Workgroup: None
Internet-Draft:
draft-prorock-cose-post-quantum-signatures-00
Published: 5 March 2022
Intended Status: Standards Track
Expires: 6 September 2022
Authors: M. Prorock (mesur.io), O. Steele (Transmute), R. Misoczki (Google), M. Osborne (IBM), C. Cloostermans (NXP)

JSON Encoding for Post Quantum Signatures

Abstract

This document describes JSON and CBOR serializations for several post quantum cryptography (PQC) based suites.

This document does not define any new cryptography, only seralizations of existing cryptographic systems.

This document registers key types for JOSE and COSE, specifically PQK, CRYDI, pset.

This document registers signature algorithms types for JOSE and COSE, specifically CRYDI3 and others as required for various post quantum signature schemes.

Status of This Memo

Copyright Notice

Table of Contents

Authors' Addresses

1. Notational Conventions

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].

2. Terminology

The following terminology is used throughout this document:

PK The public key for the signature scheme.
SK The secret key for the signature scheme.
signature The digital signature output.
message The input to be signed by the signature scheme.
sha256 The SHA-256 hash function defined in [RFC6234].
shake256 The SHAKE256 hash function defined in [RFC8702].

3. CRYSTALS-Dilithium

3.1. Overview

This section of the document describes the lattice signature scheme CRYSTALS-Dilithium (CRYDI). The scheme is based on "Fiat-Shamir with Aborts"[Lyu09, Lyu12] utlizing a matrix of polynomials for key material, and a vector of polynomials for signatures. The parameter set is strategically chosen such that the signing algorithm is large enough to maintain zero-knowledge properties but small enough to prevent forgery of signatures. An example implementation and test vectors are provided.

CRYSTALS-Dilithium is a Post Quantum approach to digital signatures that is an algorithmic apprach that seeks to ensure key pair and signing properties that is a strong implementation meeting Existential Unforgeability under Chosen Message Attack (EUF-CMA) properties, while ensuring that the security levels reached meet security needs for resistance to both classical and quantum attacks. The algoritm itself is based on hard problems over module lattices, specifically Ring Learning with Errors (Ring-LWE). For all security levels the only operations required are variants of Keccak and number theoretic transforms (NTT) for the ring Zq[X]/(X256+1). This ensures that to increase or decrease the security level invovles only the change of parameters rather than re-implementation of a related algorithm.

While based on Ring-LWE, CRYSTALS-Dilithium has less algebraic structure than direct Ring-LWE implementations and more closely resembles the unstructured lattices used in Learning with Errors

(LWE). This brings a theorectical protection against future algebraic attacks on Ring-LWE that may be developed.

CRYSTALS-Dilithium, brings several advantages over other approaches to signature suites:

* Post Quantum in nature - use of lattices and other approaches that should remain hard problems even when under attack utilizing quantum approaches
* Simple implementation while maintaing security - a danger in many possible approaches to cryptography is that it may be possible inadvertantly introduce errors in code that lead to weakness or decreases in security level
* Signature and Public Key Size - compared to other post quantum approaches a reasonable key size has been achieved that also preserves desired security properties
* Conservative parameter space - parameterization is utilized for the purposes of defining the sizes of marices in use, and thereby the number of polynomials described by the key material.
* Parameter set adjustment for greater security - increasing this matrix size increases the number of polynomials, and thereby the security level
* Performance and optimization - the approach makes use of well known transforms that can be highly optimized, especially with use of hardware optimizations without being so large that it cannot be deployed in embedded or IoT environments without some degree of optimization.

The primary known disadvantage to CRYSTALS-Dilithium is the size of keys and signatures, especially as compared to classical approaches for digital signing.

3.2. Parameters

Unlike certain other approaches such as Ed25519 that have a large set of parameters, CRYSTALS-Dilithium uses distinct numbers of paramters to increase or decrease the security level according to the required level for a particular scenario. Under DILITHIUM-Crustals, the key parameter specificiation determines the size of the matrix and thereby the number of polynomials that describe he lattice. For use according to this specification we do not recommend a parameter set of less than 3, which should be sufficient to maintain 128bits of security for all known classical and quantum attacks. Under a parameter set at NIST level 3, a 6x5 matrix is utilized that thereby consists of 30 polynomials.

3.2.1. Parameter sets

Parameter sets are identified by the corresponding NIST level per the table below

NIST Level	Matrix Size	memory in bits
2	4x4	97.8
3	6x5	138.7
5	8x7	187.4

Table 1

3.3. Core Operations

This section defines core operations used by the signature scheme, as proposed in [CRYSTALS-Dilithium].

3.3.1. Generate

See [CRYSTALS-Dilithium]

3.3.2. Sign

See [CRYSTALS-Dilithium]

3.3.3. Verify

See [CRYSTALS-Dilithium]

3.4. Using CRYDI with JOSE

Basing off of this

3.4.1. CRYDI Key Representations

A new key type (kty) value "PQK" (Post Quantum Key Pair) is defined for public key algorithms that use base 64 encoded strings of the underlying binary materia as private and public keys and that support cryptographic sponge functions. It has the following parameters:

* The parameter "kty" MUST be "PQK".

* The parameter "alg" MUST be specified, and its value MUST be one of the values specified in table **TBD**.

* The parameter "pset" MUST be specfied to indicate the not only paramter set in use for the algorithm, but SHOULD also reflect the targeted NIST level for the algorithm in combination with the specified paramter set. For "alg" "CRYDI" one of the described parameter sets "2", "3", or "5" MUST be specified. Parameter set

"3" or above SHOULD be used with "CRYDI" for any situation requiring at least 128bits of security against both quantum and classical attacks

* The parameter "x" MUST be present and contain the public key encoded using the base64url [RFC4648] encoding.

* The parameter "xs" MAY be present and contain the shake256 of the public key encoded using the base64url [RFC4648] encoding.

* The parameter "d" MUST be present for private keys and contain the private key encoded using the base64url encoding. This parameter MUST NOT be present for public keys.

* The parameter "ds" MAY be present for private keys and contain the shake256 of the private key encoded using the base64url encoding. This parameter MUST NOT be present for public keys.

Sizes of various key and signature material is as follows (for "pset" value "2")

Variable	**Paramter Name**	**Paramter Set**	**Size**	**base64url encoded size**
Signature	sig	2	3293	4393
Public Key	x	2	1952	2605
Private Key	d	2	4000	5337

Table 2

When calculating JWK Thumbprints [RFC7638], the four public key fields are included in the hash input in lexicographic order: "kty", "pset", and "x".

3.4.2. CRYDI Algorithms

In order to reduce the complexity of the key representation and signature representations we register a unique algorithm name per pset. This allows us to omit registering the pset term, and reduced the likelyhood that it will be misused. These alg values are used in both key representations and signatures.

kty	**alg**	**Paramter Set**
PQK	CRYDI5	5
PQK	CRYDI3	3
PQK	CRYDI2	2

Table 3

3.4.2.1. Public Key

Per section 5.1 of [CRYSTALS-Dilithium]:

> The public key, containing p and t1, is stored as the concatenation of the bit-packed representations of p and t1 in this order. Therefore, it has a size of 32 + 288 kbytes.

The public key is represented as x and encoded using base64url encoding as described in [RFC7517].

Example public key using only required fields:

```
=============== NOTE: '\' line wrapping per RFC 8792 ================
```


```
{
  "kty": "PQK",
  "alg": "CRYDI3",
  "x": "z7u7GwhsjjnfHH3Nkrs2xvvw020Rcw5ymdlTnhRenjDdrOO+nfXRVUZVy9q1\
5zDn77zTgrIskM3WX8bqslc+B1fq12iA/wxD2jc1d6j+YjKCtkGH26OR7vc0YC2ZiMzW\
zGl7yebt7JkmjRbN1N+u/2fAKFLuziMcLNP6WLoWbMqxoC2XOOVNAWX3QjXrCcGU23Nr\
imtdmWz5NrP43E592Sctt5M+SVlfgQeYv8pHmtkQknE8/jr7TrgNpuiV7nXmhWHTMJ4I\
zoGXgq43odFFthboEdKNT/enyu+VvUGoIJ6cN8C/1B6o1WlYHEaL0BEIFFbAiAhZ/vnf\
cUYMaVPqsDJuETsjetcE32kGCD7Jkume2tO68DlIhB/2Z2JX8mkcbxFI6KrmXiRxXQj9\
9LVn1fEzdf3Vfpcs/C3omsFGqmTpLDK+AvW/SWVkDi2NKq7hL/AyxlW2u2cqVErQZUTS\
Z+ic6V8kZfxr3gRMnH0KuF5BtjleZ/yVvqqPjwPOZegCKEl2Gd8duhcUde7CR55pil1o\
UXy5AwgCcZTdEcJn1OPObGoots9T19gw1x4vnZCQUKVDPZuZ1gIkGqDUYXS0lcNTjCMs\
miFEmnOZvB88jxULpb1vl9HoQ3ocM2oZu4AZRt9G/L07Mwcui0uFCWtAIau+2gqNAn/Z\
AS10l0j2N0LLtAaOxoF+Ctzscrt0ZMyGHmoQ9daHkpUvEq0cO8hDtLplnq3lQIIIfROQ\
jcNs9vNKBu87COBjukZD+L8vV4zy8FNO59MCSb9UCLwz2xvfdI1js9/J7hTGaVec8VPx\
md42yPFrGw5Na1oefm8vW49EDmevc8AjAtwDirRBDFv9pX3+5S+M6jhteSLYvpKJXQT1\
zs1379KvIHwkn9VHpA+PiUUw9TgF6xF8xWEGSNlOo1Vn1xtM3givehjYxJ5p5/kBEFZI\
DCyFzstAirJ2GadNhae+P1JFZzJWnX5jaLwzldquZwF3yTzNho4sgBA+fKqiXcgn2nw1\
vz0Dkbxr6cMaUool0eFScU1nAz1Z39W64LtT2nEuYsORx/ht2RzJxxFc21X3nLeEDFCe\
NkNDxQFBSfpZjKKgJtXEx23mp+CbBVMrbagsLnzsAGLYbnroVmATU5Iqr6LgYBpuFs+N\
Rkq7ZXh6CZPukMGQbcOGuNwO6NBuuMNhir5ayGk1ZBiW82C7Nu0hs2pLcgNqWMtt1+LW\
8R96KyoSc784ZYAZ40QqvoySwmxQPBRTRJ+wB0sVpGBLTxdY9Gw3pXeXN5nao340d2ZA\
7YEMlqcTHCAv3F8B9ewl7OfQlmg6bvdMuoVdVE+p0er7IAmWMRgviIzYv9sKEEQrCmua\
2qL5xPSbD05KRf8ZAZ2B8lSCDR1nzXrQXZbXBKJivsCVQDuzxrwGE0gqRMpbk4f5GYCG\
4i/O8Knoru+jjf6wVQDYKfyz1QUGRlXHkGUGlXfv03r7UbJugycjVO5kbGxhoZkqOq8z\
ZEpkefvrrNoxeotw/z4QpjI8JlY97GDb0mGVHbmdHugjMtVTGhVJFBbPIinmR+emt7O+\
4qOr7ywRxCvt2lziWtpPBwaf/1XDnN5Gesex1gR1YrcTRNmB808b01sxLQmxcTt4eQ0/\
LUkas7qTJ3AQThOfDdtIpkqsthsBFy+WjSQuoXCYMRcPi6MlpxJndDF32lCnL1ranV6e\
F2ST0SYT+NwNDesMzTRmNbHUW5KAhu0k9WABTvcM5ba0Uq6iOa1NsFrcLag+KhxN6HPn\
oobwJ/EsDi5S7TAl8WrjqIhZ8x6h9eRRXerpaOw/FYk+2MpWByp/98VE12/EwOqAIiPp\
elAvUeMOlRkpG64bJsmyYtHuNWgcv5Qiy7/eGw9ZpvB3J3G3jxvbynExqdFyDc067EKi\
5WxDFPuZUjkfKpekNvzQuIrqs49BzcRyMt5ndEVE21TPPfZ/R8B7Rxnb2LiK+hQc+cc9\
pEEaWgwAOiMILcp/1CyY6ImdO6RHsxwflMH7gej+hN41kaoEghIOl9kMGTLZbq5Pc8Pz\
6F2LKTBMJWg9o/0blvilMH9EPblcLeF/bR1AZTUD6ZFdi2TxN6Epn3QVqeG/qPm1EBTF\
Gw1V92m6/08Dd6zI1HPqwKbkHx4F567owofKHaM2imin0yVUpwxoRJrulRHMCB3tn8C4\
ZpFl+sGV3Gip3tKlS7PKQkTqI6DMwxEbdrvtdY1sHZagpclLDisA/yFT4RR2m3VNJR9P\
6Nx3teqN1eg6RXmD/MlKCdWrlcjZ/6yeIQYwbr9CjItY/tLQX2gtAR1SXOh99UUBVv+Z\
E03VOZ+Ecsc78lSB9G/6n6CFzlbk/HgAF+cu0yMbGnEM8W3mTUspS4JBACwk5w0XWNNQ\
DWVEdgzuLGhPq+hYExDjVZrLELhkH8YgZA+7RXXUZHM/joNOGHUhpUG/bFo3ktnaILCu\
xsOXMUbDC3VcitFFHsGK1svtcERDFxk1HA8pGa59jT0do6n3wEbnBDU1soKNFtpmcVkE\
Ul3XpvuoW3BgCwJzBUCWvPs47DJRgGxO11bSaEYYlhTVaaShcvzgz46AkqO+Q7TjckDP\
/8uzsSQk0AbuhxWFQpSiBP8OZ/U="
}
```


