sshm
Internet-Draft
Updates: 6187 (if approved)
Intended status: Standards Track
Expires: 6 July 2026

R. Petrov
2 January 2026

X.509v3 ML-DSA Certificates for the Secure Shell (SSH) Protocol
draft-rpe-ssh-x509-mldsa-00

Abstract

This document describes the use of Module-Lattice-Based Digital Signature Algorithm (ML-DSA) in Internet X.509 version 3 Public Key Certificate in the Secure Shell protocol. Accordingly, the document updates RFC6187.

Document and implementation details

This note is to be removed before publishing as an RFC.

The datatracker status page of the draft is draft-rpe-ssh-x509-mldsa (https://datatracker.ietf.org/doc/draft-rpe-ssh-x509-mldsa).

The source of this document is located at I-D ssh-x509-mldsa (https://gitlab.com/secsh/pkixssh/-/blob/mldsa_demo/draft-rpe-ssh-x509-mldsa.xml). Implementation could be found at PKIX-SSHMLDSA-DEMO (https://gitlab.com/secsh/pkixssh/-/tree/mldsa_demo) branch.

Discussion of this document takes place on the Secure Shell Maintenance (sshm)" (https://datatracker.ietf.org/group/sshm/about) mailing list (mailto:ssh@ietf.org) which is archived here (https://mailarchive.ietf.org/arch/browse/ssh/).

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 6 July 2026.

Table of Contents

1. Introduction

Secure Shell (SSH) [RFC4251] is a secure remote-login protocol. It provides for an extensible variety of public key algorithms for identifying servers and users to one another.

The Module-Lattice-Based Digital Signature Algorithm (ML-DSA) is a post-quantum digital signature algorithm. It is one of NIST's Post-Quantum Cryptography (PQC) project results standardised in [FIPS-204]. Note ML-DSA was known as Dilithium but standardised ML-DSA and Dilithium are not compatible.

X.509 Version 3(x509v3) digital certificate format is specified in [RFC5280]. The use of ML-DSA in Public Key Infrastructure X.509 (PKIX) is specified in [RFC9881].

The Secure Shell (SSH) Transport Layer Protocol, see [RFC4253], describes how server is authenticated to the client. The meaning of SSH Public Key Algorithms is described in the same document, see [RFC4253], Section 6.6. Authentication of the client to the server is described in SSH Authentication Protocol, see [RFC4252].

In [RFC6187], Section 2 are described currently standardised X.509 V3 certificates used in SSH Public Key Algorithms. This document details the use of X.509 digital certificates with ML-DSA signature algorithm to be implemented by SSH and standardize the use of names *x509v3-mldsa-44*, *x509v3-mldsa-65*, and *x509v3-mldsa-87*.

2. Conventions Used in This Document

The descriptions of key and signature formats use the notation introduced in [RFC4251], Section 3 and the string data type from [RFC4251], Section 5.

2.1. Requirements Language

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.

3. Public Key Algorithms using X.509v3 Certificates with ML-DSA public key

The SSH Public Key Algorithms [RFC4253], Section 6.6 define the type, how the key or certificate is encoded, the signature and/or encryption algorithms, and their encoding.

For X.509 certificates is used following "Public Key Format", added here only for reference:

```
|  string  key-type
|  uint32  certificate-count
|  string  certificate[1..certificate-count]
|  uint32  ocsp-response-count
|  string  ocsp-response[0..ocsp-response-count]
```

For complete description of each item see [RFC6187], Section 2.1. In scope of this document is first(sender's) certificate from certificate list whose "subjectPublicKeyInfo" field is a ML-DSA public key in a certificate is specified in [RFC9881], Section 4. The respective algorithm identifiers are in listed in [RFC9881], Section 2. In this document ML-DSA certificate and X.509 certificate with ML-DSA public key are used interchangeably.

For ML-DSA Certificates key-type field uses prefix "x509v3-" followed by corresponding plain key algorithm. For more details about ML-DSA plain key algorithms see [I-D.rpe-ssh-mldsa]. Signatures are generated as for plain key algorithms i.e., according to the "Pure ML-DSA Signature Generation" procedure described in [FIPS-204] Algorithm 2 step 10(sign) and Algorithm 3 step 5(verify).

* The *x509v3-mldsa-44* key-type is used when algorithm identifier, in "subjectPublicKeyInfo" field, is *id-ml-dsa-44*. This means that public key is an octet string of size 1312 without ASN.1 wrapping.

  Corresponding plain key algorithm is *mldsa-44*. For both public key algorithms signature is generated and encoded in the same way:

  ```
  string  mldsa-44
  string  signature
  ```

  Here, *signature* is the 2420-octet signature produced in accordance with [FIPS-204] Algorithm 2.

* The *x509v3-mldsa-65* key-type is used when algorithm identifier, in "subjectPublicKeyInfo" field, is *id-ml-dsa-65*. This means that public key is an octet string of size 1952 without ASN.1 wrapping.

  Corresponding plain key algorithm is *mldsa-65*. For both public key algorithms signature is generated and encoded in the same way:

  ```
  string  mldsa-65
  string  signature
  ```

  Here, *signature* is the 3309-octet signature produced in accordance with [FIPS-204] Algorithm 2.

* The *x509v3-mldsa-87* key-type is used when algorithm identifier, in "subjectPublicKeyInfo" field, is *id-ml-dsa-87*. This means that public key is an octet string of size 2592 without ASN.1 wrapping.

Corresponding plain key algorithm is *mldsa-87*. For both public key algorithms signature is generated and encoded in the same way:

```
string  mldsa-87
string  signature
```

Here, *signature* is the 4627-octet signature produced in accordance with [FIPS-204] Algorithm 2.

