

None  
Internet-Draft  
Intended status: Standards Track  
Expires: 6 September 2022

M. Prorock  
mesur.io  
O. Steele  
Transmute  
R. Misoczki  
Google  
M. Osborne  
IBM  
C. Cloostermans  
NXP  
5 March 2022

JSON Encoding for Post Quantum Signatures  
draft-prorock-cose-post-quantum-signatures-00

## 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 serializations of existing cryptographic systems.

This document registers key types for JOSE and COSE, specifically PQC, 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

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 September 2022.

Internet-Draft

post-quantum-signatures

March 2022

## Copyright Notice

Copyright (c) 2022 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

<a href="#">1.</a>	Notational Conventions . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Terminology . . . . .	<a href="#">3</a>
<a href="#">3.</a>	CRYSTALS-Dilithium . . . . .	<a href="#">3</a>
<a href="#">3.1.</a>	Overview . . . . .	<a href="#">3</a>
<a href="#">3.2.</a>	Parameters . . . . .	<a href="#">5</a>
<a href="#">3.2.1.</a>	Parameter sets . . . . .	<a href="#">5</a>
<a href="#">3.3.</a>	Core Operations . . . . .	<a href="#">5</a>
<a href="#">3.3.1.</a>	Generate . . . . .	<a href="#">5</a>
<a href="#">3.3.2.</a>	Sign . . . . .	<a href="#">5</a>
<a href="#">3.3.3.</a>	Verify . . . . .	<a href="#">5</a>
<a href="#">3.4.</a>	Using CRYDI with JOSE . . . . .	<a href="#">6</a>
<a href="#">3.4.1.</a>	CRYDI Key Representations . . . . .	<a href="#">6</a>
<a href="#">3.4.2.</a>	CRYDI Algorithms . . . . .	<a href="#">7</a>
<a href="#">3.4.3.</a>	CRYDI Signature Representation . . . . .	<a href="#">16</a>
<a href="#">3.5.</a>	Using CRYDI with COSE . . . . .	<a href="#">20</a>
<a href="#">4.</a>	Falcon . . . . .	<a href="#">20</a>
<a href="#">5.</a>	SPHINCS+ . . . . .	<a href="#">20</a>
<a href="#">5.1.</a>	Overview . . . . .	<a href="#">20</a>
<a href="#">5.2.</a>	Parameters . . . . .	<a href="#">22</a>
<a href="#">5.2.1.</a>	Parameter sets . . . . .	<a href="#">22</a>
<a href="#">5.3.</a>	Core Operations . . . . .	<a href="#">22</a>
<a href="#">5.3.1.</a>	Generate . . . . .	<a href="#">22</a>
<a href="#">5.3.2.</a>	Sign . . . . .	<a href="#">22</a>
<a href="#">5.3.3.</a>	Verify . . . . .	<a href="#">22</a>
<a href="#">5.4.</a>	Using SPHINCS+ with JOSE . . . . .	<a href="#">22</a>
<a href="#">5.4.1.</a>	SPHINCS+ Key Representations . . . . .	<a href="#">22</a>
<a href="#">5.4.2.</a>	SPHINCS+ Algorithms . . . . .	<a href="#">22</a>

<a href="#">5.4.3.</a>	SPHINCS+ Signature Representation . . . . .	<a href="#">23</a>
<a href="#">6.</a>	Security Considerations . . . . .	<a href="#">23</a>
<a href="#">6.1.</a>	Validating public keys . . . . .	<a href="#">23</a>
<a href="#">6.2.</a>	Side channel attacks . . . . .	<a href="#">23</a>
<a href="#">6.3.</a>	Randomness considerations . . . . .	<a href="#">23</a>

<a href="#">7.</a>	IANA Considerations . . . . .	<a href="#">24</a>
<a href="#">8.</a>	Appendix . . . . .	<a href="#">25</a>
<a href="#">8.1.</a>	Test Vectors . . . . .	<a href="#">26</a>
<a href="#">9.</a>	Normative References . . . . .	<a href="#">26</a>
<a href="#">10.</a>	Informative References . . . . .	<a href="#">26</a>
	Authors' Addresses . . . . .	<a href="#">27</a>

## [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] utilizing 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 approach 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 algorithm 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  $\mathbb{Z}_q[X]/(X^{256}+1)$ . This ensures that to increase or decrease the security level involves 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 theoretical 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 maintaining security - a danger in many possible approaches to cryptography is that it may be possible inadvertently introduce errors in code that lead to weakness or

- \* 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 matrices 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.