Example public key including optional fields:

```
=============== NOTE: '\' line wrapping per RFC 8792 ================

{
  "kid": "key-0",
  "kty": "PQK",
  "alg": "CRYDI3",
  "key_ops": ["verify"],
  "xs": "z3uZQVjflnRZDSZn1e8g4oKH4YUU6TnpvkU4WrrGdXw=",
  "ds": "5DuZ8XoJQirc/5TE23tBcoGoHo+JTj1+9ULLXtCiySU=",
  "x": "z7u7GwhsjjnfHH3Nkrs2xvvw020Rcw5ymdlTnhRenjDdrOO+nfXRVUZVy9q1\
5zDn77zTgrIskM3WX8bqslc+B1fq12iA/wxD2jc1d6j+YjKCtkGH26OR7vc0YC2ZiMzW\
zGl7yebt7JkmjRbN1N+u/2fAKFLuziMcLNP6WLoWbMqxoC2XOOVNAWX3QjXrCcGU23Nr\
imtdmWz5NrP43E592Sctt5M+SVlfgQeYv8pHmtkQknE8/jr7TrgNpuiV7nXmhWHTMJ4I\
zoGXgq43odFFthboEdKNT/enyu+VvUGoIJ6cN8C/1B6o1WlYHEaL0BEIFFbAiAhZ/vnf\
cUYMaVPqsDJuETsjetcE32kGCD7Jkume2tO68DlIhB/2Z2JX8mkcbxFI6KrmXiRxXQj9\
9LVn1fEzdf3Vfpcs/C3omsFGqmTpLDK+AvW/SWVkDi2NKq7hL/AyxlW2u2cqVErQZUTS\
Z+ic6V8kZfxr3gRMnH0KuF5BtjleZ/yVvqqPjwPOZegCKEl2Gd8duhcUde7CR55pil1o\
UXy5AwgCcZTdEcJn1OPObGoots9T19gw1x4vnZCQUKVDPZuZ1gIkGqDUYXS0lcNTjCMs\
miFEmnOZvB88jxULpb1vl9HoQ3ocM2oZu4AZRt9G/L07Mwcui0uFCWtAIau+2gqNAn/Z\
AS10l0j2N0LLtAaOxoF+Ctzscrt0ZMyGHmoQ9daHkpUvEq0cO8hDtLplnq3lQIIIfROQ\
jcNs9vNKBu87COBjukZD+L8vV4zy8FNO59MCSb9UCLwz2xvfdI1js9/J7hTGaVec8VPx\
md42yPFrGw5Na1oefm8vW49EDmevc8AjAtwDirRBDFv9pX3+5S+M6jhteSLYvpKJXQT1\
zs1379KvIHwkn9VHpA+PiUUw9TgF6xF8xWEGSNlOo1Vn1xtM3givehjYxJ5p5/kBEFZI\
DCyFzstAirJ2GadNhae+P1JFZzJWnX5jaLwzldquZwF3yTzNho4sgBA+fKqiXcgn2nw1\
vz0Dkbxr6cMaUool0eFScU1nAz1Z39W64LtT2nEuYsORx/ht2RzJxxFc21X3nLeEDFCe\
NkNDxQFBSfpZjKKgJtXEx23mp+CbBVMrbagsLnzsAGLYbnroVmATU5Iqr6LgYBpuFs+N\
Rkq7ZXh6CZPukMGQbcOGuNwO6NBuuMNhir5ayGk1ZBiW82C7Nu0hs2pLcgNqWMtt1+LW\
8R96KyoSc784ZYAZ40QqvoySwmxQPBRTRJ+wB0sVpGBLTxdY9Gw3pXeXN5nao340d2ZA\
7YEMlqcTHCAv3F8B9ewl7OfQlmg6bvdMuoVdVE+p0er7IAmWMRgviIzYv9sKEEQrCmua\
2qL5xPSbD05KRf8ZAZ2B8lSCDR1nzXrQXZbXBKJivsCVQDuzxrwGE0gqRMpbk4f5GYCG\
4i/O8Knoru+jjf6wVQDYKfyz1QUGRlXHkGUGlXfv03r7UbJugycjVO5kbGxhoZkqOq8z\
ZEpkefvrrNoxeotw/z4QpjI8JlY97GDb0mGVHbmdHugjMtVTGhVJFBbPIinmR+emt7O+\
4qOr7ywRxCvt2lziWtpPBwaf/1XDnN5Gesex1gR1YrcTRNmB808b01sxLQmxcTt4eQ0/\
LUkas7qTJ3AQThOfDdtIpkqsthsBFy+WjSQuoXCYMRcPi6MlpxJndDF32lCnL1ranV6e\
F2ST0SYT+NwNDesMzTRmNbHUW5KAhu0k9WABTvcM5ba0Uq6iOa1NsFrcLag+KhxN6HPn\
oobwJ/EsDi5S7TAl8WrjqIhZ8x6h9eRRXerpaOw/FYk+2MpWByp/98VE12/EwOqAIiPp\
elAvUeMOlRkpG64bJsmyYtHuNWgcv5Qiy7/eGw9ZpvB3J3G3jxvbynExqdFyDc067EKi\
5WxDFPuZUjkfKpekNvzQuIrqs49BzcRyMt5ndEVE21TPPfZ/R8B7Rxnb2LiK+hQc+cc9\
pEEaWgwAOiMILcp/1CyY6ImdO6RHsxwflMH7gej+hN41kaoEghIOl9kMGTLZbq5Pc8Pz\
6F2LKTBMJWg9o/0blvilMH9EPblcLeF/bR1AZTUD6ZFdi2TxN6Epn3QVqeG/qPm1EBTF\
Gw1V92m6/08Dd6zI1HPqwKbkHx4F567owofKHaM2imin0yVUpwxoRJrulRHMCB3tn8C4\
ZpFl+sGV3Gip3tKlS7PKQkTqI6DMwxEbdrvtdY1sHZagpclLDisA/yFT4RR2m3VNJR9P\
6Nx3teqN1eg6RXmD/MlKCdWrlcjZ/6yeIQYwbr9CjItY/tLQX2gtAR1SXOh99UUBVv+Z\
E03VOZ+Ecsc78lSB9G/6n6CFzlbk/HgAF+cu0yMbGnEM8W3mTUspS4JBACwk5w0XWNNQ\
DWVEdgzuLGhPq+hYExDjVZrLELhkH8YgZA+7RXXUZHM/joNOGHUhpUG/bFo3ktnaILCu\
xsOXMUbDC3VcitFFHsGK1svtcERDFxk1HA8pGa59jT0do6n3wEbnBDU1soKNFtpmcVkE\
Ul3XpvuoW3BgCwJzBUCWvPs47DJRgGxO11bSaEYYlhTVaaShcvzgz46AkqO+Q7TjckDP\
/8uzsSQk0AbuhxWFQpSiBP8OZ/U="
}
```

3.4.2.2. Private Key

Per section 5.1 of [CRYSTALS-Dilithium]:

> The secret key contains p,K,tr,s1,s2 and t0 and is also stored as a bit-packed representation of these quantities in the given order. Consequently, a secret key requires 64 + 48 + 32((k+l) * dlog (2n+1)e + 14k) bytes. For the weak, medium and high security level this is equal to 112 + 576k+ 128l bytes. With the very high security parameters one needs 112 + 544k + 96l = 3856 bytes.

The private key is represented as d and encoded using base64url encoding as described in [RFC7517].

