
| TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA | [RFC5932]
|
+---
+---+
| TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 | [RFC5246]
|
+---
+---+
| TLS_DHE_DSS_WITH_AES_256_CBC_SHA256 | [RFC5246]
|
+---
+---+
| TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 | [RFC5246]
|
+---
+---+
| TLS_DHE_DSS_WITH_CAMELLIA_256_CBC_SHA | [RFC5932]
|
+---
+---+
| TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA | [RFC5932]
|
+---
+---+
| TLS_DHE_PSK_WITH_RC4_128_SHA | [RFC4279]
[RFC6347] |
+---

TLS_DHE_PSK_WITH_3DES_EDE_CBC_SHA	[RFC4279]
TLS_DHE_PSK_WITH_AES_128_CBC_SHA	[RFC4279]
TLS_DHE_PSK_WITH_AES_256_CBC_SHA	[RFC4279]
TLS_DHE_DSS_WITH_SEED_CBC_SHA	[RFC4162]
TLS_DHE_RSA_WITH_SEED_CBC_SHA	[RFC4162]
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	[RFC5288]
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	[RFC5288]
TLS_DHE_DSS_WITH_AES_128_GCM_SHA256	[RFC5288]
TLS_DHE_DSS_WITH_AES_256_GCM_SHA384	[RFC5288]
TLS_DHE_PSK_WITH_AES_128_GCM_SHA256	[RFC5487]
TLS_DHE_PSK_WITH_AES_256_GCM_SHA384	[RFC5487]
TLS_DHE_PSK_WITH_AES_128_CBC_SHA256	[RFC5487]
TLS_DHE_PSK_WITH_AES_256_CBC_SHA384	[RFC5487]

TLS_DHE_PSK_WITH_NULL_SHA256	[RFC5487]
TLS_DHE_PSK_WITH_NULL_SHA384	[RFC5487]
TLS_DHE_DSS_WITH_CAMELLIA_128_CBC_SHA256	[RFC5932]
TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA256	[RFC5932]
TLS_DHE_DSS_WITH_CAMELLIA_256_CBC_SHA256	[RFC5932]
TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA256	[RFC5932]
TLS_DHE_DSS_WITH_ARIA_128_CBC_SHA256	[RFC6209]
TLS_DHE_DSS_WITH_ARIA_256_CBC_SHA384	[RFC6209]
TLS_DHE_RSA_WITH_ARIA_128_CBC_SHA256	[RFC6209]
TLS_DHE_RSA_WITH_ARIA_256_CBC_SHA384	[RFC6209]
TLS_DHE_RSA_WITH_ARIA_128_GCM_SHA256	[RFC6209]
TLS_DHE_RSA_WITH_ARIA_256_GCM_SHA384	[RFC6209]
TLS_DHE_DSS_WITH_ARIA_128_GCM_SHA256	[RFC6209]

```
|
   +----------------------------------------
+-----------------------------------------------+
   | TLS_DHE_DSS_WITH_ARIA_256_GCM_SHA384    | [RFC6209]
|
   +----------------------------------------
+-----------------------------------------------+
   | TLS_DHE_PSK_WITH_ARIA_128_CBC_SHA256    | [RFC6209]
|
   +----------------------------------------
+-----------------------------------------------+
   | TLS_DHE_PSK_WITH_ARIA_256_CBC_SHA384    | [RFC6209]
|
   +----------------------------------------
+-----------------------------------------------+
   | TLS_DHE_PSK_WITH_ARIA_128_GCM_SHA256    | [RFC6209]
|
   +----------------------------------------
+-----------------------------------------------+
   | TLS_DHE_PSK_WITH_ARIA_256_GCM_SHA384    | [RFC6209]
|
   +----------------------------------------
+-----------------------------------------------+
   |TLS_DHE_RSA_WITH_CAMELLIA_128_GCM_SHA256 | [RFC6367]
|
   +----------------------------------------
+-----------------------------------------------+
   |TLS_DHE_RSA_WITH_CAMELLIA_256_GCM_SHA384 | [RFC6367]
|
   +----------------------------------------
+-----------------------------------------------+
   |TLS_DHE_DSS_WITH_CAMELLIA_128_GCM_SHA256 | [RFC6367]
|
   +----------------------------------------
+-----------------------------------------------+
   |TLS_DHE_DSS_WITH_CAMELLIA_256_GCM_SHA384 | [RFC6367]
|
   +----------------------------------------
+-----------------------------------------------+
   |TLS_DHE_PSK_WITH_CAMELLIA_128_GCM_SHA256 | [RFC6367]
|
   +----------------------------------------
+-----------------------------------------------+
   |TLS_DHE_PSK_WITH_CAMELLIA_256_GCM_SHA384 | [RFC6367]
|
   +----------------------------------------
+-----------------------------------------------+
   |TLS_DHE_PSK_WITH_CAMELLIA_128_CBC_SHA256 | [RFC6367]
|
   +----------------------------------------
+-----------------------------------------------+
```

