Internet-Draft:	draft-pereira-pq-signature-oids-00
Updates:	3279 (if approved)
Published:	20 November 2019
Intended Status:	Informational
Expires:	23 May 2020
Author:	G.C. C. F. Pereira
	evolutionQ / IQC, UWaterloo

Internet X.509 Public Key Infrastructure: Additional Post-Quantum Signature Algorithms and Identifiers

Abstract

This document updates RFC 3279 by specifying additional algorithm identifiers and ASN.1 encoding rules for the nine selected post-quantum digital signature algorithms running in the second round of NIST standardization process when using SHA3 or SHAKE as the underlying hashing algorithms. This specification applies (but it is not limited to) to the Internet X.509 Public Key infrastructure (PKI) and its post-quantum counterpart (not yet standardized), when post-quantum digital signatures are used to sign certificates and certificate revocation lists (CRLs).

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 23 May 2020.

Copyright Notice

Table of Contents

1. Introduction

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

This document profiles material under standardization by the NIST Post-Quantum Cryptography standardization process [NIST-PQ]. In particular, it provides the definition of algorithm identifiers and ASN.1 encoding formats for post-quantum digital signatures and subject public keys used in a post-quantum setting (not yet standardized) of the Internet X.509 Public Key Infrastructure (PKI) [RFC5280]. The nine digital signature algorithm candidates in the second round of the NIST process are considered along with their multiple security categories. Although not all nine algorithms may become actual standards, this specification will certainly cover the standardized ones. This way, implementors can run experiments with interoperable OIDs in order to evaluate the new post-quantum algorithms.

Precisely, this document defines additional contents for the "signatureAlgorithm", "signatureValue", "signature" and "subjectPublicKeyInfo" fields within Internet X.509 certificates and CRLs [RFC5280] when these objects are signed using pure post-quantum signature algorithms with SHA3 or SHAKE hash algorithms [FIPS202]. The SHA3 and SHAKE instances explicitly used in the object identifiers (OID) definitions are the ones adopted by the NIST second round candidates.

This document does not identify (optional) parameters for each algorithm identifier. This is because the ongoing candidates are likely to change parameter sets in the coming years and this document rather focuses on high-level identifiers.

The OIDs provided are suggested to go under the NIST signature algorithm arc in the OID tree [NIST-OIDS], i.e.:

joint-iso-itu-t(2) - country(16) - us(840) - organization(1) - gov(101) - csor(3) - nistAlgorithm(4) - sigAlgs(4).

Note that the field sigAlgs already contains identifiers ranging from 1 to 16, allocated for (EC)DSA and RSA. In this specification identifiers starting from 20 and above are reserved for the post-quantum signatures with their different parameter sets.

This document extends [RFC3279] that defines algorithm identifiers for classical signatures, and complements the update [RFC5758], which focuses on the specification of OIDs for extra classical signatures.

1.1. Terminology

- B: bytes
- CRL: Certificate Revocation List
- KiB: Kibibytes
- MiB: Mibibytes
- NIST: National Institute of Standards and Technology.
- OID: Object Identifier
- SHA3: Secure Hash Algorithm-3
- XOF: eXtendable-Output hash Function

2. Hash Functions

This section identifies hash functions for use with post-quantum digital signature algorithms. The hash functions SHA3-256, SHA3-384, SHA3-512 produce 256-bit, 384-bit and 512-bit outputs, respectively. On the other hand, functions SHAKE-128 and SHAKE-256 provide extendable (and thus variable) output sizes. They are all described in the "Secure Hash Standard" [FIPS202]. It is important to note that most of signature candidates in the NIST process adopt SHAKE which is the eXtendable-Output Function (XOF) of SHA3. The OIDs for SHAKE instances are retrieved from [NIST-OIDS]. The hash functions adopted by the NIST candidates have the following OIDs:

id-sha3-256 OBJECT IDENTIFIER ::= { joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) hashAlgs(2) 8 }

id-sha3-384 OBJECT IDENTIFIER ::= { joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) hashAlgs(2) 9 }

id-sha3-512 OBJECT IDENTIFIER ::= { joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) hashAlgs(2) 10 }

id-shake-128 OBJECT IDENTIFIER ::= { joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) hashAlgs(2) 11} }

id-shake-256 OBJECT IDENTIFIER ::= { joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistAlgorithm(4) hashAlgs(2) 12} }

Where id-sha3-<output size> refers to the Permutation-Based Hash functions and id-shake-<output size> to the eXtendable-Output Hash functions defined in [FIPS202]. When one of these OIDs appears in an AlgorithmIdentifier, all implementations MUST accept both NULL and absent parameters as legal and equivalent encodings. Conforming certification authority (CA) implementations SHOULD use SHA3-256, SHA3-384, SHA3-512, SHAKE-128 or SHAKE-256 when generating post-quantum certificates or CRLs.