Prorock, et al. Expires 6 September 2022 [Page 4]

### 3.2. Parameters

### 3.2.1. Parameter sets

NIST Level	Matrix Size	memory in bits
1	1024	1024
2	2048	2048
3	4096	4096
4	8192	8192
5	16384	16384
6	32768	32768
7	65536	65536
8	131072	131072
9	262144	262144
10	524288	524288
11	1048576	1048576
12	2097152	2097152
13	4194304	4194304
14	8388608	8388608
15	16777216	16777216
16	33554432	33554432
17	67108864	67108864
18	134217728	134217728
19	268435456	268435456
20	536870912	536870912
21	1073741824	1073741824
22	2147483648	2147483648
23	4294967296	4294967296
24	8589934592	8589934592
25	17179869184	17179869184
26	34359738368	34359738368
27	68719476736	68719476736
28	137438953472	137438953472
29	274877906944	274877906944
30	549755813888	549755813888
31	1099511627776	1099511627776
32	2199023255552	2199023255552
33	4398046511104	4398046511104
34	8796093022208	8796093022208
35	17592186044416	17592186044416
36	35184372088832	35184372088832
37	70368744177664	70368744177664
38	140737488355328	140737488355328
39	281474976710656	281474976710656
40	562949953421312	562949953421312
41	1125899906842624	1125899906842624
42	2251799813685248	2251799813685248
43	4503599627370496	4503599627370496
44	9007199254740992	9007199254740992
45	18014398509481984	18014398509481984
46	36028797018963968	36028797018963968
47	72057594037927936	72057594037927936
48	144115188075855872	144115188075855872
49	288230376151711744	288230376151711744
50	576460752303423488	576460752303423488
51	1152921504606846976	1152921504606846976
52	2305843009213693952	2305843009213693952
53	4611686018427387904	4611686018427387904
54	9223372036854775808	9223372036854775808
55	18446744073709551616	18446744073709551616
56	36893488147419103232	36893488147419103232
57	73786976294838206464	73786976294838206464
58	147573952589676412928	147573952589676412928
59	295147905179352825856	295147905179352825856
60	590295810358705651712	590295810358705651712
61	1180591620717411303424	1180591620717411303424
62	2361183241434822606848	2361183241434822606848
63	4722366482869645213696	4722366482869645213696
64	9444732965739290427392	9444732965739290427392
65	18889465931478580854784	18889465931478580854784
66	37778931862957161709568	37778931862957161709568
67	75557863725914323419136	75557863725914323419136
68	151115727451828646838272	151115727451828646838272
69	302231454903657293676544	302231454903657293676544
70	604462909807314587353088	604462909807314587353088
71	1208925819614629174706176	1208925819614629174706176
72	2417851639229258349412352	2417851639229258349412352
73	4835703278458516698824704	4835703278458516698824704</

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 (<https://datatracker.ietf.org/doc/html/rfc8812#section-3>)

#### 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 specified to indicate the not only parameter set in use for the algorithm, but SHOULD also reflect the targeted NIST level for the algorithm in combination with the specified parameter 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	d	2	4000	5337
Key				