TLS_DHE_PSK_WITH_CAMELLIA_256_CBC_SHA384	[RFC6367]
TLS_DHE_RSA_WITH_AES_128_CCM	[RFC6655]
TLS_DHE_RSA_WITH_AES_256_CCM	[RFC6655]
TLS_DHE_RSA_WITH_AES_128_CCM_8	[RFC6655]
TLS_DHE_RSA_WITH_AES_256_CCM_8	[RFC6655]
TLS_DHE_PSK_WITH_AES_128_CCM	[RFC6655]
TLS_DHE_PSK_WITH_AES_256_CCM	[RFC6655]
TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256	[RFC7905]
TLS_DHE_PSK_WITH_CHACHA20_POLY1305_SHA256	[RFC7905]

Table 2

Note that FFDH cipher suites are already deprecated in [deprecate-ffdh].

4. IANA Considerations

This document makes no requests to IANA. Note that all cipher suites listed in Section 2 are already marked as not recommended in the "TLS Cipher Suites" registry.

5. Security Considerations

This document is entirely about security.

6. Acknowledgments

This document was inspired by discussion on the TLS WG mailing list and a suggestion by Filippo Valsorda following the release of the [Raccoon] attack. Thanks to Christopher A. Wood and Carrick D. Bartle for useful feedback, discussions, and ideas.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4162] Lee, H.J., Yoon, J.H., and J.I. Lee, "Addition of SEED Cipher Suites to Transport Layer Security (TLS)", RFC 4162, DOI 10.17487/RFC4162, August 2005, <https://www.rfc-editor.org/info/rfc4162>.

[RFC4279] Eronen, P., Ed. and H. Tschofenig, Ed., "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)", RFC 4279, DOI 10.17487/RFC4279, December 2005, <https://www.rfc-editor.org/info/rfc4279>.

[RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, DOI 10.17487/RFC4346, April 2006, <https://www.rfc-editor.org/info/rfc4346>.

[RFC4785] Blumenthal, U. and P. Goel, "Pre-Shared Key (PSK) Ciphersuites with NULL Encryption for Transport Layer Security (TLS)", RFC 4785, DOI 10.17487/RFC4785, January 2007, <https://www.rfc-editor.org/info/rfc4785>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <https://www.rfc-editor.org/info/rfc5246>.

[RFC5288] Salowey, J., Choudhury, A., and D. McGrew, "AES Galois Counter Mode (GCM) Cipher Suites for TLS", RFC 5288, DOI 10.17487/RFC5288, August 2008, <https://www.rfc-editor.org/info/rfc5288>.

[RFC5289] Rescorla, E., "TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM)", RFC 5289, DOI 10.17487/RFC5289, August 2008, <https://www.rfc-editor.org/info/rfc5289>.

[RFC5469] Eronen, P., Ed., "DES and IDEA Cipher Suites for Transport Layer Security (TLS)", RFC 5469, DOI 10.17487/RFC5469, February 2009, <https://www.rfc-editor.org/info/rfc5469>.

[RFC5487] Badra, M., "Pre-Shared Key Cipher Suites for TLS with SHA-256/384 and AES Galois Counter Mode", RFC 5487, DOI 10.17487/RFC5487, March 2009, <https://www.rfc-editor.org/info/rfc5487>.

[RFC5932] Kato, A., Kanda, M., and S. Kanno, "Camellia Cipher Suites for TLS", RFC 5932, DOI 10.17487/RFC5932, June 2010, <https://www.rfc-editor.org/info/rfc5932>.

[RFC6209] Kim, W., Lee, J., Park, J., and D. Kwon, "Addition of the ARIA Cipher Suites to Transport Layer Security (TLS)", RFC 6209, DOI 10.17487/RFC6209, April 2011, <https://www.rfc-editor.org/info/rfc6209>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/info/rfc6347>.

[RFC6367] Kanno, S. and M. Kanda, "Addition of the Camellia Cipher Suites to Transport Layer Security (TLS)", RFC 6367, DOI 10.17487/RFC6367, September 2011, <https://www.rfc-editor.org/info/rfc6367>.