Example private key using only required fields:

```
=============== NOTE: '\' line wrapping per RFC 8792 ================

{
  "kty": "PQK",
  "alg": "CRYDI3",
  "x": "z7u7GwhsjjnfHH3Nkrs2xvvw020Rcw5ymdlTnhRenjDdrOO+nfXRVUZVy9q1\
5zDn77zTgrIskM3WX8bqslc+B1fq12iA/wxD2jc1d6j+YjKCtkGH26OR7vc0YC2ZiMzW\
zGl7yebt7JkmjRbN1N+u/2fAKFLuziMcLNP6WLoWbMqxoC2XOOVNAWX3QjXrCcGU23Nr\
imtdmWz5NrP43E592Sctt5M+SVlfgQeYv8pHmtkQknE8/jr7TrgNpuiV7nXmhWHTMJ4I\
zoGXgq43odFFthboEdKNT/enyu+VvUGoIJ6cN8C/1B6o1WlYHEaL0BEIFFbAiAhZ/vnf\
cUYMaVPqsDJuETsjetcE32kGCD7Jkume2tO68DlIhB/2Z2JX8mkcbxFI6KrmXiRxXQj9\
9LVn1fEzdf3Vfpcs/C3omsFGqmTpLDK+AvW/SWVkDi2NKq7hL/AyxlW2u2cqVErQZUTS\
Z+ic6V8kZfxr3gRMnH0KuF5BtjleZ/yVvqqPjwPOZegCKEl2Gd8duhcUde7CR55pil1o\
UXy5AwgCcZTdEcJn1OPObGoots9T19gw1x4vnZCQUKVDPZuZ1gIkGqDUYXS0lcNTjCMs\
miFEmnOZvB88jxULpb1vl9HoQ3ocM2oZu4AZRt9G/L07Mwcui0uFCWtAIau+2gqNAn/Z\
AS10l0j2N0LLtAaOxoF+Ctzscrt0ZMyGHmoQ9daHkpUvEq0cO8hDtLplnq3lQIIIfROQ\
jcNs9vNKBu87COBjukZD+L8vV4zy8FNO59MCSb9UCLwz2xvfdI1js9/J7hTGaVec8VPx\
md42yPFrGw5Na1oefm8vW49EDmevc8AjAtwDirRBDFv9pX3+5S+M6jhteSLYvpKJXQT1\
zs1379KvIHwkn9VHpA+PiUUw9TgF6xF8xWEGSNlOo1Vn1xtM3givehjYxJ5p5/kBEFZI\
DCyFzstAirJ2GadNhae+P1JFZzJWnX5jaLwzldquZwF3yTzNho4sgBA+fKqiXcgn2nw1\
vz0Dkbxr6cMaUool0eFScU1nAz1Z39W64LtT2nEuYsORx/ht2RzJxxFc21X3nLeEDFCe\
NkNDxQFBSfpZjKKgJtXEx23mp+CbBVMrbagsLnzsAGLYbnroVmATU5Iqr6LgYBpuFs+N\
Rkq7ZXh6CZPukMGQbcOGuNwO6NBuuMNhir5ayGk1ZBiW82C7Nu0hs2pLcgNqWMtt1+LW\
8R96KyoSc784ZYAZ40QqvoySwmxQPBRTRJ+wB0sVpGBLTxdY9Gw3pXeXN5nao340d2ZA\
7YEMlqcTHCAv3F8B9ewl7OfQlmg6bvdMuoVdVE+p0er7IAmWMRgviIzYv9sKEEQrCmua\
2qL5xPSbD05KRf8ZAZ2B8lSCDR1nzXrQXZbXBKJivsCVQDuzxrwGE0gqRMpbk4f5GYCG\
4i/O8Knoru+jjf6wVQDYKfyz1QUGRlXHkGUGlXfv03r7UbJugycjVO5kbGxhoZkqOq8z\
ZEpkefvrrNoxeotw/z4QpjI8JlY97GDb0mGVHbmdHugjMtVTGhVJFBbPIinmR+emt7O+\
4qOr7ywRxCvt2lziWtpPBwaf/1XDnN5Gesex1gR1YrcTRNmB808b01sxLQmxcTt4eQ0/\
LUkas7qTJ3AQThOfDdtIpkqsthsBFy+WjSQuoXCYMRcPi6MlpxJndDF32lCnL1ranV6e\
F2ST0SYT+NwNDesMzTRmNbHUW5KAhu0k9WABTvcM5ba0Uq6iOa1NsFrcLag+KhxN6HPn\
oobwJ/EsDi5S7TAl8WrjqIhZ8x6h9eRRXerpaOw/FYk+2MpWByp/98VE12/EwOqAIiPp\
elAvUeMOlRkpG64bJsmyYtHuNWgcv5Qiy7/eGw9ZpvB3J3G3jxvbynExqdFyDc067EKi\
5WxDFPuZUjkfKpekNvzQuIrqs49BzcRyMt5ndEVE21TPPfZ/R8B7Rxnb2LiK+hQc+cc9\
pEEaWgwAOiMILcp/1CyY6ImdO6RHsxwflMH7gej+hN41kaoEghIOl9kMGTLZbq5Pc8Pz\
6F2LKTBMJWg9o/0blvilMH9EPblcLeF/bR1AZTUD6ZFdi2TxN6Epn3QVqeG/qPm1EBTF\
Gw1V92m6/08Dd6zI1HPqwKbkHx4F567owofKHaM2imin0yVUpwxoRJrulRHMCB3tn8C4\
ZpFl+sGV3Gip3tKlS7PKQkTqI6DMwxEbdrvtdY1sHZagpclLDisA/yFT4RR2m3VNJR9P\
6Nx3teqN1eg6RXmD/MlKCdWrlcjZ/6yeIQYwbr9CjItY/tLQX2gtAR1SXOh99UUBVv+Z\
E03VOZ+Ecsc78lSB9G/6n6CFzlbk/HgAF+cu0yMbGnEM8W3mTUspS4JBACwk5w0XWNNQ\
DWVEdgzuLGhPq+hYExDjVZrLELhkH8YgZA+7RXXUZHM/joNOGHUhpUG/bFo3ktnaILCu\
xsOXMUbDC3VcitFFHsGK1svtcERDFxk1HA8pGa59jT0do6n3wEbnBDU1soKNFtpmcVkE\
Ul3XpvuoW3BgCwJzBUCWvPs47DJRgGxO11bSaEYYlhTVaaShcvzgz46AkqO+Q7TjckDP\
/8uzsSQk0AbuhxWFQpSiBP8OZ/U=",
  "d": "z7u7GwhsjjnfHH3Nkrs2xvvw020Rcw5ymdlTnhRenjDUBgL6FklHURz5btM5\
yrI5FQdWk+U2srVuSmfDV7EYG897mUFY35Z0WQ0mZ9XvIOKCh+GFFOk56b5FOFq6xnV8\
UDQnFyY2JREUOHdiUjcUNxA1YxR3QiQ0BkE1AUBmFEOAUHZGBzQAU2dxVIgTQRV3U3g4\
GGiISEYQhHRSWDIBQ2Z3UIIWdSV1EWhwBTYiWGI3VmJVI1UIU2REdUhHBoJ2gRhFUThy\
BSQnhBIGI1AoMVB2MCNhUXQiNUGCKHgzUmQxU3dEgBhmQyIQgmFjdxY1dCJgGBSEB4Ij\
```

