

TLS  
Internet-Draft  
Intended status: Standards Track  
Expires: 22 May 2026

T. Reddy  
Nokia  
T. Hollebeek  
DigiCert  
J. Gray  
Entrust  
S. Fluhrer  
Cisco Systems  
18 November 2025

Use of SLH-DSA in TLS 1.3  
draft-reddy-tls-slhdsa-02

## Abstract

This memo specifies how the post-quantum signature scheme SLH-DSA [FIPS205] is used for authentication in TLS 1.3.

## 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 22 May 2026.

## Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. Conventions and Terminology . . . . .	2
2. Applicability of SLH-DSA . . . . .	3
3. SLH-DSA SignatureSchemes Types . . . . .	4
4. SLH-DSA Variant Selection Guidance . . . . .	7
5. Security Considerations . . . . .	7
5.1. Key Lifetime Management . . . . .	7
6. IANA Considerations . . . . .	8
7. References . . . . .	8
7.1. Normative References . . . . .	8
7.2. Informative References . . . . .	9
Acknowledgments . . . . .	10
Authors' Addresses . . . . .	11

## 1. Introduction

Stateless Hash-Based Digital Signatures (SLH-DSA) [FIPS205] is a quantum-resistant digital signature scheme standardized by the US National Institute of Standards and Technology (NIST) PQC project.

This memo specifies how SLH-DSA can be negotiated for authentication in TLS 1.3 via the "signature\_algorithms" and "signature\_algorithms\_cert" extensions.

### 1.1. Conventions and 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. These words may also appear in this document in lower case as plain English words, absent their normative meanings.

This document uses terms defined in [I-D.ietf-pquip-pqt-hybrid-terminology]. For the purposes of this document, it is helpful to be able to divide cryptographic algorithms into two classes:

"Asymmetric Traditional Cryptographic Algorithm": An asymmetric cryptographic algorithm based on integer factorisation, finite field discrete logarithms or elliptic curve discrete logarithms, elliptic curve discrete logarithms, or related mathematical problems.

"Post-Quantum Algorithm": An asymmetric cryptographic algorithm that is believed to be secure against attacks using quantum computers as well as classical computers. Post-quantum algorithms can also be called quantum-resistant or quantum-safe algorithms. Examples of quantum-resistant digital signature schemes include ML-DSA and SLH-DSA.

## 2. Applicability of SLH-DSA

Applications that use SLH-DSA need to be aware that the signature sizes of the algorithms specified in this document are generally large. SLH-DSA offers three security levels: 1, 3, and 5, and two parameter variants for each level:

- \* Small (s) variant, which are optimized for minimal signature size, have signature sizes ranging from 7856 bytes (128-bit) to 29792 bytes (256-bit).
- \* Fast (f) variant, optimized for faster key generation and signing, have signature sizes ranging from 17088 bytes (128-bit) to 29792 bytes (256-bit). However, they are slower at signature verification.

Despite offering trade-offs between size and performance, all SLH-DSA variants produce significantly larger signatures than traditional signature algorithms. While SLH-DSA increases the size of the TLS 1.3 handshake, its impact on connection performance is minimal in the context of large data transfers, especially over low-loss networks. For instance, TLS-based protocols are increasingly used to secure long-lived interfaces in critical infrastructure, such as telecommunication networks. In particular, DTLS-in-SCTP has been mandated in 3GPP for interfaces such as N2 that use long-lived TLS connections.

In deployments aiming to minimize handshake size, SLH-DSA may still be adopted for signing X.509 certificates while avoiding its use in the CertificateVerify message, which involves generating a signature over the entire TLS handshake transcript. This helps avoid

performance concerns related to large signatures or expensive verification. SLH-DSA is well suited for enhancing the post-quantum security of root and intermediate certificates without affecting TLS handshake performance.

Mechanisms such as Abridged TLS Certificate Chains [I-D.ietf-tls-cert-abridge] and Suppressing CA Certificates [I-D.kampanakis-tls-scas-latest] reduce handshake size by limiting certificate exchange to only end-entity certificates. In such cases, intermediate certificates are assumed to be known to the peer, allowing the use of larger signature algorithms like SLH-DSA for those certificates without adding overhead to the handshake.