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 likelihood 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": "z7u7GwhsjjnFHH3Nkrs2xvvw020Rcw5ymdlTnhRenjDdr00+nfXRVUZVy9q1\
5zDn77zTgrIskM3WX8bqslc+B1fq12iA/wxD2jc1d6j+YjKctkGH260R7vc0YC2ZiMzW\
zG17yebt7JkmjRbN1N+u/2fAKFLuziMcLNP6WLoWbMqxoC2X00VNAWX3QjXrCcGU23Nr\
imtdmWz5NrP43E592Sctt5M+SVlfgQeYv8pHmtkQknE8/jr7TrgNpuiV7nXmhWHTMJ4I\
zoGXgq43odFFthboEdKNT/enyu+VvUGoIJ6cN8C/1B6o1WLYHEaL0BEIFFbAiAhZ/vnf\
cUYMaVPqsDJuETsjetcE32kGCD7Jkume2t068DLiHb/2Z2JX8mkcbxFI6KrmXiRxXQj9\
9LVn1fEzdf3Vfpcs/C3omsFGqmTpLDK+AvW/SWVkdI2NKq7hL/Ayx1W2u2cqVERqZUTS\
Z+ic6V8kZfxr3gRMnH0KuF5BtjleZ/yVvqqPjwP0ZegCKEL2Gd8duhcUde7CR55pil1o\
UXy5AwgCcZTdEcJn10P0bGoots9T19gw1x4vnZCQUKVPZuZ1gIkGqDUYXS0lcNTjCMs\
miFEmn0ZvB88jxULpb1vl9HoQ3ocM2oZu4AZRt9G/L07MwcuioFCWtAIau+2gqNAn/Z\
AS10l0j2N0LLtAa0xoF+Ctzscrt0ZMyGHmoQ9daHkpUvEq0c08hDtLp1nq3lQIIIfROQ\
jcNs9vNKBu87COBjukZD+L8vV4zy8FN059MCSb9UCLwz2xvfdI1js9/J7hTGaVec8VPx\
md42yPFRgW5Na1oefm8vW49EDmevc8AjAtwDirRBDfv9pX3+5S+M6jhteSLYvpKJXQT1\
zs1379KvIHwkn9VHpA+PiUUw9TgF6x8xWEGSNl0o1Vn1xtM3givehjYxJ5p5/kBEFZI\
DCyFzstAirJ2GadNhae+P1JFZzJWnX5jaLwzldquZwF3yTzNho4sgBA+fKqiXcgn2nw1\
vz0Dkbxr6cMaUool0eFScU1nAz1Z39W64Lt2nEuYsORx/ht2RzJxxFc21X3nLeEDFCe\
NkNDxQFBSfpZjKKgJtXEx23mp+CbBVMrbagsLnzsAGLYbnroVmATU5Iqr6LgYBpuFs+N\
Rkq7ZXh6CZPukMGQbc0GuNw06NBuuMNHir5ayGk1ZBiW82C7Nu0hs2pLcgNqWMtt1+LW\
8R96KyoSc784ZYAZ40QqvoySwwxQPBRTRJ+wB0sVpGBLTxdY9Gw3pXeXN5nao340d2ZA\
7YEMlqcTHCAv3F8B9ewl70fQlmg6bvdMuoVdVE+p0er7IAmWMMRgviIzYv9sKEEQrCmua\
2qL5xPSbD05KRf8ZAZ2B8LSCDR1nzXrQXZbXBKJivsCVQDuzxrwGE0gqRmpbk4f5GYCG\
4i/08Knoru+jjf6wVQDYKfyz1QUGRlXHkGUGlXfv03r7UbJugycjV05kbGxhoZkq0q8z\
ZEpkfvrNoxeotw/z4QpjI8JlY97GDb0mGVHbmdHugjMtVTGhVJFBbPIinmR+emt70+\
4qOr7yWRxCvt2lziWtpPBwaf/1XDnN5Gesex1gR1YrcTRNmB808b01sxLQmxcTt4eQ0/\
LUkas7qTJ3AQThOfDdtIpkqsthsBFy+WjSQuoXCymRcPi6MlpxJndDF32lCnL1ranV6e\
F2ST0SYT+NwNDesMzTRmNbHUW5KAhu0k9WABTvcM5ba0Uq6i0a1NsFrcLag+KhxN6HPn\
oobwJ/EsDi5S7TAl8WrjqIhZ8x6h9eRRXerpa0w/FYk+2MpWByp/98VE12/Ew0qAIiPp\
elAvUeM0lRkpG64bJsmYtHuNWgcv5Qiy7/eGw9ZpvB3J3G3jxvbynExqdFyDc067EKi\