CEJ0MBGIQWRRN3QjRmRSQWQIJgNjcjdnMlJhJIU1MlJRd1NmF4dwhHIIdEYYcAhEclBQ\
JjESAiBwBQYzYlAIIocBcoZFcGVkA2SDMCVTBjgzCAAzNnQGYHI1VwJzYxQRckBIZBV4\
VxZmZiVlYXgHFRNjdEFIYVOFIVdhcnIINEhhIURjg0cxJ0SCIWYUUVcHJzdDUTASciAW\
UiJAglIoQkIwNjMlcwACZxcHZVJAh4EnNnWAZVVoBjNnNEcicTdyEEUHBVFjETETNjd0\
YUFmFDVXNUcHFVJoE2AlVwFhMzc0dQckMYJUaBJ0JkUBdyd1AnZiJ3hYYkWAgyR1VziD\
BERVh1NQAFIGWIhUZXCEQxIThyR1FIGGNTKAcwdRNBGDYEd2RVEwUDMzWAAhNQEEMYAS\
hUSCgTJ3VIdXUlFBFxFkhVCCZEABJjQCAxFGNkhHQzM1YSSHNlEBEmJkgWZCVwZHRSVD\
GDZzRCQiNhE3IghDhSFoBYCHNFVHMxZAZTSGAVMUQkhBKIFRQEVBB0cHNGEiJVVScQQh\
hzQiBzKBYRY4Q0R2MVUldwVCIkQYEEgFYEREY2NERyFVdHJzdAZHBmhmdSCFIgh1IwGB\
AxNzFjEoUWFCF4AhZRJSEROEEThxAGYBJTgUcUdFJBOFRmcVYnZUUFCAY1AnZEZBVThS\
ECBQYBNGeAdzQ3KGhIE3ZiFxQoVCgQBjEFdFcIV0IAaFcgI0iAgAUVQAhEInQWECY1I4\
E2U0MkAXQSZkdSRRc2I4BjEiSGd4hgQEEDcTRmhFd3MBMmY2gERxNwiISAVkFVETGCcI\
gYhzRXByGBgzKGVXJUhoGEY1hgFmZWgmEWYWVoISd4Vlhid4VUMHgSZXhXUSaCgmJ3Eg\
MQIzIDIThwRSARQhBFB2QQBjEgIoEHN1BhMCiIIBUlZWCHdBMyZFQoQ2UUJXJCFnZyFD\
RTFUUTQoQmUjB0aHJXJHeDZlJGBCExATEUEDg3BTAChWImN0AWcFB3IUgBEARxVUREdX\
QzhVBEBBF4UgcRhCg4gUcoRkdYCAIUQBRUJUYjFjEBYUhSBjIyV2cXBYckEiMic1N4ED\
gDUGVSCHhCYURXcQB1ODg4hDJFJ3Jld2gyYnQYclRjE3VWcQNXJBYnOGhYFoU2dHATAw\
OEeBUGQjJHcDgUJTRXBXJ3IAEjEXhXeEEmghIAAGdjUVBndnI2FCAVcyV4cxIyZ2dYE1\
ZEiHcYJYaCcXQXJCaER1coIDRWJkcSNhEVF3JGOCQkYWYkcndRA1QTh0MFgzIyVocYJY\
cwIAFRNiE1VAUECBI3MzQiZkUhR1cid1NSAXFXN0gUNnRCV0ISNmYnB3NIZyMIQWEVVz\
ElEohnQyKBNoY1cCdmdjVYiGhVUlA0CHMTZIURBgR4aIJwJQJlIDMYhwNGNwiGcIBUdA\
NXMBETUgICJyMkMIN1BAIAB4gEMDUwhndFJkQDEgRRMld4EzhRM0ZhGAYHMgZ4NhEEhn\
U2I2VicBBXZUFUNoczcmBzBnUEBxUWcDg4RHUXZSOEZogABHRAISVEAzdoQhRmBgEWYE\
Z4U1dgQ2gjgQUjMScgIIFQR3YFczhTYoB1IiVCYGBAVmVAFIdhRmZ1InhwdTRjJjNhVx\
IzcWgjFlFCaChlIWUySIaDFwAWV3RAIxg2QWYgAYMUjP7wmwOwPp7Ukl3L1KalY/6dN4\
dBr1AYS8JnkVq6pPeBfO7ccX95SrVfAO7EX7RVEYyhVR9QOQyEpLBUMcfcfnHCZWKM0o\
OBF7BXiWMR9BQo4ybtpJGKQ+IZyCKUJVRhZ+uae182qYcBKFMdOOzXiO8kAa98eUy6SR\
pPfKPD6D+xXgtJ0FWtYnp1Jy2aIG3HqMiTHoSdVIvccGkf94gpVWTMeJQsQpgq7dAJiJ\
5JOMQjk7JIHcIzxb4T8sQHzA55MFfvM7Hus/8FUX7NfIN1JRmc2zHL/7kdfCFSwG67iW\
U4ob2kTwdKzPvOL+d3e+AOE0PihJ4vVJAOjhWmO2fIFNvFhNqPh0MSiSkatPGbSVdqQ1\
PsG6C+1YqMrTM7KFr4hTQM8a3+tAOsImMjXSSPDkVeuJFq1rw642SJJx8yZTXVe8g75D\
ZTYghbeX5LLzaVkt9mZS7cW16Zy+C3MwnWDrGQ6hUDxYaYJp7SOGJHepcmVV214oD6nw\
5QprgpGIxVcdXQUO0fhKwerYDkoOIj+uqk7NYDvOt8zANphYcE3v+6yVFyYh3eg7DYRJ\
rIzIcbaG91ySv2iRRC+cWaymH6xuqaHRwZu/p962/u8/c3rITJzCoVc+ObnZ5oItZFBe\
AYFhLBx7PvPdBULXyCqmtkOtnT/jnaCUVxtGeaIeQmmeM4yPq3d5uWBfOvIyuPmfBSKd\
Y0NETGlsaoQuqFpOkCmQdMVZKh3UZ8AOjw22LlqaZlrUf0akb0fs7le2HT47KV9yOJHC\
tec9tjHUeBVmma5O4AofGcVXLbkqKv+Soax9GooHVOv+uxa8iwjAdTZKtqwKnKDx4jaR\
+zotCsYi4BuB2JbkjnHG6NL7ubN+aNKnwnzZnMKQZIh2Q7vSRYKTM8j9OGLq7IP8q2NS\
oc7iT//eAvb4oF6LaY7qebxQ6ROXCSRrrXgpo+pw3ltfuUCuGzAxD4+wMZU3dlXsivhJ\
PnTEjI/V6GmkRlfZ9XnYfj8SILETWk03dMFJh3LmUwkbRV+C3mL2GzjgQVTkvP82KDBL\
DAR9iKyPkJnMnK9Ix/StVyJbGAtGp4jHnp+PSjz9ja4qI9jVRjGgIUQhw0DnI0fnplUn\
Qhz3F9MQXMPLSPvFw8M0xkUKsAcxQvxZGb5LkYByZ0ZrO/ipphwnE6zQuOva+8uTyBX/\
B9VR24tUItvlhy7SS6JrULrvTA+D/ZCiqKRx61iF6pU3BoC8fgA9D/AifiQnPz0SI5kx\
FJfDTz1LWMjUlQKBHFvRFLE9eFD0rnwAGx7Pgpyc/KrLqVmcmj/96TYtoedp/iW4asfY\
C2vs+GVyxVoumIdFPHJpencWbE/niZnVDaJCih1iqgXzDsI8bENh2B9cutDWX+bsHZSC\
jSQb9YkGN+MoNiJlXmQHSJDyfPhzWPibdS/lpS90ppPWIY+PpLOfzDSGFFWswQ4q5Phc\
pLWHx5lw9KSye+T86p6kadnBBTLTyfn0dG7NpO9QKQObMN60MnybkVGx5nH9yLJlFlmV\
0H+K0VZIKm4UzYV+RYfqqXYtMqTQxeQ1U7L7o0H+6viErxuKj5rS3i+r1rdfECAGgCoq\
0mixATHISAHi2eSV5fk3r5xMkKSwwPIRuMt50+kklRPUoLohTj7G1CnL6O2xwBdQMTUx\

```
4Jq5JBWnfB+U4D9n0si1DwikIhpaUyOoBeaWo4iFQiWVLwjeeQvY6zj66l7OXsPHjZXg\
uCitsWfp5MYV3cLTkb80uCM/xhp4Y0Edobt6x3k1FD8vbh8g3YAG0Xe/U+Iz3klnpCt2\
ROQ2lGQa0JMl4nbQr3tqTLoXv4szaErfP/Xw05Cnt9DsBzN5DNrmfF6EDcfVf/hn8v9a\
wrg6Rfv8Jpys1YFpwLanhb3Wz+x1yaDsa54IdlFOFnyBxv8GppbFrMpVFx/nLAXGIocc\
WjcRKs0tBJUW/IoXeKOMPUd1wHR4dqUCEXsoexHJiNe5sH+akr6UIDObF70hhupBoiY9\
AzVXi5zXf2VdafyQrkGfKz4BEUkiqcaajHr1CF9ZJ+Mjdmfr3z0xyCmCAWir5ZLOBXDj\
T7sYCV3QjCz4a2mGvee9IxC9kSLapCq90UMAxnTLjJGQM/dlpgjDsjsCZX5wKdsnMs79\
60Z75BGDOC1dDINj4f5kHZmwwcmw/04mi/1RPBUABXse3Up3eJQOX2haZPqmY0+2PZTF\
exku9pETHtKcfSdRe1oJLmlB34JSogRmNp1eBxakcIL09huiFVtGVZng/pC/ryoJ/T9q\
9w4aV5H+4u2dHc29Vb77SasxCdRH0sDaLaPpesRXsrdJwjbizOgzRlIx+83oO7NuhE+C\
kKfO7cZMrFm8r8g1MlzDiFrTf3RTusMtiW6CVlVuTROPZFngqaR5yeYPprpSELtQHSwz\
U5AaY5Qd8tbky5ec+2/QkXO+cdyWhQUuBRpibwpRpD3x1yTgT4E91cwTFpvSLk54ZHf+\
D3EsZf0PYMN6d4jVdh9iv+0tCebnfMqP65wY26YBopSLtCXXb1anUlRPlzPzRq99yKnt\
FM7gK1XnBAZoZBBqCyZw9OHWmttIFWcml4Wd5BxF9uZh2Y8gtcN8UKWHv43tsNBa7j/T\
ikIBSkIVI/6EQvyPW4YTdyz2V8RKHN5XcdpdWFaVhgSJMC4I6Bm0Lwenhkmal7Sd247q\
uCtEow8qh+w7Jk4SxrmvJxd5sBnvz15OKEaHPeWNNJW00bWEDT+0ZzzD8vMN1/GkbbB3\
s7UfcJXZbRu7HtQ+wHIblBKVstX3hMonra+k6wS9KPhcAaC3IjZ7ZApSedKk1sW1SuDg\
l48YW2/cyS3LvmISQn9KPWK7yEpNQnV0vurn3ZFOGO0eDjSXUjI+xIrRia5GQ1yb31ma\
nJnf2PdHcMmVr0wu4lMGno7a14nMRdnXkBU8bVOp8wF6Toz59hBJ3a/F+mP4/a19Ixra\
wiVVeEPgoi9QQ9NcLgQEFCoskA+EpcLK0FxV2rYI9JFNF/nDxP5nmGtnkmlFaLo+pleH\
CJYS0OTGKQr6X+Y65NOllx5nNwsnWkIUkCodoSt4Givdoe/S9JNIu8tW+jTBae2hNr9c\
glErCNKDYe1+T+Ldyr9rfOKm9LKNyTBsodgF4KI/hFh9Iv/i55DTWtqjpN0eQnPTB3/6\
+7KzTfSE9il5UMcP3zKKC2mAQvtyYxF3k0m24ZTwPs2LAPJkr/xtPH3BnGE/UfUDmvDS\
TBp9m049Nh9oDZvI4HKsY8auiyENk0ys67F9GTHhOYM0FgHyP5qk4/IR5YC3lnq7xx6i\
owebEJAy63htMytq+xd3cJyZR0lWBUOqvSpd/A=="
}
```