### 3. SLH-DSA SignatureSchemes Types

SLH-DSA [FIPS205] utilizes the concept of stateless hash-based signatures. In contrast to stateful signature algorithms, SLH-DSA eliminates the need for maintaining state information during the signing process. SLH-DSA is designed to sign up to  $2^{64}$  messages and it offers three security levels. The parameters for security levels 1, 3, and 5 were chosen to provide the equivalent of AES-128, AES-192, and AES-256 level of security respectively (see Table 2 in Section 10 of [I-D.ietf-pquip-pqc-engineers]).

This document specifies the use of the SLH-DSA algorithm in TLS at three security levels. Each security level (1, 3, and 5) defines two variants of the algorithm: a small (S) version and a fast (F) version. The small version prioritizes smaller signature sizes, making them suitable for resource-constrained environments IoT devices. Conversely, the fast version prioritizes speed over signature size, minimizing the time required to generate signatures. However, signature verification with the small version is faster than with the fast version. For hash function selection, the algorithm uses SHA-256 ([FIPS180]) for security level 1 and both SHA-256 and SHA-512 ([FIPS180]) for security levels 3 and 5. Alternatively, SHAKE256 ([FIPS202]) can be used across all security levels.

The following combinations are defined in SLH-DSA [FIPS205]:

- \* SLH-DSA-128S-SHA2
- \* SLH-DSA-128F-SHA2
- \* SLH-DSA-192S-SHA2
- \* SLH-DSA-192F-SHA2
- \* SLH-DSA-256S-SHA2

- \* SLH-DSA-256F-SHA2
- \* SLH-DSA-128S-SHAKE
- \* SLH-DSA-128F-SHAKE
- \* SLH-DSA-192S-SHAKE
- \* SLH-DSA-192F-SHAKE
- \* SLH-DSA-256S-SHAKE
- \* SLH-DSA-256F-SHAKE

SLH-DSA does not introduce any new hardness assumptions beyond those inherent to its underlying hash functions. It builds upon established foundations in cryptography, making it a reliable and robust digital signature scheme for a post-quantum world. While attacks on lattice-based schemes like ML-DSA are currently hypothetical at the time of writing this document, such attacks, if realized, could compromise their security. SLH-DSA would remain unaffected by these attacks due to its distinct mathematical foundations. This ensures the ongoing security of systems and protocols that use SLH-DSA for digital signatures.

However, ML-DSA outperforms SLH-DSA in both signature generation and validation time, as well as in signature size, making it a recommended choice for end-entity certificates. SLH-DSA, in contrast, offers smaller key sizes but larger signature sizes. Given its well-established hardness assumption, SLH-DSA may be preferred for TLS applications where high confidence in security is a priority, such as for long-lived TLS sessions and deployments where computational costs of signature generation and validation are minor compared to data transmission and processing demands of user data. The findings in [PQ-TLS-TTLB] shows that while PQ algorithms increase the TLS 1.3 handshake data size, their effect on connection performance is minimal for large data transfers, especially in low-loss networks. Additionally, SLH-DSA is suitable for use in CA certificates due to its strong cryptographic assurances and smaller key sizes. Its robustness against emerging quantum attacks makes it a dependable choice for trust anchors and long-term security, even though it has larger signature sizes.

As defined in [RFC8446], the SignatureScheme namespace is used for the negotiation of signature scheme for authentication via the "signature\_algorithms" and "signature\_algorithms\_cert" extensions. This document adds new SignatureSchemes types for the SLH-DSA as follows.

```
enum {  
    slhdsa_sha2_128s (0x0911),  
    slhdsa_sha2_128f (0x0912),  
    slhdsa_sha2_192s (0x0913),  
    slhdsa_sha2_192f (0x0914),  
    slhdsa_sha2_256s (0x0915),  
    slhdsa_sha2_256f (0x0916),  
    slhdsa_shake_128s (0x0917),  
    slhdsa_shake_128f (0x0918),  
    slhdsa_shake_192s (0x0919),  
    slhdsa_shake_192f (0x091A),  
    slhdsa_shake_256s (0x091B),  
    slhdsa_shake_256f (0x091C)  
} SignatureScheme;
```

It is important to note that the slhdsa\* entries represent the pure versions of these algorithms and should not be confused with prehashed variant HashSLH-DSA, also defined in [FIPS205].