5WxDfPuZUjkfKpekNvzQuIrs49BzcRyMt5ndEVE21TPPfZ/R8B7Rxn2LiK+hQc+cc9\
pEEaWgwA0iMILcp/1CyY6Imd06RHsxwflMH7gej+hN41kaoEghI0l9kMGT LZbq5Pc8Pz\
6F2LKTBMJWg9o/0blvilMH9EPblcLeF/bR1AZTUD6ZFdi2TxN6Epn3QVqeG/qPm1EBTF\
Gw1V92m6/08Dd6zI1HPqwKbkHx4F567owofKHaM2imin0yVUPwxoRJRulRHMCB3tn8C4\
ZpFl+sGV3Gip3tKLS7PKQkTqI6DMwxEbdrvtDy1sHZagpcLLDisA/yFT4RR2m3VNJR9P\
6Nx3teqN1eg6RXmD/MlKcdWrlcjZ/6yeIQYwbr9CjItY/tLQX2gtAR1SX0h99UUBVv+Z\
E03VOZ+Ecsc78lSB9G/6n6CFzlbk/HgAF+cu0yMbGnEM8W3mTUSpS4JBACwk5w0XWNNQ\
DWVEdgzuLGHpQ+hYExDjVZrLELhkh8YgZA+7RXXUZHM/joN0GHUhpUG/bFo3ktnaILCu\
xs0XMUBDC3VcitFFHsGK1svtcERDFxk1HA8pGa59jT0do6n3wEbnBDU1soKNFtpmcVKE\
Ul3XpvuoW3BgCwJzBUCWvPs47DJRgX011bSaEYYlhTVaaShcvzgz46Akq0+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": "z7u7GwhsjjnfHH3Nkrs2xvbw020Rcw5ymdlTnhRenjDdr00+nfXRVUZVy9q1\
5zDn77zTgrIskM3WX8bqslc+B1fq12iA/wxD2jc1d6j+YjKctkGH260R7vc0YC2ZiMzW\
zG17yebt7JkmjRbN1N+u/2fAKFLuziMcLNP6WLoWbMqxoC2X00VNAWX3QjXrCcGU23Nr\
imtdmWz5NrP43E592Sctt5M+SVlfgQeYv8pHmtkQknE8/jr7TrgNpuiV7nXmhWHTMJ4I\
zoGXgq43odFFthboEdKNT/enyu+VvUGoIJ6cN8C/1B6o1WLYHEaL0BEIFFbAiAhZ/vnf\
cUYMaVPqsDJuETsjetcE32kGCD7Jkume2t068DLIhB/2Z2JX8mkcbxFI6KrmXiRxXQj9\
9LVn1fEzdf3Vfpcs/C3omsFGqmTpLDK+AvW/SWVkdI2NKq7hL/Ayx1W2u2cqVERQZUTS\
Z+ic6V8kZfxr3gRMnH0KuF5BtjleZ/yVvqqPjwP0ZegCKEL2Gd8duhcUde7CR55pil1o\
UXy5AwgCcZTdEcJn10P0bGoots9T19gw1x4vnZCQUKVDPZuZ1gIkGqDUYXS0lcNTjCMs\
miFEmn0ZvB88jxULpb1vl9HoQ3ocM2oZu4AZRt9G/L07MwcuioFCWtAIau+2gqNAN/Z\
AS10l0j2N0LLtAa0xoF+Ctzscrt0ZMyGHmoQ9daHkpUvEq0c08hDtLp1nq3lQIIIfROQ\
jcNs9vNKBu87COBjukZD+L8vV4zy8FN059MCSb9UCLwz2xvfdI1js9/J7hTGaVec8VPx\
md42yPFRGw5Na1oefm8vW49EDmvec8AjAtwDirRBDfV9pX3+5S+M6jhteSLYvpKJXQT1\
zs1379KvIHwkn9VHpA+PiUUw9TgF6x8xWEGSNl0o1Vn1xtM3givehjYxJ5p5/kBEFZI\
DCyFzstAirJ2GadNhae+P1JFZZJWnX5jaLwzldquZwF3yTzNho4sgBA+fKqiXcgn2nw1\
vz0DKbXR6cMaUool0eFScU1nAz1Z39W64LtT2nEuYsORx/ht2RzJxxFc21X3nLeEDFCe\
NkNDxQFBSfpZjKKgJtXEx23mp+CbBVMrbagsLnzsAGLYbnroVmATU5Iqr6LgYBpuFs+N\
Rkq7ZXh6CZPukMGQbc0GuNw06NBuuMnhir5ayGk1ZBiW82C7Nu0hs2pLcgNqWMtt1+LW\
8R96KyoSc784ZYAZ40QqvoySwwxQPBRTRJ+wB0sVpGBLTxdY9Gw3pXeXN5nao340d2ZA\
7YEMlqcTHCAv3F8B9ewl70fQlmg6bvdMuoVdVE+p0er7IAmWMRgviIzYv9sKEEQrCmua\