Example private key using optional fields:

```
=============== NOTE: '\' line wrapping per RFC 8792 ================

{
  "kid": "key-0",
  "kty": "PQK",
  "alg": "CRYDI3",
  "key_ops": ["sign"],
  "xs": "z3uZQVjflnRZDSZn1e8g4oKH4YUU6TnpvkU4WrrGdXw=",
  "ds": "5DuZ8XoJQirc/5TE23tBcoGoHo+JTj1+9ULLXtCiySU=",
  "x": "z7u7GwhsjjnfHH3Nkrs2xvvw020Rcw5ymdlTnhRenjDdrOO+nfXRVUZVy9q1\
5zDn77zTgrIskM3WX8bqslc+B1fq12iA/wxD2jc1d6j+YjKCtkGH26OR7vc0YC2ZiMzW\
zGl7yebt7JkmjRbN1N+u/2fAKFLuziMcLNP6WLoWbMqxoC2XOOVNAWX3QjXrCcGU23Nr\
imtdmWz5NrP43E592Sctt5M+SVlfgQeYv8pHmtkQknE8/jr7TrgNpuiV7nXmhWHTMJ4I\
zoGXgq43odFFthboEdKNT/enyu+VvUGoIJ6cN8C/1B6o1WlYHEaL0BEIFFbAiAhZ/vnf\
cUYMaVPqsDJuETsjetcE32kGCD7Jkume2tO68DlIhB/2Z2JX8mkcbxFI6KrmXiRxXQj9\
9LVn1fEzdf3Vfpcs/C3omsFGqmTpLDK+AvW/SWVkDi2NKq7hL/AyxlW2u2cqVErQZUTS\
Z+ic6V8kZfxr3gRMnH0KuF5BtjleZ/yVvqqPjwPOZegCKEl2Gd8duhcUde7CR55pil1o\
UXy5AwgCcZTdEcJn1OPObGoots9T19gw1x4vnZCQUKVDPZuZ1gIkGqDUYXS0lcNTjCMs\
miFEmnOZvB88jxULpb1vl9HoQ3ocM2oZu4AZRt9G/L07Mwcui0uFCWtAIau+2gqNAn/Z\
AS10l0j2N0LLtAaOxoF+Ctzscrt0ZMyGHmoQ9daHkpUvEq0cO8hDtLplnq3lQIIIfROQ\
jcNs9vNKBu87COBjukZD+L8vV4zy8FNO59MCSb9UCLwz2xvfdI1js9/J7hTGaVec8VPx\
md42yPFrGw5Na1oefm8vW49EDmevc8AjAtwDirRBDFv9pX3+5S+M6jhteSLYvpKJXQT1\
zs1379KvIHwkn9VHpA+PiUUw9TgF6xF8xWEGSNlOo1Vn1xtM3givehjYxJ5p5/kBEFZI\
DCyFzstAirJ2GadNhae+P1JFZzJWnX5jaLwzldquZwF3yTzNho4sgBA+fKqiXcgn2nw1\
vz0Dkbxr6cMaUool0eFScU1nAz1Z39W64LtT2nEuYsORx/ht2RzJxxFc21X3nLeEDFCe\
NkNDxQFBSfpZjKKgJtXEx23mp+CbBVMrbagsLnzsAGLYbnroVmATU5Iqr6LgYBpuFs+N\
Rkq7ZXh6CZPukMGQbcOGuNwO6NBuuMNhir5ayGk1ZBiW82C7Nu0hs2pLcgNqWMtt1+LW\
8R96KyoSc784ZYAZ40QqvoySwmxQPBRTRJ+wB0sVpGBLTxdY9Gw3pXeXN5nao340d2ZA\
7YEMlqcTHCAv3F8B9ewl7OfQlmg6bvdMuoVdVE+p0er7IAmWMRgviIzYv9sKEEQrCmua\
2qL5xPSbD05KRf8ZAZ2B8lSCDR1nzXrQXZbXBKJivsCVQDuzxrwGE0gqRMpbk4f5GYCG\
4i/O8Knoru+jjf6wVQDYKfyz1QUGRlXHkGUGlXfv03r7UbJugycjVO5kbGxhoZkqOq8z\
ZEpkefvrrNoxeotw/z4QpjI8JlY97GDb0mGVHbmdHugjMtVTGhVJFBbPIinmR+emt7O+\
4qOr7ywRxCvt2lziWtpPBwaf/1XDnN5Gesex1gR1YrcTRNmB808b01sxLQmxcTt4eQ0/\
LUkas7qTJ3AQThOfDdtIpkqsthsBFy+WjSQuoXCYMRcPi6MlpxJndDF32lCnL1ranV6e\
F2ST0SYT+NwNDesMzTRmNbHUW5KAhu0k9WABTvcM5ba0Uq6iOa1NsFrcLag+KhxN6HPn\
oobwJ/EsDi5S7TAl8WrjqIhZ8x6h9eRRXerpaOw/FYk+2MpWByp/98VE12/EwOqAIiPp\
elAvUeMOlRkpG64bJsmyYtHuNWgcv5Qiy7/eGw9ZpvB3J3G3jxvbynExqdFyDc067EKi\
5WxDFPuZUjkfKpekNvzQuIrqs49BzcRyMt5ndEVE21TPPfZ/R8B7Rxnb2LiK+hQc+cc9\
pEEaWgwAOiMILcp/1CyY6ImdO6RHsxwflMH7gej+hN41kaoEghIOl9kMGTLZbq5Pc8Pz\
6F2LKTBMJWg9o/0blvilMH9EPblcLeF/bR1AZTUD6ZFdi2TxN6Epn3QVqeG/qPm1EBTF\
Gw1V92m6/08Dd6zI1HPqwKbkHx4F567owofKHaM2imin0yVUpwxoRJrulRHMCB3tn8C4\
ZpFl+sGV3Gip3tKlS7PKQkTqI6DMwxEbdrvtdY1sHZagpclLDisA/yFT4RR2m3VNJR9P\
6Nx3teqN1eg6RXmD/MlKCdWrlcjZ/6yeIQYwbr9CjItY/tLQX2gtAR1SXOh99UUBVv+Z\
E03VOZ+Ecsc78lSB9G/6n6CFzlbk/HgAF+cu0yMbGnEM8W3mTUspS4JBACwk5w0XWNNQ\
DWVEdgzuLGhPq+hYExDjVZrLELhkH8YgZA+7RXXUZHM/joNOGHUhpUG/bFo3ktnaILCu\
xsOXMUbDC3VcitFFHsGK1svtcERDFxk1HA8pGa59jT0do6n3wEbnBDU1soKNFtpmcVkE\
Ul3XpvuoW3BgCwJzBUCWvPs47DJRgGxO11bSaEYYlhTVaaShcvzgz46AkqO+Q7TjckDP\
/8uzsSQk0AbuhxWFQpSiBP8OZ/U=",
  "d": "z7u7GwhsjjnfHH3Nkrs2xvvw020Rcw5ymdlTnhRenjDUBgL6FklHURz5btM5\
```