[RFC6655] McGrew, D. and D. Bailey, "AES-CCM Cipher Suites for Transport Layer Security (TLS)", RFC 6655, DOI 10.17487/RFC6655, July 2012, <https://www.rfc-editor.org/info/rfc6655>.

[RFC7905] Langley, A., Chang, W., Mavrogiannopoulos, N., Strombergson, J., and S. Josefsson, "ChaCha20-Poly1305 Cipher Suites for Transport Layer Security (TLS)", RFC 7905, DOI 10.17487/RFC7905, June 2016, <https://www.rfc-editor.org/info/rfc7905>.

[RFC7919] Gillmor, D., "Negotiated Finite Field Diffie-Hellman Ephemeral Parameters for Transport Layer Security (TLS)", RFC 7919, DOI 10.17487/RFC7919, August 2016, <https://www.rfc-editor.org/info/rfc7919>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8422] Nir, Y., Josefsson, S., and M. Pegourie-Gonnard, "Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS) Versions 1.2 and Earlier", RFC 8422, DOI 10.17487/RFC8422, August 2018, <https://www.rfc-editor.org/info/rfc8422>.

7.2. Informative References

[BLEI] Bleichenbacher, D., "Chosen Ciphertext Attacks against Protocols Based on RSA Encryption Standard PKCS #1", Advances in Cryptology -- CRYPTO'98, LNCS vol. 1462, pages: 1-12 , 1998.

[deprecate-ffdh] Bartle, C., Aviram, N., and F. Valsorda, "Deprecating FFDH Ciphersuites in TLS", June 2021, <https://datatracker.ietf.org/doc/draft-bartle-tls-deprecate-ffdhe/>.

[DLOG795] Boudot, F., Gaudry, P., Guillevic, A., Heninger, N., Thomé, E., and P. Zimmermann, "Comparing the difficulty of factorization and discrete logarithm: a 240-digit experiment", 17 August 2020, <https://eprint.iacr.org/2020/697>.

[DROWN] Aviram, N., Schinzel, S., Somorovsky, J., Heninger, N., Dankel, M., Steube, J., Valenta, L., Adrian, D., Halderman, J.A., Dukhovni, V., Käsper, E., Cohney, S., Engels, S., Paar, C., and Y. Shavitt, "DROWN: Breaking TLS using SSLv2", August 2016, <https://drownattack.com/drown-attack-paper.pdf>.

[NEW-BLEI] Meyer, C., Somorovsky, J., Weiss, E., Schwenk, J., Schinzel, S., and E. Tews, "Revisiting SSL/TLS Implementations: New Bleichenbacher Side Channels and Attacks", August 2014, <https://www.usenix.org/system/files/conference/usenixsecurity14/sec14-paper-meyer.pdf>.

[Raccoon] Merget, R., Brinkmann, M., Aviram, N., Somorovsky, J., Mittmann, J., and J. Schwenk, "Raccoon Attack: Finding and Exploiting Most-Significant-Bit-Oracles in TLS-DH(E)", 9 September 2020, <https://raccoon-attack.com/RacoonAttack.pdf>.

[ROBOT] Boeck, H., Somorovsky, J., and C. Young, "Return Of Bleichenbacher's Oracle Threat (ROBOT)", 27th USENIX Security Symposium , 2018.

[SC-tls-des-idea-ciphers-to-historic] "Moving single-DES and IDEA TLS ciphersuites to Historic", 25 January 2021, <https://datatracker.ietf.org/doc/status-change-tls-des-idea-ciphers-to-historic/>.

[server_side_tls] King, A., "Server Side TLS", July 2020, <https://wiki.mozilla.org/Security/Server_Side_TLS>.

[subgroups] Valenta, L., Adrian, D., Sanso, A., Cohney, S., Fried, J., Hastings, M., Halderman, J.A., and N. Heninger, "Measuring small subgroup attacks against Diffie-Hellman", 15 October 2016, <https://eprint.iacr.org/2016/995/20161017:193515>.

[weak-dh] Adrian, D., Bhargavan, K., Durumeric, Z., Gaudry, P., Green, M., Halderman, J.A., Heninger, N., Springall, D., Thomé, E., Valenta, L., VanderSloot, B., Wustrow, E., Zanella-Béguelin, S., and P. Zimmermann, "Weak Diffie-Hellman and the Logjam Attack", October 2015, <https://weakdh.org/>.

[XPROT] Jager, T., Schwenk, J., and J. Somorovsky, "On the Security of TLS 1.3 and QUIC Against Weaknesses in PKCS#1 v1.5 Encryption", Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security , 2015.

Author's Address

Nimrod Aviram

Email: nimrod.aviram@gmail.com