2qL5xPSbD05KRf8ZAZ2B8lSCDR1nzXrQXZbXBKJivsCVQDuzxrwGE0gqRmpbk4f5GYCG\
4i/08Knoru+jjf6wVQDYKfyz1QUGRlXHkGUGLXfv03r7UbJugycjV05kbGxhoZkqQq8z\
ZEpkfvrNoxeotw/z4QpjI8JlY97GDb0mGVHbmdHugjMtVTGhVJFBbPIinmR+emt70+\
4qOr7yWRxCvt2lziWtpPBwaf/1XDnN5Gesex1gR1YrcTRNmB808b01sxLQmxcTt4eQ0/\
LUkas7qTJ3AQThOfDdtIpkqsthsBFy+WjSQuoXCymRcPi6MlpxJndDF32lCnL1ranV6e\
F2ST0SYT+NwNDesMzTRmNbHUW5KAhu0k9WABTvcM5ba0Uq6i0a1NsFrcLag+KhxN6HPn\
oobwJ/EsDi5S7TAl8WrjqIhZ8x6h9eRRXerpa0w/FYk+2MpWByp/98VE12/Ew0qAIiPp\
elAvUeM0lRkpG64bJsmYtHuNWgcv5Qiy7/eGw9ZpvB3J3G3jxvbyExqdFyDc067EKi\
5WxDFPuZUjkfKpekNvzQuIrqs49BzcRyMt5ndEVE21TPPfZ/R8B7Rxbn2LiK+hQc+cc9\
pEEaWgwa0iMILcp/1CyY6Imd06RHsxwflMH7gej+hN41kaoEghIOl9kMGTLZbq5Pc8Pz\
```

6F2LKTBMJWg9o/0blvilMH9EPblcLeF/bR1AZTUD6ZFdi2TxN6Epn3QVqeG/qPm1EBTF\  
Gw1V92m6/08Dd6zI1HPqwKbkHx4F567owofKHaM2imin0yVUpwxoRJruLRHMCB3tn8C4\  
ZpFl+sGV3Gip3tKLS7PKQkTqI6DMwxEbdrvtdY1sHZagpcLLDisA/yFT4RR2m3VNJR9P\  
6Nx3teqN1eg6RXmD/MlKCdWrlcjZ/6yeIQYwbr9CjItY/tLQX2gtAR1SX0h99UUBVv+Z\  
E03V0Z+Ecsc78lSB9G/6n6CFzlbk/HgAF+cu0yMbGnEM8W3mTUspS4JBACwk5w0XWNNQ\  
DWVEdgzuLGhPq+hYExDjVZrLELhkH8YgZA+7RXXUZHM/joNOGHUhpUG/bFo3ktnaILCu\  
xs0XMUBDC3VcitFFHsGK1svtcERDFxk1HA8pGa59jT0do6n3wEbnBDU1soKNFtpmcVKE\  
Ul3XpvuoW3BgCwJzBUCWvPs47DJRgGx011bSaEYYlhTVaaShcvzgz46Akq0+Q7TjckDP\  
/8uzsSQk0AbuhxWFQpSiBP80Z/U="

}

#### [3.4.2.2](#). Private Key

Per section 5.1 of [\[CRYSTALS-Dilithium\]](#):

| The secret key contains  $p, K, tr, s_1, s_2$  and  $t_0$  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": "z7u7GwhsjjnfHH3Nkrs2xvvw020Rcw5ymdlTnhRenjDdr00+nfXRVUZVy9q1\  
5zDn77zTgrIskM3WX8bqslc+B1fq12iA/wxD2jcd6j+YjKctkGH260R7vc0YC2ZiMzW\  
zGL7yebt7JkmjRbN1N+u/2fAKFLuziMcLNP6WLoWbMqxoC2X00VNAWX3QjXrCcGU23Nr\  
imtdmWz5NrP43E592Sctt5M+SVlfgQeYv8pHmtkQknE8/jr7TrgNpuiV7nXmhWHTMJ4I\  
zoGXgq43odFFthboEdKNT/enyu+VvUGoIJ6cN8C/1B6o1WLYHEaL0BEIFFbAiAhZ/vnf\  
cUYMaVPqsDJuETsjetcE32kGCD7Jkume2t068DLIhB/2Z2JX8mkcbxFI6KrmXiRxxQj9\  
9LVn1fEzdf3Vfpcs/C3omsFGqmTpLDK+AvW/SWVkdI2NKq7hL/AyxLW2u2cqVERQZUTS\  
Z+ic6V8kZfxr3gRMnH0KuF5BtjleZ/yVvqqPjwP0ZegCKEL2Gd8duhcUde7CR55pil1o\  

