Stream: Independent Submission
RFC: 9151
Category: Informational
Published: April 2022
ISSN: 2070-1721
Author: D. Cooley
NSA

RFC 9151
Commercial National Security Algorithm (CNSA) Suite Profile for TLS and DTLS 1.2 and 1.3

Abstract

This document defines a base profile for TLS protocol versions 1.2 and 1.3 as well as DTLS protocol versions 1.2 and 1.3 for use with the US Commercial National Security Algorithm (CNSA) Suite.

The profile applies to the capabilities, configuration, and operation of all components of US National Security Systems that use TLS or DTLS. It is also appropriate for all other US Government systems that process high-value information.

The profile is made publicly available here for use by developers and operators of these and any other system deployments.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This is a contribution to the RFC Series, independently of any other RFC stream. The RFC Editor has chosen to publish this document at its discretion and makes no statement about its value for implementation or deployment. Documents approved for publication by the RFC Editor are not candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9151.

Copyright Notice

Table of Contents

1. Introduction

This document specifies a profile of TLS version 1.2 [RFC5246] and TLS version 1.3 [RFC8446] as well as DTLS version 1.2 [RFC6347] and DTLS version 1.3 [RFC9147] for use by applications that support the National Security Agency's (NSA) Commercial National Security Algorithm (CNSA) Suite [CNSA]. The profile applies to the capabilities, configuration, and operation of all components of US National Security Systems [SP80059]. It is also appropriate for all other US Government systems that process high-value information. It is made publicly available for use by developers and operators of these and any other system deployments.

This document does not define any new cipher suites; instead, it defines a CNSA-compliant profile of TLS and DTLS, and the cipher suites defined in [RFC5288], [RFC5289], and [RFC8446]. This profile uses only algorithms in the CNSA Suite.

The reader is assumed to have familiarity with the TLS 1.2 and 1.3 as well as the DTLS 1.2 and 1.3 protocol specifications: [RFC5246], [RFC8446], [RFC6347], and [RFC9147], respectively. All **MUST**-level requirements from the protocol documents apply throughout this profile; they are generally not repeated. This profile contains changes that elevate some **SHOULD**-level options to **MUST**-level; this profile also contains changes that elevate some **MAY**-level options to **SHOULD**-level or **MUST**-level. All options that are not mentioned in this profile remain at their original requirement level.

2. CNSA

The National Security Agency (NSA) profiles commercial cryptographic algorithms and protocols as part of its mission to support secure, interoperable communications for US National Security Systems. To this end, it publishes guidance both to assist with the US Government transition to new algorithms and to provide vendors -- and the Internet community in general -- with information concerning their proper use and configuration.

Recently, cryptographic transition plans have become overshadowed by the prospect of the development of a cryptographically relevant quantum computer. The NSA has established the CNSA Suite to provide vendors and IT users near-term flexibility in meeting their Information

Assurance (IA) interoperability requirements. The purpose behind this flexibility is to avoid having vendors and customers make two major transitions in a relatively short timeframe, as we anticipate a need to shift to quantum-resistant cryptography in the near future.

The NSA is authoring a set of RFCs, including this one, to provide updated guidance concerning the use of certain commonly available commercial algorithms in IETF protocols. These RFCs can be used in conjunction with other RFCs and cryptographic guidance (e.g., NIST Special Publications) to properly protect Internet traffic and data-at-rest for US National Security Systems.

3. Terminology

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.

"ECDSA" and "ECDH" refer to the use of the Elliptic Curve Digital Signature Algorithm (ECDSA) and Elliptic Curve Diffie Hellman (ECDH), respectively. ECDSA and ECDH are used with the NIST P-384 curve (which is based on a 384-bit prime modulus) and the SHA-384 hash function. Similarly, "RSA" and "DH" refer to Rivest-Shamir-Adleman (RSA) and Finite Field Diffie-Hellman (DH), respectively. RSA and DH are used with a 3072-bit or 4096-bit modulus. When RSA is used for digital signature, it is used with the SHA-384 hash function.

Henceforth, this document refers to TLS versions 1.2 and 1.3 and DTLS versions 1.2 and 1.3 collectively as "(D)TLS".

4. CNSA Suites

[CNSA] approves the use of both Finite Field and elliptic curve versions of the DH key agreement algorithm as well as RSA-based key establishment. [CNSA] also approves certain versions of the RSA and elliptic curve digital signature algorithms. The approved encryption techniques include the Advanced Encryption Standard (AES) used with a 256-bit key in an Authenticated Encryption with Associated Data (AEAD) mode.