4. Certificate Extensions

Certificate extensions specify additional attributes associated with an X.509v3 Certificate, see [RFC5280], Section 4.2.

4.1. Key Usage

For ML-DSA Certificates *keyUsage* extension is defined in [RFC9881], Section 5. As is specified in [RFC6187], Section 2.2.1, certificate used in public key algorithms *digitalSignature* bit MUST be set. As well, This is applicable to the public key algorithms *x509v3-mldsa-44*, *x509v3-mldsa-65*, and *x509v3-mldsa-87* defined in this document.

4.2. Extended Key Usage

Paragraphs in [RFC6187], Section 2.2.2 define two SSH specific extension - *secureShellClient*, and *secureShellServer*. As stated in the section, in accordance with [RFC5280], Section 4.2.1.12, ML-DSA certificate MUST be used only for the indicated purposes too.

4.3. Subject Alternative Name

At end of chapter [RFC6187] (Section 4) is detailed recommendation for *subjectAlternativeName* X.509 certificate extension. These recommendation are relevant for ML-DSA certificate used in SSH as public key algorithm.

5. Usage

The use of X.509v3 Certificates SSH "Public Key Algorithms" is described in [RFC6187], Section 4. This is applicable to ML-DSA Certificates as well.

The ML-DSA digital signature algorithms correspond to the Table 1. defined in [FIPS-204] Section 4 "Parameter Sets". The table below match parameters sets to "NIST PQC Security Strength Category":

Parameters	NIST PQC Security Strength Category
ML-DSA-44	Category 2, NIST Level 2 (128-bit equivalent)
ML-DSA-65	Category 3, NIST Level 3 (192-bit equivalent)
ML-DSA-87	Category 5, NIST Level 2 (256-bit equivalent)

Table 1

Use of ML-DSA plain key algorithms is specified in [I-D.rpe-ssh-mldsa] and standard implementations of SSH SHOULD implement *mldsa-65* public Key algorithm. Implementation of ML-DSA Certificates MUST follow recommendation for plain-key algorithms. In addition certificate algorithm must be offered in preference to plain-key algorithm. This means that *x509v3-mldsa-NN* must precede *mldsa-NN*, where NN match number in parameter set. Also if ML-DSA Certificates are supported the public key algorithm *x509v3-mldsa-65* SHOULD implemented.

6. IANA Considerations

This document augments the Public Key Algorithm Names described in [RFC6187], Section 2.

This document requests new entries to "Public Key Algorithm Names" in the "Secure Shell (SSH) Protocol Parameters" registry [IANA-SSH] according to the procedures in [RFC9519], Section 3:

Public Key Algorithm Name	Reference
x509v3-mldsa-44	This document.
x509v3-mldsa-65	This document.
x509v3-mldsa-87	This document.

Table 2

7. Security Considerations

This documents inherits security considerations for public key algorithms used for user and for server authentication. For "user", see [RFC4252], Section 11 , and for "server" see [RFC4253], Section 14. The both documents refer to [RFC4251], Section 9 as full security considerations for SSH protocol.

For X.509v3 Certificates used in secure shell authentication are applicable the security considerations detailed in [RFC6187], Section 5. The security considerations for ML-DSA plain-keys, see [I-D.rpe-ssh-mldsa] applies to this specification as well. For ML-DSA Certificates applies as well specification of ML-DSA for Internet X.509 Public Key Infrastructure, see [RFC9881], Section 9.

8. Normative References

[FIPS-204] "Module-lattice-based digital signature standard", National Institute of Standards and Technology (U.S.), DOI 10.6028/nist.fips.204, August 2024, <https://csrc.nist.gov/pubs/fips/204/final>.

[I-D.rpe-ssh-mldsa] Petrov, R., "ML-DSA Public Key Algorithms for the Secure Shell (SSH) Protocol", Work in Progress, Internet-Draft, draft-rpe-ssh-mldsa-02, 19 October 2025, <https://datatracker.ietf.org/doc/html/draft-rpe-ssh-mldsa-02>.

[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>.

[RFC4251] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Protocol Architecture", RFC 4251, DOI 10.17487/RFC4251, January 2006, <https://www.rfc-editor.org/info/rfc4251>.

[RFC4252] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Authentication Protocol", RFC 4252, DOI 10.17487/RFC4252, January 2006, <https://www.rfc-editor.org/info/rfc4252>.

[RFC4253] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Transport Layer Protocol", RFC 4253, DOI 10.17487/RFC4253, January 2006, <https://www.rfc-editor.org/info/rfc4253>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[RFC6187] Igoe, K. and D. Stebila, "X.509v3 Certificates for Secure Shell Authentication", RFC 6187, DOI 10.17487/RFC6187, March 2011, <https://www.rfc-editor.org/info/rfc6187>.

[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>.

[RFC9519] Yee, P., "Update to the IANA SSH Protocol Parameters Registry Requirements", RFC 9519, DOI 10.17487/RFC9519, January 2024, <https://www.rfc-editor.org/info/rfc9519>.

[RFC9881] Massimo, J., Kampanakis, P., Turner, S., and B. E. Westerbaan, "Internet X.509 Public Key Infrastructure -- Algorithm Identifiers for the Module-Lattice-Based Digital Signature Algorithm (ML-DSA)", RFC 9881, DOI 10.17487/RFC9881, October 2025, <https://www.rfc-editor.org/info/rfc9881>.

9. Informative References

[IANA-SSH] IANA, "Secure Shell (SSH) Protocol Parameters", <https://www.iana.org/assignments/ssh-parameters>.

Acknowledgements

TBD

Author’s Address

Roumen Petrov
1750 Sofia
Bulgaria
Email: pkixssh@roumenpetrov.info
URI: https://roumenpetrov.info/secsh/