3. Post-quantum Digital Signature Algorithms

This section defines OIDs for the following signature algorithms: Crystals-Dilithium, Falcon, GeMSS, LUOV, MQDSS, PICNIC, qTESLA, Rainbow and SPHINCS+ when instantiated with SHA3-256, SHA3-384, SHA3-512, SHAKE-128 or SHAKE-256. Note that each algorithm can offer parameter sets for at most five quantum security categories (numbered 1 to 5), as suggested by NIST [NIST-PQ]. Moreover, different category parameters may use the same hash output size, differently from conventional elliptic curve-based signatures where the hash size indicates the security level.

3.1. CRYSTALS-Dilithium Signature

CRYSTALS-Dilithium is a digital signature whose security is based on the hardness of lattice problems [CRYSTALS-Dilithium]. It provides four parameter sets for the security categories 1, 2, 3 and 4, and offers a comparatively small footprint for the public key + signature size combination, which ranges from 2 to 5 KiB. All operations (KeyGen, Sign and Verify) are relatively quite efficient to compute (hundreds of Kcycles using AVX2 instructions). CRYSTALS-Dilithium SHALL be used in conjunction with SHAKE-256 for all security categories above.

When SHAKE-256 is used, the OIDs are:

id-dilithium-shake256 OBJECT IDENTIFIER ::= { joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) algorithms(4) id-dilithium-shake(3) 20 }.

When the id-dilithium-shake256 identifier appears in the algorithm field as an AlgorithmIdentifier, the encoding MUST omit the parameters field. That is, the AlgorithmIdentifier SHALL be a SEQUENCE of one component, the OID id-dilithium-shake256.

3.2. FALCON Signature

FALCON is a digital signature whose security is based on the hardness of lattice problems [FALCON-ref]. Authors provide two parameter sets for security categories 1 and somewhere between 4-5. FALCON offers relatively small footprints for the public key + signature size metric, which ranges

from about 1.5-3.0 KiB. In terms of computational efficiency, KeyGen is relatively slow (tens of Mcycles), Sign is moderate (hundreds of Kcycles) and Verify is relatively fast (hundreds of Kcycles). FALCON SHALL be used in conjunction with SHAKE-256 for all security categories above.

When SHAKE-256 is used, the OIDs are:

id-falcon-shake256 OBJECT IDENTIFIER ::= { joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) algorithms(4) id-falcon-shake(3) 21 }.

When the id-falcon-shake256 algorithm identifier appears in the algorithm field as an AlgorithmIdentifier, the encoding MUST omit the parameters field. That is, the AlgorithmIdentifier SHALL be a SEQUENCE of one component, the OID id-falcon-shake256.

3.3. GeMSS Signature

GeMSS is a digital signature whose security is based on the hardness of solving non-linear systems of multivariate equations over finite fields [GeMSS-ref]. It provides nine parameter sets, three for each one of the security categories 1, 3 and 5. GeMSS uses a name convention to distinguish between the three parameter sets within a category, i.e., GeMSS, BlueGeMSS and RedGeMSS. In terms of space, public-keys are relatively large (about 400 KiB - 3 MiB) but signatures are very small (about 33-75 B). In terms of computational efficiency, KeyGen is relatively slow (hundreds of Mcycles), Sign is moderate (few million cycles) and Verify is relatively fast (hundreds of Kcycles). GeMSS SHALL be used in conjunction with a SHA3 hash function with output sizes of 256, 384 and 512 bits for matching the three security categories above, respectively.

When SHA3 is used, the OIDs are:

id-gemss-sha3-256 OBJECT IDENTIFIER ::= { joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) algorithms(4) id-gemss-sha3(3) 22 }.

id-gemss-sha3-384 OBJECT IDENTIFIER ::= { joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) algorithms(4) id-gemss-sha3(3) 23 }.

id-gemss-sha3-512 OBJECT IDENTIFIER ::= { joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) algorithms(4) id-gemss-sha3(3) 24 }.