SLH-DSA supports two signing modes: deterministic and hedged. In the deterministic mode, the signature is derived solely from the message and the private key, without requiring fresh randomness at signing time, this eliminates dependence on an external random number generator (RNG). It instead uses a precomputed randomness embedded in the private key. The hedged mode incorporates both fresh randomness generated at signing time, as well as the precomputed randomness, thereby offering somewhat stronger assurances. In the context of TLS, authentication signatures are computed over unique handshake transcripts, making each signature input distinct for every session. This property allows the use of either signing mode. The choice between deterministic and hedged modes does not affect interoperability, as the verification process is the same for both. In both modes, the context parameter defined in Algorithm 22 and Algorithm 24 of [FIPS205] MUST be set to the empty string.

The signature MUST be computed and verified as specified in Section 4.4.3 of [RFC8446].

The corresponding end-entity certificate when negotiated MUST use id-slh-dsa-sha2-128s, id-slh-dsa-sha2-128f, id-slh-dsa-sha2-192s, id-slh-dsa-sha2-192f, id-slh-dsa-sha2-256s, id-slh-dsa-sha2-256f, id-slh-dsa-shake-128s, id-slh-dsa-shake-128f, id-slh-dsa-shake-192s, id-slh-dsa-shake-192f, id-slh-dsa-shake-256s and id-slh-dsa-shake-256f respectively as defined in [I-D.ietf-lamps-x509-slhdsa]}.

The schemes defined in this document MUST NOT be used in TLS 1.2 [RFC5246]. A peer that receives ServerKeyExchange or CertificateVerify message in a TLS 1.2 connection with schemes defined in this document MUST abort the connection with an `illegal_parameter` alert.

#### 4. SLH-DSA Variant Selection Guidance

When deploying SLH-DSA in TLS 1.3, the choice of variant involves trade-offs among signing speed, verification cost, signature size, and the underlying hash function. The decision depends heavily on the characteristics and constraints of the target environment. If SHAKE is supported and offers acceptable performance, SHAKE-based variants are generally recommended for their performance and flexibility. In environments where SHAKE is unavailable or performs poorly, SHA-2 based variants offer comparable security and are a suitable alternative.

The choice between "fast" and "small" variants depends on whether signing or verification performance is more critical in the target environment. Fast variants provide significantly faster signing but incur higher verification costs. Conversely, small variants enable more efficient verification but have slower signing performance.

Security level requirements also guide the selection. If 128-bit security is sufficient, 128-bit variants can be used. For applications requiring higher assurance, 192-bit or 256-bit variants may be more appropriate.

#### 5. Security Considerations

The security considerations discussed in Section 9 of [I-D.ietf-lamps-x509-slhdsa] need to be taken into account.

SLH-DSA imposes an upper bound of  $2^{64}$  signatures per key. If a key pair were used to sign 10 billion messages per second, it would take over 58 years to sign  $2^{64}$  messages. While this limit is extremely large, it may need to be considered if a single SLH-DSA private key was shared between a huge number of TLS servers all making extremely frequent negotiations.

##### 5.1. Key Lifetime Management

In order to maintain cryptographic safety in high-scale environments, deployments MUST:

- \* Rotate SLH-DSA certificates and keys based on expected signature usage, ensuring ample margin from the  $2^{64}$  signature limit.

- \* Monitor the number of signatures generated per SLH-DSA private key to ensure it remains well below the  $2^{64}$  signature limit.

## 6. IANA Considerations

This document requests twelve new entries to the TLS Named Group (or Supported Group) registry, according to the procedures in Section 6 of Section 6 of [TLSIANA].

Value	Description	Recommended	Reference
0x0911	slhdsa_sha2_128s	N	This document.
0x0912	slhdsa_sha2_128f	N	This document.
0x0913	slhdsa_sha2_192s	N	This document.
0x0914	slhdsa_sha2_192f	N	This document.
0x0915	slhdsa_sha2_256s	N	This document.
0x0916	slhdsa_sha2_256f	N	This document.
0x0917	slhdsa_shake_128s	N	This document.
0x0918	slhdsa_shake_128f	N	This document.
0x0919	slhdsa_shake_192s	N	This document.
0x091A	slhdsa_shake_192f	N	This document.
0x091B	slhdsa_shake_256s	N	This document.
0x091C	slhdsa_shake_256f	N	This document.