yrI5FQdWk+U2srVuSmfDV7EYG897mUFY35Z0WQ0mZ9XvIOKCh+GFFOk56b5FOFq6xnV8\
UDQnFyY2JREUOHdiUjcUNxA1YxR3QiQ0BkE1AUBmFEOAUHZGBzQAU2dxVIgTQRV3U3g4\
GGiISEYQhHRSWDIBQ2Z3UIIWdSV1EWhwBTYiWGI3VmJVI1UIU2REdUhHBoJ2gRhFUThy\
BSQnhBIGI1AoMVB2MCNhUXQiNUGCKHgzUmQxU3dEgBhmQyIQgmFjdxY1dCJgGBSEB4Ij\
CEJ0MBGIQWRRN3QjRmRSQWQIJgNjcjdnMlJhJIU1MlJRd1NmF4dwhHIIdEYYcAhEclBQ\
JjESAiBwBQYzYlAIIocBcoZFcGVkA2SDMCVTBjgzCAAzNnQGYHI1VwJzYxQRckBIZBV4\
VxZmZiVlYXgHFRNjdEFIYVOFIVdhcnIINEhhIURjg0cxJ0SCIWYUUVcHJzdDUTASciAW\
UiJAglIoQkIwNjMlcwACZxcHZVJAh4EnNnWAZVVoBjNnNEcicTdyEEUHBVFjETETNjd0\
YUFmFDVXNUcHFVJoE2AlVwFhMzc0dQckMYJUaBJ0JkUBdyd1AnZiJ3hYYkWAgyR1VziD\
BERVh1NQAFIGWIhUZXCEQxIThyR1FIGGNTKAcwdRNBGDYEd2RVEwUDMzWAAhNQEEMYAS\
hUSCgTJ3VIdXUlFBFxFkhVCCZEABJjQCAxFGNkhHQzM1YSSHNlEBEmJkgWZCVwZHRSVD\
GDZzRCQiNhE3IghDhSFoBYCHNFVHMxZAZTSGAVMUQkhBKIFRQEVBB0cHNGEiJVVScQQh\
hzQiBzKBYRY4Q0R2MVUldwVCIkQYEEgFYEREY2NERyFVdHJzdAZHBmhmdSCFIgh1IwGB\
AxNzFjEoUWFCF4AhZRJSEROEEThxAGYBJTgUcUdFJBOFRmcVYnZUUFCAY1AnZEZBVThS\
ECBQYBNGeAdzQ3KGhIE3ZiFxQoVCgQBjEFdFcIV0IAaFcgI0iAgAUVQAhEInQWECY1I4\
E2U0MkAXQSZkdSRRc2I4BjEiSGd4hgQEEDcTRmhFd3MBMmY2gERxNwiISAVkFVETGCcI\
gYhzRXByGBgzKGVXJUhoGEY1hgFmZWgmEWYWVoISd4Vlhid4VUMHgSZXhXUSaCgmJ3Eg\
MQIzIDIThwRSARQhBFB2QQBjEgIoEHN1BhMCiIIBUlZWCHdBMyZFQoQ2UUJXJCFnZyFD\
RTFUUTQoQmUjB0aHJXJHeDZlJGBCExATEUEDg3BTAChWImN0AWcFB3IUgBEARxVUREdX\
QzhVBEBBF4UgcRhCg4gUcoRkdYCAIUQBRUJUYjFjEBYUhSBjIyV2cXBYckEiMic1N4ED\
gDUGVSCHhCYURXcQB1ODg4hDJFJ3Jld2gyYnQYclRjE3VWcQNXJBYnOGhYFoU2dHATAw\
OEeBUGQjJHcDgUJTRXBXJ3IAEjEXhXeEEmghIAAGdjUVBndnI2FCAVcyV4cxIyZ2dYE1\
ZEiHcYJYaCcXQXJCaER1coIDRWJkcSNhEVF3JGOCQkYWYkcndRA1QTh0MFgzIyVocYJY\
cwIAFRNiE1VAUECBI3MzQiZkUhR1cid1NSAXFXN0gUNnRCV0ISNmYnB3NIZyMIQWEVVz\
ElEohnQyKBNoY1cCdmdjVYiGhVUlA0CHMTZIURBgR4aIJwJQJlIDMYhwNGNwiGcIBUdA\
NXMBETUgICJyMkMIN1BAIAB4gEMDUwhndFJkQDEgRRMld4EzhRM0ZhGAYHMgZ4NhEEhn\
U2I2VicBBXZUFUNoczcmBzBnUEBxUWcDg4RHUXZSOEZogABHRAISVEAzdoQhRmBgEWYE\
Z4U1dgQ2gjgQUjMScgIIFQR3YFczhTYoB1IiVCYGBAVmVAFIdhRmZ1InhwdTRjJjNhVx\
IzcWgjFlFCaChlIWUySIaDFwAWV3RAIxg2QWYgAYMUjP7wmwOwPp7Ukl3L1KalY/6dN4\
dBr1AYS8JnkVq6pPeBfO7ccX95SrVfAO7EX7RVEYyhVR9QOQyEpLBUMcfcfnHCZWKM0o\
OBF7BXiWMR9BQo4ybtpJGKQ+IZyCKUJVRhZ+uae182qYcBKFMdOOzXiO8kAa98eUy6SR\
pPfKPD6D+xXgtJ0FWtYnp1Jy2aIG3HqMiTHoSdVIvccGkf94gpVWTMeJQsQpgq7dAJiJ\
5JOMQjk7JIHcIzxb4T8sQHzA55MFfvM7Hus/8FUX7NfIN1JRmc2zHL/7kdfCFSwG67iW\
U4ob2kTwdKzPvOL+d3e+AOE0PihJ4vVJAOjhWmO2fIFNvFhNqPh0MSiSkatPGbSVdqQ1\
PsG6C+1YqMrTM7KFr4hTQM8a3+tAOsImMjXSSPDkVeuJFq1rw642SJJx8yZTXVe8g75D\
ZTYghbeX5LLzaVkt9mZS7cW16Zy+C3MwnWDrGQ6hUDxYaYJp7SOGJHepcmVV214oD6nw\
5QprgpGIxVcdXQUO0fhKwerYDkoOIj+uqk7NYDvOt8zANphYcE3v+6yVFyYh3eg7DYRJ\
rIzIcbaG91ySv2iRRC+cWaymH6xuqaHRwZu/p962/u8/c3rITJzCoVc+ObnZ5oItZFBe\
AYFhLBx7PvPdBULXyCqmtkOtnT/jnaCUVxtGeaIeQmmeM4yPq3d5uWBfOvIyuPmfBSKd\
Y0NETGlsaoQuqFpOkCmQdMVZKh3UZ8AOjw22LlqaZlrUf0akb0fs7le2HT47KV9yOJHC\
tec9tjHUeBVmma5O4AofGcVXLbkqKv+Soax9GooHVOv+uxa8iwjAdTZKtqwKnKDx4jaR\
+zotCsYi4BuB2JbkjnHG6NL7ubN+aNKnwnzZnMKQZIh2Q7vSRYKTM8j9OGLq7IP8q2NS\
oc7iT//eAvb4oF6LaY7qebxQ6ROXCSRrrXgpo+pw3ltfuUCuGzAxD4+wMZU3dlXsivhJ\
PnTEjI/V6GmkRlfZ9XnYfj8SILETWk03dMFJh3LmUwkbRV+C3mL2GzjgQVTkvP82KDBL\
DAR9iKyPkJnMnK9Ix/StVyJbGAtGp4jHnp+PSjz9ja4qI9jVRjGgIUQhw0DnI0fnplUn\
Qhz3F9MQXMPLSPvFw8M0xkUKsAcxQvxZGb5LkYByZ0ZrO/ipphwnE6zQuOva+8uTyBX/\
B9VR24tUItvlhy7SS6JrULrvTA+D/ZCiqKRx61iF6pU3BoC8fgA9D/AifiQnPz0SI5kx\
FJfDTz1LWMjUlQKBHFvRFLE9eFD0rnwAGx7Pgpyc/KrLqVmcmj/96TYtoedp/iW4asfY\
C2vs+GVyxVoumIdFPHJpencWbE/niZnVDaJCih1iqgXzDsI8bENh2B9cutDWX+bsHZSC\

```
jSQb9YkGN+MoNiJlXmQHSJDyfPhzWPibdS/lpS90ppPWIY+PpLOfzDSGFFWswQ4q5Phc\
pLWHx5lw9KSye+T86p6kadnBBTLTyfn0dG7NpO9QKQObMN60MnybkVGx5nH9yLJlFlmV\
0H+K0VZIKm4UzYV+RYfqqXYtMqTQxeQ1U7L7o0H+6viErxuKj5rS3i+r1rdfECAGgCoq\
0mixATHISAHi2eSV5fk3r5xMkKSwwPIRuMt50+kklRPUoLohTj7G1CnL6O2xwBdQMTUx\
4Jq5JBWnfB+U4D9n0si1DwikIhpaUyOoBeaWo4iFQiWVLwjeeQvY6zj66l7OXsPHjZXg\
uCitsWfp5MYV3cLTkb80uCM/xhp4Y0Edobt6x3k1FD8vbh8g3YAG0Xe/U+Iz3klnpCt2\
ROQ2lGQa0JMl4nbQr3tqTLoXv4szaErfP/Xw05Cnt9DsBzN5DNrmfF6EDcfVf/hn8v9a\
wrg6Rfv8Jpys1YFpwLanhb3Wz+x1yaDsa54IdlFOFnyBxv8GppbFrMpVFx/nLAXGIocc\
WjcRKs0tBJUW/IoXeKOMPUd1wHR4dqUCEXsoexHJiNe5sH+akr6UIDObF70hhupBoiY9\
AzVXi5zXf2VdafyQrkGfKz4BEUkiqcaajHr1CF9ZJ+Mjdmfr3z0xyCmCAWir5ZLOBXDj\
T7sYCV3QjCz4a2mGvee9IxC9kSLapCq90UMAxnTLjJGQM/dlpgjDsjsCZX5wKdsnMs79\
60Z75BGDOC1dDINj4f5kHZmwwcmw/04mi/1RPBUABXse3Up3eJQOX2haZPqmY0+2PZTF\
exku9pETHtKcfSdRe1oJLmlB34JSogRmNp1eBxakcIL09huiFVtGVZng/pC/ryoJ/T9q\
9w4aV5H+4u2dHc29Vb77SasxCdRH0sDaLaPpesRXsrdJwjbizOgzRlIx+83oO7NuhE+C\
kKfO7cZMrFm8r8g1MlzDiFrTf3RTusMtiW6CVlVuTROPZFngqaR5yeYPprpSELtQHSwz\
U5AaY5Qd8tbky5ec+2/QkXO+cdyWhQUuBRpibwpRpD3x1yTgT4E91cwTFpvSLk54ZHf+\
D3EsZf0PYMN6d4jVdh9iv+0tCebnfMqP65wY26YBopSLtCXXb1anUlRPlzPzRq99yKnt\
FM7gK1XnBAZoZBBqCyZw9OHWmttIFWcml4Wd5BxF9uZh2Y8gtcN8UKWHv43tsNBa7j/T\
ikIBSkIVI/6EQvyPW4YTdyz2V8RKHN5XcdpdWFaVhgSJMC4I6Bm0Lwenhkmal7Sd247q\
uCtEow8qh+w7Jk4SxrmvJxd5sBnvz15OKEaHPeWNNJW00bWEDT+0ZzzD8vMN1/GkbbB3\
s7UfcJXZbRu7HtQ+wHIblBKVstX3hMonra+k6wS9KPhcAaC3IjZ7ZApSedKk1sW1SuDg\
l48YW2/cyS3LvmISQn9KPWK7yEpNQnV0vurn3ZFOGO0eDjSXUjI+xIrRia5GQ1yb31ma\
nJnf2PdHcMmVr0wu4lMGno7a14nMRdnXkBU8bVOp8wF6Toz59hBJ3a/F+mP4/a19Ixra\
wiVVeEPgoi9QQ9NcLgQEFCoskA+EpcLK0FxV2rYI9JFNF/nDxP5nmGtnkmlFaLo+pleH\
CJYS0OTGKQr6X+Y65NOllx5nNwsnWkIUkCodoSt4Givdoe/S9JNIu8tW+jTBae2hNr9c\
glErCNKDYe1+T+Ldyr9rfOKm9LKNyTBsodgF4KI/hFh9Iv/i55DTWtqjpN0eQnPTB3/6\
+7KzTfSE9il5UMcP3zKKC2mAQvtyYxF3k0m24ZTwPs2LAPJkr/xtPH3BnGE/UfUDmvDS\
TBp9m049Nh9oDZvI4HKsY8auiyENk0ys67F9GTHhOYM0FgHyP5qk4/IR5YC3lnq7xx6i\
owebEJAy63htMytq+xd3cJyZR0lWBUOqvSpd/A=="
}
```


3.4.3. CRYDI Signature Representation

For the purpose of using the CRYSTALS-Dilithium Signature Algorithm (CRYDI) for signing data using "JSON Web Signature (JWS)" [RFC7515], algorithm "CRYDI" is defined here, to be applied as the value of the "alg" parameter.

The following key subtypes are defined here for use with CRYDI:

"pset"	CRYDI Paramter Set
5	CRYDI5
3	CRYDI3
2	CRYDI2

Table 4

The key type used with these keys is "PQK" and the algorithm used for signing is "CRYDI". These subtypes MUST NOT be used for key agreement.

The CRYDI variant used is determined by the subtype of the key (CRYDI3 for "pset 3" and CRYDI2 for "pset 2").

Implementations need to check that the key type is "PQK" for JOSE and that the pset of the key is a valid subtype when creating a signature.

The CRYDI digital signature is generated as follows:

1. Generate a digital signature of the JWS Signing Input using CRYDI with the desired private key, as described in Section 3.2. The signature bit string is the concatenation of a bit packed representation of z and encodings of h and c in this order.

2. The resulting octet sequence is the JWS Signature.

When using a JWK for this algorithm, the following checks are made:

* The "kty" field MUST be present, and it MUST be "PQK" for JOSE.

* The "alg" field MUST be present, and it MUST represent the pset subtype.

* If the "key_ops" field is present, it MUST include "sign" when creating an CRYDI signature.

* If the "key_ops" field is present, it MUST include "verify" when verifying an CRYDI signature.

* If the JWK "use" field is present, its value MUST be "sig".