When the id-gemss-sha3-256, id-gemss-sha3-384 or id-gemss-sha3-512 algorithm identifier appears in the algorithm field as an AlgorithmIdentifier, the encoding MUST omit the parameters field. That is, the AlgorithmIdentifier SHALL be a SEQUENCE of one component, the OID id-gemss-sha3-256, id-gemss-sha3-384, or id-gemss-sha3-512.

3.4. LUOV Signature

LUOV is a digital signature whose security is based on the hardness of solving non-linear systems of multivariate equations over finite fields [LUOV]. Authors provide 6 parameter sets, two for each one of the security categories 2, 4 and 5. In order to differentiate parameter sets within the same category, LUOV uses the name convention LUOV-r-m-v, where r, m, and v are integers indicating the extension degree, number of equations and number of vinegar variables, respectively. Note that it is likely that values for r, m and v may slightly change over time as cryptanalysis evolve and adopting these parameters in the OIDs is likely to make this specification obsolete in a near future. In terms of space, LUOV public keys have moderate size (12-75 KiB) and signatures are small (300-4400 B). In terms of computational efficiency, KeyGey, Sign and Verify have moderate speed (a few Mcycles, hundreds of Kcycles and hundreds of Kcycles, respectively). LUOV SHALL be used in conjunction with SHAKE-128 for matching category 2 or SHAKE-256 for categories 4 and 5.

When SHAKE is used, the OIDs are:

id-luov-shake-128 OBJECT IDENTIFIER ::= { joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) algorithms(4) id-luov-shake(3) 25 }.

id-luov-shake-256 OBJECT IDENTIFIER ::= { joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) algorithms(4) id-luov-shake(3) 26 }.

When the id-luov-shake-128 or id-luov-shake-256 algorithm identifier appears in the algorithm field as an AlgorithmIdentifier, the encoding MUST omit the parameters field. That is, the AlgorithmIdentifier SHALL be a SEQUENCE of one component, the OID id-luov-shake-128, or id-luov-shake-256.

3.5. MQDSS Signature

MQDSS is a digital signature whose security is based on the hardness of solving non-linear systems of multivariate equations over finite fields [MQDSS]. It provides two parameter sets, one for a security category 1-2 (in between) and another for category 3-4 (in between). In terms of space, public keys are small (46-64 B) and signatures are moderate (about 2.6-5.5 KB). From a computational efficiency point of view, KeyGen, Sign and Verify have moderate speed (few million cycles). MQDSS SHALL be used in conjunction with SHAKE-256 for matching the security categories above.

When SHAKE is used, the OIDs are:

id-mqdss-shake-256 OBJECT IDENTIFIER ::= { joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) algorithms(4) id-mqdss-shake(3) 27 }.

When the id-mqdss-shake-256 algorithm identifier appears in the algorithm field as an AlgorithmIdentifier, the encoding MUST omit the parameters field. That is, the AlgorithmIdentifier SHALL be a SEQUENCE of one component, the OID id-mqdss-shake-256.

3.6. PICNIC Signature

PICNIC is a digital signature whose security is based on the hardness of breaking symmetric primitives such as block ciphers and hash functions [PICNIC]. Authors provide nine parameter sets, three for each one of the security categories 1, 3 and 5. In terms of space, PICNIC public keys are small (32-64 B) and signatures have moderate sizes (1.7-26.2 KiB). From the computational efficiency point of view, KeyGen is fast (tens of Kcycles), Sign and Verify have moderate speed (a few up to hundreds of Mcycles). PICNIC SHALL be used in conjunction with SHAKE-128 for matching category 1 or SHAKE-256 for categories 3 and 5.

When SHAKE is used, the OIDs are:

id-picnic-shake-128 OBJECT IDENTIFIER ::= { joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) algorithms(4) id-picnic-shake(3) 28 }.

id-picnic-shake-256 OBJECT IDENTIFIER ::= { joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) algorithms(4) id-picnic-shake(3) 29 }.

When the id-picnic-shake-128 or id-picnic-shake-256 algorithm identifier appears in the algorithm field as an AlgorithmIdentifier, the encoding MUST omit the parameters field. That is, the AlgorithmIdentifier SHALL be a SEQUENCE of one component, the OID id-picnic-shake-128 or id-picnic-shake-256.