In particular, CNSA includes the following:

Encryption:
  AES [AES] (with key size 256 bits), operating in Galois/Counter Mode (GCM) [GCM]

Digital Signature:
  ECDSA [DSS] (using the NIST P-384 elliptic curve)

  RSA [DSS] (with a modulus of 3072 bits or 4096 bits)

Key Establishment (includes key agreement and key transport):
  ECDH [PWKE-A] (using the NIST P-384 elliptic curve)

DH [PWKE-A] (with a prime modulus of 3072 or 4096 bits)

RSA [PWKE-B] (with a modulus of 3072 or 4096 bits, but only in (D)TLS 1.2)

[CNSA] also approves the use of SHA-384 [SHS] as the hash algorithm for mask generation, signature generation, Pseudorandom Function (PRF) in TLS 1.2 and HMAC-based Key Derivation Function (HKDF) in TLS 1.3.

4.1. CNSA (D)TLS Key Establishment Algorithms

The following combination of algorithms and key sizes are used in CNSA (D)TLS:

AES with 256-bit key, operating in GCM mode

ECDH [PWKE-A] using the Ephemeral Unified Model Scheme with cofactor set to 1 (see Section 6.1.2.2 in [PWKE-A])

TLS PRF/HKDF with SHA-384 [SHS]

Or

AES with 256-bit key, operating in GCM mode

RSA key transport using 3072-bit or 4096-bit modulus [PWKE-B][RFC8017]

TLS PRF/HKDF with SHA-384 [SHS]

Or

AES with 256-bit key, operating in GCM mode

DH using dhEphem with domain parameters specified below in Section 5.3 (see Section 6.1.2.1 in [PWKE-A])

TLS PRF/HKDF with SHA-384 [SHS]

The specific CNSA-compliant cipher suites are listed in Section 5.

4.2. CNSA TLS Authentication

For server and/or client authentication, CNSA (D)TLS **MUST** generate and verify either ECDSA signatures or RSA signatures.

In all cases, the client **MUST** authenticate the server. The server **MAY** also authenticate the client, as needed by the specific application.

The public keys used to verify these signatures **MUST** be contained in a certificate (see Section 5.4 for more information).

5. CNSA Compliance and Interoperability Requirements

CNSA (D)TLS **MUST NOT** use TLS versions prior to (D)TLS 1.2 in a CNSA-compliant system. CNSA (D)TLS servers and clients **MUST** implement and use either (D)TLS version 1.2 [RFC5246] [RFC6347] or (D)TLS version 1.3 [RFC8446] [RFC9147].

5.1. Acceptable Elliptic Curve Cryptography (ECC) Curves

The elliptic curves used in the CNSA Suite appear in the literature under two different names [DSS] [SECG]. For the sake of clarity, both names are listed below:

Curve	NIST name	SECG name
P-384	nistp384	secp384r1

Table 1: Elliptic Curves in CNSA Suite

[RFC8422] defines a variety of elliptic curves. CNSA (D)TLS connections **MUST** use secp384r1 (also called nistp384), and the uncompressed form **MUST** be used, as required by [RFC8422] and [RFC8446].

Key pairs **MUST** be generated following Section 5.6.1.2 of [PWKE-A].

5.2. Acceptable RSA Schemes, Parameters, and Checks

[CNSA] specifies a minimum modulus size of 3072 bits; however, only two modulus sizes (3072 bits and 4096 bits) are supported by this profile.

For (D)TLS 1.2:

> For certificate signatures, RSASSA-PKCS1-v1_5 [RFC8017] **MUST** be supported, and RSASSA-PSS [DSS] **SHOULD** be supported.
>
> For signatures in TLS handshake messages, RSASSA-PKCS1-v1_5 [RFC8017] **MUST** be supported, and RSASSA-PSS [DSS] **SHOULD** be supported.
>
> For key transport, RSAES-PKCS1-v1_5 [RFC8017] **MUST** be supported.

For (D)TLS 1.3:

> For certificate signatures, RSASSA-PKCS1-v1_5 [RFC8017] **MUST** be supported, and RSASSA-PSS [DSS] **SHOULD** be supported.
>
> For signatures in TLS handshake messages, RSASSA-PSS [DSS] **MUST** be supported.
>
> For key transport, TLS 1.3 does not support RSA key transport.

For all versions of (D)TLS:

> RSA exponent e **MUST** satisfy $2^{16}<e<2^{256}$ and be odd per [DSS].