Example signature using only required fields, represented in compact form:

eyJhbGciOiJQUzM4NCIsImtpZCI6ImJpbGJvLmJhZ2dpbnNAaG9iYml0b24uZX
hhbXBsZSJ9
.
SXTigJlzIGEgZGFuZ2Vyb3VzIGJ1c2luZXNzLCBGcm9kbywgZ29pbmcgb3V0IH
lvdXIgZG9vci4gWW91IHN0ZXAgb250byB0aGUgcm9hZCwgYW5kIGlmIHlvdSBk
b24ndCBrZWVwIHlvdXIgZmVldCwgdGhlcmXigJlzIG5vIGtub3dpbmcgd2hlcm
UgeW91IG1pZ2h0IGJlIHN3ZXB0IG9mZiB0by4
.
cu22eBqkYDKgIlTpzDXGvaFfz6WGoz7fUDcfT0kkOy42miAh2qyBzk1xEsnk2I
pN6-tPid6VrklHkqsGqDqHCdP6O8TTB5dDDItllVo6_1OLPpcbUrhiUSMxbbXU
vdvWXzg-UD8biiReQFlfz28zGWVsdiNAUf8ZnyPEgVFn442ZdNqiVJRmBqrYRX
e8P_ijQ7p8Vdz0TTrxUeT3lm8d9shnr2lfJT8ImUjvAA2Xez2Mlp8cBE5awDzT
0qI0n6uiP1aCN_2_jLAeQTlqRHtfa64QQSUmFAAjVKPbByi7xho0uTOcbH510a
6GYmJUAfmWjwZ6oD4ifKo8DYM-X72Eaw

The same example decoded for readability:

```
=============== NOTE: '\\' line wrapping per RFC 8792 ===============

{
  "header": { "alg": "CRYDI3", "kid": "did:example:123#key-0" },
  "payload": "It's a dangerous business, Frodo, going out your door.\
\ You step onto the road, and if you don't keep your feet, there's\
\ no knowing where you might be swept off to.",
  "signature": "2As8T1AHenWzLuTojcAYFDnT05n4bmDGIWenHqoXVizL7311HtVg\
\7PEJHYmpc1fIvFNrm0xJt0asD5bQk3ZY8WuEQDUjsn4j+zbyob8MPQI5u3p5ZkqlLhG\
\6Q8p1q0Hd5voY4a78vNxFJpYsETc0bECAft196z5hml2VjuDBqI7W4ju/iDKambJIDz\
\NLYgYinNyPcHjlfBP7aCfOqGBAOQrWuVgrAkdeM+uH6djaXW25+FeUl4Lg1uOIBPrcj\
\ZJO4MO7j7BmiuHJDB74QG/ifVqnvr4z2alMWHjjR7nPPr2CIKpuRthSpNWYVTRSN3mM\
\v0GjVLyaqhJpmUmewhjaQCi3iP7c59yKatGYjLPPEapsbN7ypIo1Bod/R2PZR0zeool\
\d9k30VmGsVLkJ4OEIFnlA8epv+bJISApZWrGuU6NBP8vr4UB2D9DRd8zwvd/vI0BWdq\
\nfglX4x18lWe8Tnd+21UC9n4zUb+KQlo9RR14VfXEOt9g5aOIzCWjAN+Oz8vqJ/ZwgH\
\zZotZNF+nZehtFPcPLM3dpoUkEI391VH3QQ6VTYfbMW9wGJ6UnylxZFEzNCnMFF9Qhs\
\7Xehy4yEDgJBFYIvbTRCfD+EbZbWQAnLKsm7UXXBR7HdUsJMhTkwdffGziWJBTf1UsG\
\tqaCF1bvXgbcCSe0XGhc0QkQKwwj3kNWY9/hnhH1bn7kyySqaI+W4Ph3pKwRb38sCS/\
\Gb3ryptI8zez0JR+lClWnu18noJjGincZq7jCGMiMCRFpzUV6rpY/FiM26IpZ8M9ShF\
\BHsTN7KGpyIqG6Yc3GzOJ/4ir7V3I3wYguK7iBUiuTM+OKwxtM75carZJPX/2lkn2Hh\
\TC+JVb2/yaHS2oDrlCwQosNhA/cB/cm+YmYgHc3KQwrZ/3Axr6weUSrWJ8qV+vJ5QKK\
\a1+CLrkVGJX6vh+ps1NB5EV9yyMhBXAbbJZ3K+dGed3G7Vj/qF2kbnIUlSIeP1f6LsH\
\XASuVLTU6qG0rTaCMYKwaAc5ROwAgyZmPXuOUwyCtNFb0+S73uX5/N2drrUPdXiURW+\
\luFKCtaNimU8myoz6YkoY234kz8pedST8eqBZAioe8HeYEKtZSAyYov4YfLgkqHqJG6\
\ycD2uA33kwnMim+jg/hIrWAIYYP9R90KECTvFR877RtfFgfn3+tZjWlmsxHZ5pNsTIA\
\dNR+VmNpoUZkQ0dgHuFLztyAnCaumL38LHYHFHj2boa0zYsMGw8WtpEQ3+BNgoanNax\
\dJ5THRRmhvMS3EDwanERimsZ6ZjdK8uchuVhytNiiKvBwEFWYIyoK9uUMBoEfDje4DX\
\wIAefXYCqPK8eXhL+9qDLxADlDQbCu+Ey3whX/r4r2Q6l+34HpRrn3g5ok+GtO/3ni9\
\dYiIYpcXYfhMGDoXJLZ3IMkK7L6e5u4/Wye7lot2B5ekSGRrkLkjv+bTIkppxbTU4Pi\
\n40qbD91sRzw2/GzZmJsfcFaKbj5dhoNWyh5cZr1PqsxMI5EdXSxJ69VWf8e+h4iPoB\
\YS1JnUjhicVWslpA1rdAvTkAsVY8rC22e09Hxzbkb/E7bt3iLDpekbbQAghZ31AwDv5\
\KEG72bBbXIYHzPvhJzrlS2LR0XKTJVd8tAxOSdxQDOt8tE2eKpmWZ38MJfRIxt2Rzol\
\p+bpKrR++pMLRrpViekVpZl/tlEojImNO5rLqxZhLxvZOyDfcT37jc1oqire527/Y9L\
\3k894eHNYcXxjb0LGGPDeLuTSEX+afHZLNbd93Qa5VTmLwsPxEW/Erua6nXUrAR/87P\
\0gIyce3h3sl5jzCXsQm/iODgyn7PTEo5ksQCFRPiyXq5xgiXGKGGkqTGg28Ohdby+lN\
\DPnNHU2J0F6GLTqwK3qGbBLzDGIMR2sePGpxZ/pecoX7yn5bTOf4iY1OCyLo5nEgSeb\
\JdBJh0ZU+QodLRN0cnenLmP1oNK2yCuT9uIAlWH9C1CLhBiEOfoIs9/r1W1XHPiPsX7\
\c21w+B1IPfzUX1cVdndNo4XdHl9CH1tYJDLr8LfeuYnz+bnaFlqEUryTc8zUl4A+qB\
\SIDDDjefCbmDsTrdqzGT2J89MKViOogy3qJzyt3jo04xq+Q3OGjbOFJikyJEqUm8BmX\
\d3ctGfzsEr+5w7fDRco40/tDQUSH0qOWOsPkhuelLqKDziJXwhPQI42miVN2A4+OAS4\
\f2uTgpDNn1gIfH2+dOCkBjlhZeA1Tgrp8FHQxcaO5kut6cTLrL7CSBqINa7Khe1zyXa\
\PZG/tXUk+iv0BYT92b7CRNmg1qhE0G8V3q3QrB6EePYa1WxRQ7ij4rRcQWcj66A1hZ5\
\KjDUVJh+02cZTFrv97wM/im3vb3dbiSxAiQExSa2KATfLI2oS+y7RlRNJ+9nF/vTaFc\
\0HOdKfmuJAUkAcyk/h0Quvdaf9jxEcstj95mva+HkIqPuFifidlvGiafKr4fHZryp1h\
\g7QUtDRU2a4BRfzcLz6PKOBFV3xVI7qoQbKEqQyldv8mZRd0LBRKprxHW7PdUqutH2V\
\GEmZ4UuCYXT11UweBx2W9lHrQX+xaKAjTu6oLYIOvmFVCUr4mCrYRcLZnzwORcsqIl4\
\G88x8r5aeilL4lsQZO3kNotR4n0qzFVRU2+EXO7QJFm+NKxB7aRZ5oH+dSy+Ye6aMeG\
\Epv491LU0LVnZNMBP2eUhoEoOgimmZGtUobjRdLuYyNiJfJzVkjwF3gYQtY59zb+46N\
\SzvWUqpFUG80Vswns8GNAQ5hfLoH8OGGohT+UvoqvpTEXhiAAFstT/EQrHLZrYpXHJI\
```

```
\YaICW+6uo9ixL0oWkfI0HlYaXyNkaFKHQ5ZbPaP45dbWq/dqXdrRe2YU8AqdjCxyyzO\
\lyZR6zH9wHj0k1AIOHvnKZ/B2v4bS8YAtNZ1zgKbOvM4qqSIFETfr8N4yIteumHEznP\
\prD7Gr6W2VCS/0FXnQt5y0QC8z4ffrnggwPjcZfsCRSknktQB1q6Cx8KUOipf+RhOvs\
\HnNN3qJZmxz6YCvo2M7fxJtyRvm34UEVaj8QKXrmzX70Y9rDl6wEhhvSThaeq4dcfAC\
\vczGXWgCLB10gl+Iz6hVDTgCx7bC2BQ2oHtzSDc+v/UuJewvVaIL9tn4CtMZU86f3Zc\
\fTN2zke5alNpoJP9A+mkbcfy0aD6yFcn3nw2ueFDsssRg1ZcS5CujNeylAwxRYaNSmU\
\zDzMygHu0CTxfGEG2c63J32HkG4Ds58KSk7HSD58jgScBv+QjBUAGSJozFG9y7yIF5R\
\kD74aSJMYmuzow2UnGayR9yM5ONbW1brD4wNyJHqDIroCUvrL8zu24ErFWDKy6VaZ0m\
\ggPvX38IoxIPnE+PmOtRGr+ua9r9zO47TtEoADjIEtwQuNem0S1fqeVx2Fd1TmKc5+v\
\cxMsmuEKQiewTbviLdIWHTz4snU/dt77cxQEFWkS3pu31kCLyLbpiCKMrn1nELafNBg\
\RbzwEqGTT0i24Kz/kvC5RYr2USuHKksZxPfgx7Y0OpY3IbemFO11EmnG9odSwnVcww+\
\9/IIevZHUw1qqTxZu1re/AMfqhKgaD8XiwuKxPZQQo7Z6jj3yOugAVWYOw/88bAXO2k\
\deVc0mG53sKH9ChLg35LdPrpLgHeFjIHJ27L9ucqUw70Pu58vRJUnYDey997y57k1vh\
\9RwPkNvIs269v6s/xfg9VM9N4aY4X25EWCxchMWlH9LMamYF6JTP0v7v00cdHycmX5D\
\EnwqspYYNomVpJlOOxgMAO9oy1E4dhg3IJo+fJgL9rgOxJ4INTJOg/9tUz21LPI3c1c\
\D5pPs/y0zy0cF9f6ahaYxMDk/nfout2FGmoesMCaTN11JngYYC5H95cDeMwErm6ppSU\
\woCqut45noJq0VS4V3PKfASIfuUwP3vgFKo+82Wy3dqEr+sBAsve44CKQ8Tq1GLYjet\
\L3xugCkl0uaGh6TFqj2X/vJlXOW0Ouyvzt62fxeQ4esOrs4LdRxkJbKT2I2p6rQAlBi\
\GaZLvOuccQh7NSt7BEJBy8QUrPV10vPmCNGQrKS6alC/JNFLaxmsP4CPQqwRQ3fg2ia\
\qQRol0htD+UFjWUBXrQdrs48b9TdLHmbPHPbG6+ZeuCi87kJ/zJyjHA0SYUP6awkfga\
\ckiLUppo0oNIc9/qsVr2lFIWIO9+UWnIFR9nNFPzgbqw/cMOC/uWAOOsGS8ADQ/rePO\
\fTXx0mfkvI2YeTdiIayy+uwUxoLdz90DGhUysP+JGU9kZTqYNJYsjC4OgLXS+qKCYai\
\oW/leFs1fdP6SH+E24pOOJARU/f/ZajcMMXAwQdIVeOo7jvDhMydne90/18fcwpNVN0\
\tswhRsnW4uMCSAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAIEBMZHyE="
}
```


3.5. Using CRYDI with COSE

The approach taken here matches the work done to support secp256k1 in JOSE and COSE in [RFC8812].