3.7. qTesla Signatures

qTesla is a digital signature whose security is based on the hardness of lattice problems [qTesla]. qTesla features two types of design with respect to security, i.e., provable and heuristic security. The heuristic parameters were recently considered weaker and are being removed. For the provably-secure versions, two parameter sets achieving categories 1 and 3 are provided. qTesla SHALL be used in conjunction with SHAKE-128 for matching category 1 or SHAKE-256 for matching category 3.

When SHAKE is used, the OIDs are:

id-qtesla-shake-128 OBJECT IDENTIFIER ::= { joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) algorithms(4) id-qtesla-shake(3) 30 }.

id-qtesla-shake-256 OBJECT IDENTIFIER ::= { joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) algorithms(4) id-qtesla-shake(3) 31 }.

When the id-qtesla-shake-128 or id-qtesla-shake-256 algorithm identifier appears in the algorithm field as an AlgorithmIdentifier, the encoding MUST omit the parameters field. That is, the AlgorithmIdentifier SHALL be a SEQUENCE of one component, the OID id-qtesla-shake-128 or id-qtesla-shake-256.

3.8. Rainbow Signatures

Rainbow is a digital signature whose security is based on the hardness of solving non-linear systems of multivariate equations over finite fields [RAINBOW]. Authors provide 6 parameter sets, two for each one of the security categories 1, 3 and 5. Half of the parameter sets are intended for a compressed key variant of Rainbow. In terms of space, Rainbow public keys are moderate to large (58-492 KiB for the compressed versions) and signatures are small (64-204 B). From a computational efficiency point of view, KeyGen has moderate speed (tens of Mcyles), Sign and Verify are fast (hundreds of Kcycles). Rainbow SHALL be used in conjunction with SHA3-256, SHA3-384 and SHA3-512 for matching categories 1, 3 and 5, respectively.

When SHA3 is used, the OIDs are:

id-rainbow-sha3-256 OBJECT IDENTIFIER ::= { joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) algorithms(4) id-rainbow-sha3(3) 32 }.

id-rainbow-sha3-384 OBJECT IDENTIFIER ::= { joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) algorithms(4) id-rainbow-sha3(3) 33 }.

id-rainbow-sha3-512 OBJECT IDENTIFIER ::= { joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) algorithms(4) id-rainbow-sha3(3) 34 }.

When the id-rainbow-sha3-256, id-rainbow-sha3-384 or id-rainbow-sha3-512 algorithm identifier appears in the algorithm field as an AlgorithmIdentifier, the encoding MUST omit the parameters field. That is, the AlgorithmIdentifier SHALL be a SEQUENCE of one component, the OID id-rainbow-sha3-256, id-rainbow-sha3-384 or id-rainbow-sha3-512.

3.9. SPHINCS+ Signature

SPHINCS+ is a digital signature whose security is based on problems related to hash functions [SPHINCS-PLUS]. Authors provide multiple parameter sets per security category covering categories 1, 3 and 5. In terms of space, SPHINCS+ public keys are small (32-64 B) and signatures have

moderate sizes (8-50 KiB). From a computational efficiency point of view, KeyGen, Sign and Verify have moderate to slow speed (ranging from a few to hundreds of Mcycles). This document considers only instances based on the NIST standardized hash functions [FIPS202], i.e., SHAKE and SHA3.

When SHAKE or SHA3 is used, the OIDs are:

id-sphincsp-shake-256 OBJECT IDENTIFIER ::= { joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) algorithms(4) id-sphincsp-shake(3) 35 }.

id-sphincsp-sha3-256 OBJECT IDENTIFIER ::= { joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) algorithms(4) id-sphincsp-sha3(3) 36 }.

When the id-sphincsp-shake-256 or id-sphincsp-sha3-256 algorithm identifier appears in the algorithm field as an AlgorithmIdentifier, the encoding MUST omit the parameters field. That is, the AlgorithmIdentifier SHALL be a SEQUENCE of one component, the OID id-sphincsp-shake-256 or id-sphincsp-sha3-256.

4. IANA Considerations

The OIDs defined in this document are suggested to go under the NIST arc (sigAlgs) as it appears to be the most natural for them. Since we expect these algorithms to be a NIST standard. Other arcs may be considered as well.