Table 1

## 7. References

### 7.1. Normative References

- [I-D.ietf-lamps-x509-slhdsa]  
 Bashiri, K., Fluhrer, S., Gazdag, S., Van Geest, D., and  
 S. Kousidis, "Internet X.509 Public Key Infrastructure:  
 Algorithm Identifiers for SLH-DSA", Work in Progress,



Internet-Draft, draft-ietf-lamps-x509-slhdsa-09, 30 June 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-lamps-x509-slhdsa-09>>.

- [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/rfc/rfc2119>>.
- [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/rfc/rfc8174>>.
- [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/rfc/rfc8446>>.
- [TLSIANA] Salowey, J. A. and S. Turner, "IANA Registry Updates for TLS and DTLS", Work in Progress, Internet-Draft, draft-ietf-tls-rfc8447bis-15, 21 July 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-tls-rfc8447bis-15>>.

## 7.2. Informative References

- [FIPS180] "NIST, Secure Hash Standard (SHS), FIPS PUB 180-4, August 2015", <<https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf>>.
- [FIPS202] "NIST, SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, FIPS PUB 202, August 2015.", <<https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.202.pdf>>.
- [FIPS205] "FIPS 205: Stateless Hash-Based Digital Signature Standard", <<https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.205.pdf>>.
- [I-D.ietf-pquip-pqc-engineers]  
Banerjee, A., Reddy, K. T., Schoiniakakis, D., Hollebeek, T., and M. Ounsworth, "Post-Quantum Cryptography for Engineers", Work in Progress, Internet-Draft, draft-ietf-pquip-pqc-engineers-14, 25 August 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-pquip-pqc-engineers-14>>.

[I-D.ietf-pquip-pqt-hybrid-terminology]

D, F., P, M., and B. Hale, "Terminology for Post-Quantum Traditional Hybrid Schemes", Work in Progress, Internet-Draft, draft-ietf-pquip-pqt-hybrid-terminology-06, 10 January 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-pquip-pqt-hybrid-terminology-06>>.

[I-D.ietf-tls-cert-abridge]

Jackson, D., "Abridged Compression for WebPKI Certificates", Work in Progress, Internet-Draft, draft-ietf-tls-cert-abridge-02, 16 September 2024, <<https://datatracker.ietf.org/doc/html/draft-ietf-tls-cert-abridge-02>>.

[I-D.kampanakis-tls-scas-latest]

Kampanakis, P., Bytheway, C., Westerbaan, B., and M. Thomson, "Suppressing CA Certificates in TLS 1.3", Work in Progress, Internet-Draft, draft-kampanakis-tls-scas-latest-03, 5 January 2023, <<https://datatracker.ietf.org/doc/html/draft-kampanakis-tls-scas-latest-03>>.

[PQ-TLS-TTLB]

"The impact of data-heavy, post-quantum TLS 1.3 on the time-to-last-byte of real-world connections.", <<https://www.amazon.science/publications/the-impact-of-data-heavy-post-quantum-tls-1-3-on-the-time-to-last-byte-of-real-world-connections>>.

[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/rfc/rfc5246>>.

[RFC8555] Barnes, R., Hoffman-Andrews, J., McCarney, D., and J. Kasten, "Automatic Certificate Management Environment (ACME)", RFC 8555, DOI 10.17487/RFC8555, March 2019, <<https://www.rfc-editor.org/rfc/rfc8555>>.

[RFC9261] Sullivan, N., "Exported Authenticators in TLS", RFC 9261, DOI 10.17487/RFC9261, July 2022, <<https://www.rfc-editor.org/rfc/rfc9261>>.

## Acknowledgments

Thanks to Bas Westerbaan, John Mattsson, D.J. Bernstein, Alicja Kario, and Peter Campbell for the discussion and comments.

Authors' Addresses

Tirumaleswar Reddy  
Nokia  
Bangalore  
Karnataka  
India  
Email: kondtir@gmail.com

Timothy Hollebeek  
DigiCert  
Pittsburgh,  
United States of America  
Email: tim.hollebeek@digicert.com

John Gray  
Entrust Limited  
2500 Solandt Road Suite 100  
Ottawa, Ontario K2K 3G5  
Canada  
Email: john.gray@entrust.com

Scott Fluhrer  
Cisco Systems  
Email: sfluhrer@cisco.com