The following tables map terms between JOSE and COSE for signatures.

Name	Value	Description	Recommended
CRYDI5	TBD	TBD	No
CRYDI3	TBD	TBD	No
CRYDI2	TBD	TBD	No

Table 5

The following tables map terms between JOSE and COSE for key types.

Name	Value	Description	Recommended
PQK	TBD	TBD	No

Table 6

4. Falcon

TODO

5. SPHINCS+

5.1. Overview

This section of the document describes the hash-based signature scheme SPHINCS+. The scheme is based on the concept of authenticating a large number or few-time signatures keypair using a combination of Markle-tree signatures, a so-called hypertree. For each message to be signed a (pseudo-)random FTS keypair is selected with which the message can be signed. Combining this signature along with an authentication path through the hyper-tree consisting of hash-based many-time signatures then gives the SPHINC+ signature. The parameter set is strategically chosen such that the probability of signing too many messages with a specific FTS keypair to impact security is small enough to prevent forgery attacks. A trade-off in parameter set can be made on security guarantees, performance and signature size.

SPHINCS+ is a post-quantum approach to digital signatures that is promises Post-Quantum Existential Unforgeability under Chosen Message Attack (PQ-EU-CMA), while ensuring that the security levels reached meet security needs for resistance to both classical and quantum attacks. The algoritm itself is based on the hardness assumptions of its underlying hash functions, which can be chosen from the set Haraka, SHA-256 or SHAKE256. For all security levels the only operations required are calls to these hash functions on various combinations of parameters and internal states.

Contrary to CRYSTALS-Dilithium and Falcon, SPHINCS+ is not based on any algebraic structure. This reduces the possible attack surface of the algorithm.

SPHINCS+ brings several advantages over other approaches to signature suites:

* Post Quantum in nature - use of cryptographically secure hash functions and other approaches that should remain hard problems even when under an attack utilizing quantum approaches
* Minimal security assumptions - compared to other schemes does not base its security on a new paradigm. The security is solely based on the security of the assumptions of the underlying hash function.
* Performance and Optimization - based on combining a great many hash function calls of SHA-256, SHAKE256 or Haraka means existing (secure) SW and HW implementations of those hash functions can be re-used for increased performance

* Private and Public Key Size - compared to other post quantum approaches a very small key size is the form of hash inputs-outputs. This then has the drawback that either a large signature or low signing speed has to be accepted
* Cryptanalysis assuarance - attacks (both pre-quantum and quantum) are easy to relate to existing attacks on hash functions. This allows for precise quantification of the security levels
* Overlap with stateful hash-based algorithms - means there are possibilities to combine implementions with those of XMSS and LMS (TODO refs)
* Inherent resistance against side-channel attacks - since its core primitive is a hash function, it thereby is hard to attack with side-channels.

The primary known disadvantage to SPHINCS+ is the size signatures, or the speed of signing, depending on the chosen parameter set. Especially in IoT applications this might pose a problem. Additionally hash-based schemes are also vulnerable to differential and fault attacks.

5.2. Parameters

TODO

5.2.1. Parameter sets

TODO

5.3. Core Operations

TODO

5.3.1. Generate

TODO

5.3.2. Sign

TODO

5.3.3. Verify

TODO

5.4. Using SPHINCS+ with JOSE

Basing off of this

5.4.1. SPHINCS+ Key Representations

TODO

5.4.2. SPHINCS+ Algorithms

TODO

5.4.2.1. Public Key

TODO

5.4.2.2. Private Key

TODO

5.4.3. SPHINCS+ Signature Representation

TODO

6. Security Considerations

The following considerations SHOULD apply to all signature schemes described in this specification, unless otherwise noted.

6.1. Validating public keys

All algorithms in that operate on public keys require first validating those keys. For the sign, verify and proof schemes, the use of KeyValidate is REQUIRED.

6.2. Side channel attacks

Implementations of the signing algorithm SHOULD protect the secret key from side-channel attacks. Multiple best practices exist to protect against side-channel attacks. Any implementation of the the CRYSTALS-Dilithium signing algorithm SHOULD utilize the following best practices at a minimum:

* Constant timing - the implementation should ensure that constant time is utilized in operations
* Sequence and memory access persistance - the implemention SHOULD execute the exact same sequence of instructions (at a machine level) with the exact same memory access independent of which polynomial is being operated on.
* Uniform sampling - uniform sampling is the default in CRYSTALS-Dilithium to prevent information leakage, however care should be given in implementations to preserve the property of uniform sampling in implementation.

* Secrecy of S1 - utmost care must be given to protection of S1 and to prevent information or power leakage. As is the case with most proposed lattice based approaches to date, fogery and other attacks may succeed, for example, with Dilithium through leakage of S1 through side channel mechanisms.

6.3. Randomness considerations

It is recommended that the all nonces are from a trusted source of randomness.

7. IANA Considerations

The following has NOT YET been added to the "JSON Web Key Types" registry:

* "kty" Parameter Value: "PQK"
* Key Type Description: Base 64 encoded string key pairs
* JOSE Implementation Requirements: Optional
* Change Controller: IESG
* Specification Document(s): Section 2 of this document (TBD)

The following has NOT YET been added to the "JSON Web Key Parameters" registry:

* Parameter Name: "pset"
* Parameter Description: The parameter set of the crypto system
* Parameter Information Class: Public
* Used with "kty" Value(s): "PQK"
* Change Controller: IESG
* Specification Document(s): Section 2 of this document (TBD)

* Parameter Name: "xs"
* Parameter Description: The shake256 of the public key
* Parameter Information Class: Public
* Used with "kty" Value(s): "PQK"
* Change Controller: IESG
* Specification Document(s): Section 2 of this document (TBD)

* Parameter Name: "ds"
* Parameter Description: The shake256 of the private key
* Parameter Information Class: Private
* Used with "kty" Value(s): "PQK"
* Change Controller: IESG
* Specification Document(s): Section 2 of this document (TBD)

* Parameter Name: "d"
* Parameter Description: The private key
* Parameter Information Class: Private

*Used with "kty" Value(s): "PQK"
*Change Controller: IESG
*Specification Document(s): Section 2 of RFC 8037

*Parameter Name: "x"
*Parameter Description: The public key
*Parameter Information Class: Public
*Used with "kty" Value(s): "PQK"
*Change Controller: IESG
*Specification Document(s): Section 2 of RFC 8037

The following has NOT YET been added to the "JSON Web Signature and Encryption Algorithms" registry:

*Algorithm Name: "CRYDI3"
*Algorithm Description: CRYDI3 signature algorithms
*Algorithm Usage Location(s): "alg"
*JOSE Implementation Requirements: Optional
*Change Controller: IESG
*Specification Document(s): Section 3.1 of this document (TBD)
*Algorithm Analysis Documents(s): (TBD)

The following has been added to the "JSON Web Key Lattice" registry:

*Lattice Name: "CRYDI5"
*Lattice Description: Dilithium 5 signature algorithm key pairs
*JOSE Implementation Requirements: Optional
*Change Controller: IESG
*Specification Document(s): Section 3.1 of this document (TBD)

*Lattice Name: "CRYDI3"
*Lattice Description: Dilithium 3 signature algorithm key pairs
*JOSE Implementation Requirements: Optional
*Change Controller: IESG
*Specification Document(s): Section 3.1 of this document (TBD)

*Lattice Name: "CRYDI2"
*Lattice Description: Dilithium 2 signature algorithm key pairs
*JOSE Implementation Requirements: Optional
*Change Controller: IESG
*Specification Document(s): Section 3.1 of this document (TBD)

8. Appendix

*JSON Web Signature (JWS) - RFC7515
*JSON Web Encryption (JWE) - RFC7516
*JSON Web Key (JWK) - RFC7517
*JSON Web Algorithms (JWA) - RFC7518
*JSON Web Token (JWT) - RFC7519

*JSON Web Key Thumbprint - RFC7638
*JWS Unencoded Payload Option - RFC7797
*CFRG Elliptic Curve ECDH and Signatures - RFC8037
*CRYSTALS-Dilithium - Dilithium

8.1. Test Vectors

//TODO

9. Normative 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", 2018, <https://doi.org/10.13154/tches.v2018.i1.238-268>.

[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>.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <https://www.rfc-editor.org/info/rfc4648>.

[RFC7515] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <https://www.rfc-editor.org/info/rfc7515>.

[RFC7517] Jones, M., "JSON Web Key (JWK)", RFC 7517, DOI 10.17487/RFC7517, May 2015, <https://www.rfc-editor.org/info/rfc7517>.

[RFC7638] Jones, M. and N. Sakimura, "JSON Web Key (JWK) Thumbprint", RFC 7638, DOI 10.17487/RFC7638, September 2015, <https://www.rfc-editor.org/info/rfc7638>.

[RFC8702] Kampanakis, P. and Q. Dang, "Use of the SHAKE One-Way Hash Functions in the Cryptographic Message Syntax (CMS)", RFC 8702, DOI 10.17487/RFC8702, January 2020, <https://www.rfc-editor.org/info/rfc8702>.

[RFC8812] Jones, M., "CBOR Object Signing and Encryption (COSE) and JSON Object Signing and Encryption (JOSE) Registrations for Web Authentication (WebAuthn) Algorithms", RFC 8812, DOI 10.17487/RFC8812, August 2020, <https://www.rfc-editor.org/info/rfc8812>.

10. Informative References

[RFC6234]
Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, DOI 10.17487/RFC6234, May 2011, <https://www.rfc-editor.org/info/rfc6234>.

Authors' Addresses

Michael Prorock
mesur.io

Email: mprorock@mesur.io

Orie Steele
Transmute

Email: orie@transmute.industries

Rafael Misoczki
Google

Email: rafaelmisoczki@google.com

Michael Osborne
IBM

Email: osb@zurich.ibm.com

Christine Cloostermans
NXP

Email: christine.cloostermans@nxp.com