If RSASSA-PSS is supported (using rsa_pss_rsae_sha384 for example), then the implementation **MUST** assert rsaEncryption as the public key algorithm, the hash algorithm (used for both mask generation and signature generation) **MUST** be SHA-384, the mask generation function 1 (MGF1) from [RFC8017] **MUST** be used, and the salt length **MUST** be 48 octets.

5.3. Acceptable Finite Field Groups

[CNSA] specifies a minimum modulus size of 3072 bits; however, only two modulus sizes (3072 bits and 4096 bits) are supported by this profile.

Ephemeral key pairs **MUST** be generated following Section 5.6.1.1.1 of [PWKE-A] using the approved safe prime groups specified in [RFC7919] for DH ephemeral key agreement. The named groups are:

ffdhe3072 (ID=257)

ffdhe4096 (ID=258)

5.4. Certificates

Certificates used to establish a CNSA (D)TLS connection **MUST** be signed with ECDSA or RSA and **MUST** be compliant with the CNSA Suite Certificate and Certificate Revocation List (CRL) Profile [RFC8603].

6. (D)TLS 1.2 Requirements

Although TLS 1.2 has technically been obsoleted by the IETF in favor of TLS 1.3, many implementations and deployments of TLS 1.2 will continue to exist. For the cases where TLS 1.2 continues to be used, implementations **MUST** use [RFC5246] and **SHOULD** implement the updates specified in [RFC8446] (outlined in Section 1.3 of that document).

The CNSA (D)TLS 1.2 client **MUST** offer at least one of these cipher suites:

TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 [RFC5289] [RFC8422]

TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 [RFC5289] [RFC8422]

TLS_RSA_WITH_AES_256_GCM_SHA384 [RFC5288]

TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 [RFC5288] [RFC7919]

The CNSA cipher suites listed above **MUST** be the first (most preferred) cipher suites in the ClientHello message.

A CNSA (D)TLS client that offers interoperability with servers that are not CNSA compliant **MAY** offer additional cipher suites, but any additional cipher suites **MUST** appear after the CNSA cipher suites in the ClientHello message.

A CNSA (D)TLS server **MUST** accept one of the CNSA Suites above if they are offered in the ClientHello message before accepting a non-CNSA-compliant suite.

If interoperability is not desired with non-CNSA-compliant clients or servers, then the session **MUST** fail if no CNSA Suites are offered or accepted.

6.1. The "extended_master_secret" Extension

A CNSA (D)TLS client **SHOULD** include and a CNSA (D)TLS server **SHOULD** accept the "extended_master_secret" extension as specified in [RFC7627]. See Section 1 of [RFC7627] for security concerns when this extension is not used.

6.2. The "signature_algorithms" Extension

A CNSA (D)TLS client **MUST** include and a CNSA (D)TLS server **MUST** also accept the "signature_algorithms" extension. The CNSA (D)TLS client **MUST** offer and the CNSA (D)TLS server **MUST** also accept at least one of the following values in the "signature_algorithms" extensions as specified in [RFC8446]:

ecdsa_secp384r1_sha384

rsa_pkcs1_sha384

And, if supported, the client **SHOULD** offer and/or the server **SHOULD** also accept:

rsa_pss_pss_sha384

rsa_pss_rsae_sha384

Following the guidance in [RFC8603], CNSA (D)TLS servers **MUST** only accept ECDSA or RSA for signatures on ServerKeyExchange messages and for certification path validation.

Other client offerings **MAY** be included to indicate the acceptable signature algorithms in cipher suites that are offered for interoperability with servers not compliant with CNSA and to indicate the signature algorithms that are acceptable for ServerKeyExchange messages and for certification path validation in non-compliant CNSA (D)TLS connections. These offerings **MUST NOT** be accepted by a CNSA-compliant (D)TLS server.

6.3. The "signature_algorithms_cert" Extension

A CNSA (D)TLS client **MAY** include the "signature_algorithms_cert" extension. CNSA (D)TLS servers **MUST** process the "signature_algorithms_cert" extension if it is offered per Section 4.2.3 of [RFC8446].