5. Security Considerations

As mentioned in Section 1, the object identifiers defined in this document take into account the underlying hash function adopted in each post-quantum signature scheme, and does not introduce extra definitions for the (optional) parameters for each scheme. This is because it is likely that particular parameter sets change in the next couple years of standardization until 2024. Defining the names in terms of security categories is deemed to be a suitable approach as NIST is unlikely to change the categories themselves.

6. Acknowledgement

The author is thankful for the useful discussions and comments by John Gray and Mike Ounsworth from Entrust Datacard.

7. References

7.1. Normative References

[FIPS202] National Institute of Standards and Technology (NIST), "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions", August 2015 , <http://dx.doi.org/10.6028/NIST.FIPS.202>.

[NIST-OIDS] National Institute of Standards and Technology (NIST), "Computer Security Objects Register", November 2019 , <https://csrc.nist.gov/Projects/Computer-Security-Objects-Register/Algorithm-Registration>.

[NIST-PQ] National Institute of Standards and Technology (NIST), "Post-Quantum Cryptography", November 2019 , <https://www.nist.gov/pqcrypto>.

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

[RFC3279] Bassham, L., Polk, W., and R. Housley, "Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3279, April 2002 , <https://www.rfc-editor.org/info/rfc3279>.

[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>.

[RFC5758] Dang, Q., Santesson, S., Moriarty, K., Brown, D., and T. Polk, "Internet X.509 Public Key Infrastructure: Additional Algorithms and Identifiers for DSA and ECDSA", RFC 5758, January 2010 , <https://www.rfc-editor.org/info/rfc5758>.

7.2. Informative References

[CRYSTALS-Dilithium] Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., and D. Stehle, "Crystals-dilithium: A lattice-based digital signature scheme", IACR Cryptology ePrint Archive , 2017 , <https://eprint.iacr.org/2017/633>.

[FALCON-ref] Fouque, P-A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., and Z. Zhang, "Falcon: Fast-Fourier lattice-

based compact signatures over NTRU", IACR Cryptology ePrint Archive , 2018 , <https://falcon-sign.info/>.

[GeMSS-ref] Casanova, A., Faugere, J-C., Macario-Rat, G., Patarin, J., Perret, L., and J. Ryckeghem, "GeMSS: A Great Multivariate Short Signature", 2017 , <https://www-polsys.lip6.fr/Links/NIST/GeMSS.html>.

[LUOV] Beullens, W. and B. Preneel, "LUOV: Field lifting for smaller UOV public keys", IACR Cryptology ePrint Archive , 2017 , <https://eprint.iacr.org/2017/776>.

[MQDSS] Hulsing, A., Rijneveld, J., Samardjiska, S., and P. Schwabe, "From 5-pass MQ-based identification to MQ-based signatures", IACR Cryptology ePrint Archive , 2016 , <https://eprint.iacr.org/2016/708>.

[PICNIC] Chase, M., Derler, D., Goldfeder, S., Orlandi, C., Ramacher, S., Rechberger, C., Slamanig, D., and G. Zaverucha, "Post-quantum zero-knowledge and signatures from symmetric-key primitives", DOI 10.1145/3133956.3133997, ACM SIGSAC Conference on Computer and Communications Security pp. 1825--1842, 2017 , <https://doi.org/10.1145/3133956.3133997>.

[qTesla] Alkim, E., L.M. Barreto, P.S., Bindel, N., Longa, P., and J. E. Ricardini, "The Lattice-Based Digital Signature Scheme qTESLA", IACR Cryptology ePrint Archive , November 2019 , <https://eprint.iacr.org/2019/085>.

[RAINBOW] Ding, J. and D. Schmidt, "Rainbow, a New Multivariable Polynomial Signature Scheme", 2005 , <https://link.springer.com/chapter/10.1007/11496137_12>.

[SPHINCS-PLUS] Bernstein, D.J., Dobraunig, C., Eichlseder, M., Fluhrer, S., Gazdag, S-L., Hulsing, A., Kampanakis, P., Kolbl, S., Lange, T., Lauridsen, M., Mendel, F., Niederhagen, R., Rechberger, C., Rijneveld, J., and P. Schwabe, "SPHINCS+", 2017 , <https://sphincs.org/>.

Author's Address

Geovandro C. C. F. Pereira
evolutionQ Inc. / IQC, University of Waterloo
Email: geovandro.pereira@evolutionq.com