Both CNSA (D)TLS clients and servers **MUST** use one of the following values for certificate path validation:

ecdsa_secp384r1_sha384

rsa_pkcs1_sha384

And, if supported, **SHOULD** offer/accept:

rsa_pss_pss_sha384

rsa_pss_rsae_sha384

6.4. The CertificateRequest Message

When a CNSA (D)TLS server is configured to authenticate the client, the server **MUST** include the following values in its CertificateRequest.supported_signature_algorithms [RFC5246] offer:

ecdsa_secp384r1_sha384

rsa_pkcs1_sha384

And, if supported as specified in [RFC8446], **SHOULD** offer/accept:

rsa_pss_pss_sha384

rsa_pss_rsae_sha384

6.5. The CertificateVerify Message

A CNSA (D)TLS client **MUST** use ECDSA or RSA when sending the CertificateVerify message. CNSA (D)TLS servers **MUST** only accept ECDSA or RSA in the CertificateVerify message.

6.6. The Signature in the ServerKeyExchange Message

A CNSA (D)TLS server **MUST** sign the ServerKeyExchange message using ECDSA or RSA.

6.7. Certificate Status

Certificate Authorities (CAs) providing certificates to a CNSA (D)TLS server or client **MUST** provide certificate revocation status information via a Certificate Revocation List (CRL) distribution point or using the Online Certificate Status Protocol (OCSP). A CNSA client **SHOULD** request it according to Section 4.4.2.1 of [RFC8446]. If OCSP is supported, the (D)TLS server **SHOULD** provide OCSP responses in the CertificateStatus message.

7. (D)TLS 1.3 Requirements

The CNSA (D)TLS client **MUST** offer the following cipher suite in the ClientHello:

TLS_AES_256_GCM_SHA384

The CNSA (D)TLS client **MUST** include at least one of the following values in the "supported_groups" extension:

ECDHE: secp384r1

DHE: ffdhe3072

DHE: ffdhe4096

The CNSA cipher suite **MUST** be the first (most preferred) cipher suite in the ClientHello message and in the extensions.

A CNSA (D)TLS client that offers interoperability with servers that are not CNSA compliant **MAY** offer additional cipher suites, but any additional cipher suites **MUST** appear after the CNSA-compliant cipher suites in the ClientHello message.

A CNSA (D)TLS server **MUST** accept one of the CNSA algorithms listed above if they are offered in the ClientHello message.

If interoperability is not desired with non-CNSA-compliant clients or servers, then the session **MUST** fail if no CNSA Suites are offered or accepted.

7.1. The "signature_algorithms" Extension

A CNSA (D)TLS client **MUST** include the "signature_algorithms" extension. The CNSA (D)TLS client **MUST** offer at least one of the following values in the "signature_algorithms" extension:

ecdsa_secp384r1_sha384

rsa_pss_pss_sha384

rsa_pss_rsae_sha384

Clients that allow negotiating TLS 1.2 **MAY** offer rsa_pkcs1_sha384 for use with TLS 1.2. Other offerings **MAY** be included to indicate the acceptable signature algorithms in cipher suites that are offered for interoperability with servers not compliant with CNSA in non-compliant CNSA (D)TLS connections. These offerings **MUST NOT** be accepted by a CNSA-compliant (D)TLS server.

7.2. The "signature_algorithms_cert" Extension

A CNSA (D)TLS client **SHOULD** include the "signature_algorithms_cert" extension. And, if offered, the CNSA (D)TLS client **MUST** offer at least one of the following values in the "signature_algorithms_cert" extension:

ecdsa_secp384r1_sha384

rsa_pkcs1_sha384

And, if supported, **SHOULD** offer:

rsa_pss_pss_sha384

rsa_pss_rsae_sha384

Following the guidance in [RFC8603], CNSA (D)TLS servers **MUST** only accept ECDSA or RSA for certificate path validation.

Other offerings **MAY** be included to indicate the signature algorithms that are acceptable for certification path validation in non-compliant CNSA (D)TLS connections. These offerings **MUST NOT** be accepted by a CNSA-compliant (D)TLS server.

7.3. The "early_data" Extension

A CNSA (D)TLS client or server **MUST NOT** include the "early_data" extension. See Section 2.3 of [RFC8446] for security concerns.

7.4. Resumption

A CNSA (D)TLS server **MAY** send a CNSA (D)TLS client a NewSessionTicket message to enable resumption. A CNSA (D)TLS client **MUST** request "psk_dhe_ke" via the "psk_key_exchange_modes" ClientHello extension to resume a session. A CNSA (D)TLS client **MUST** offer Ephemeral Elliptic Curve Diffie-Hellman (ECDHE) with SHA-384 and/or Ephemeral Diffie-Hellman (DHE) with SHA-384 in the "supported_groups" and/or "key_share" extensions.

7.5. Certificate Status

Certificate Authorities (CAs) providing certificates to a CNSA (D)TLS server or client **MUST** provide certificate revocation status information via a Certificate Revocation List (CRL) distribution point or using the Online Certificate Status Protocol (OCSP). A CNSA client **SHOULD** request it according to Section 4.4.2.1 of [RFC8446]. If OCSP is supported, the (D)TLS server **SHOULD** provide OCSP responses in the "CertificateEntry".

8. Security Considerations

Most of the security considerations for this document are described in [RFC5246], [RFC8446], [RFC6347], and [RFC9147]. In addition, the security considerations for Elliptic Curve Cryptography (ECC) related to TLS are described in [RFC8422], [RFC5288], and [RFC5289]. Readers should consult those documents.

In order to meet the goal of a consistent security level for the entire cipher suite, CNSA (D)TLS implementations **MUST** only use the elliptic curves, RSA schemes, and Finite Fields defined in Section 5.1, Section 5.2, and Section 5.3, respectively. If this is not the case, then security may be weaker than is required.

As noted in TLS version 1.3 [RFC8446], TLS does not provide inherent replay protections for early data. For this reason, this profile forbids the use of early data.

9. IANA Considerations

This document has no IANA actions.

10. References

10.1. Normative References

[AES]

National Institute of Standards and Technology, "Announcing the ADVANCED ENCRYPTION STANDARD (AES)", FIPS 197, DOI 10.6028/NIST.FIPS.197, November 2001, <https://nvlpubs.nist.gov/nistpubs/fips/NIST.FIPS.197.pdf>.

[CNSA] Committee for National Security Systems, "Use of Public Standards for Secure Information Sharing", CNSSP 15, October 2016, <https://www.cnss.gov/CNSS/issuances/Policies.cfm>.

[DSS] National Institute of Standards and Technology, "Digital Signature Standard (DSS)", FIPS PUB 186-4, DOI 10.6028/NIST.FIPS.186-4, July 2013, <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf>.

[GCM] Dworkin, M., "Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC", NIST Special Publication 800-38D, DOI 10.6028/NIST.SP.800-38D, November 2007, <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf>.

[PWKE-A] Barker, E., Chen, L., Roginsky, A., Vassilev, A., and R. Davis, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography", NIST Special Publication 800-56A Revision 3, DOI 10.6028/NIST.SP.800-56Ar3, April 2018, <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar3.pdf>.

[PWKE-B] Barker, E., Chen, L., Roginsky, A., Vassilev, A., Davis, R., and S. Simon, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography", NIST Special Publication 800-56B Revision 2, DOI 10.6028/NIST.SP.800-56Br2, March 2019, <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Br2.pdf>.

[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>.

[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>.

[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>.

[RFC7627] Bhargavan, K., Ed., Delignat-Lavaud, A., Pironti, A., Langley, A., and M. Ray, "Transport Layer Security (TLS) Session Hash and Extended Master Secret Extension", RFC 7627, DOI 10.17487/RFC7627, September 2015, <https://www.rfc-editor.org/info/rfc7627>.

[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>.

[RFC8017] Moriarty, K., Ed., Kaliski, B., Jonsson, J., and A. Rusch, "PKCS #1: RSA Cryptography Specifications Version 2.2", RFC 8017, DOI 10.17487/RFC8017, November 2016, <https://www.rfc-editor.org/info/rfc8017>.

[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>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8603] Jenkins, M. and L. Zieglar, "Commercial National Security Algorithm (CNSA) Suite Certificate and Certificate Revocation List (CRL) Profile", RFC 8603, DOI 10.17487/RFC8603, May 2019, <https://www.rfc-editor.org/info/rfc8603>.

[RFC9147] Rescorla, E., Tschofenig, H., and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", RFC 9147, DOI 10.17487/RFC9147, April 2022, <https://www.rfc-editor.org/info/rfc9147>.

[SHS] National Institute of Standards and Technology (NIST), "Secure Hash Standard (SHS)", DOI 10.6028/NIST.FIPS.180-4, FIPS PUB 180-4, August 2015, <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf>.

10.2. Informative References

[SECG] Brown, D., "SEC 2: Recommended Elliptic Curve Domain Parameters", Version 2.0, February 2010, <https://www.secg.org/sec2-v2.pdf>.

[SP80059] Barker, W., "Guideline for Identifying an Information System as a National Security System", DOI 10.6028/NIST.SP.800-59, NIST Special Publication 800-59, August 2003, <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-59.pdf>.

Author's Address

Dorothy Cooley
National Security Agency
Email: decoole@nsa.gov