```

UXy5AwgCcZTdEcJn10P0bGoots9T19gw1x4vnZCQUKVDPZuZ1gIkGqDUYXS0lcNTjCMs\  
miFEmn0ZvB88jxULpb1vl9HoQ3ocM2oZu4AZRt9G/L07MwcuioFCWtAIau+2gqNAN/Z\  
AS10l0j2N0LLtAa0xoF+Ctzscrt0ZMYGHmoQ9daHkpUvEq0c08hDtLpInq3lQIIIfROQ\  
jcNs9vNKBu87COBjukZD+L8vV4zy8FN059MCSb9UCLwz2xvfdI1js9/J7hTGaVec8VPx\  
md42yPFRGw5Na1oefm8vW49EDmevc8AjAtwDirRBDfV9pX3+5S+M6jhteSLYvpKJXQT1\  
zs1379KvIHwkn9VHPA+PiUUw9TgF6x8xWEGSNl0o1Vn1xtM3givehjYxJ5p5/kBEFZI\  
DCyFzstAirJ2GadNhae+P1JFZzJWnX5jaLwzldquZwF3yTzNho4sgBA+fKqiXcgn2nw1\  
vz0Dkbxr6cMaUool0eFScU1nAz1Z39W64LtT2nEuYsORx/ht2RzJxxFc21X3nLeEDFCe\  
NkNDxQFBSfpZjKKgJtXEx23mp+CbBVMrbagsLnzsAGLYbnroVmATU5Iqr6LgYBpuFs+N\  
Rkq7ZXh6CZPukMGQbc0GuNw06NBuuMNHir5ayGk1ZBiW82C7Nu0hs2pLcgNqWMtt1+LW\  
8R96KyoSc784ZYAZ40QqvoySwwxQPBRTJ+wB0sVpGBLTxdY9Gw3pXexN5nao340d2ZA\  
7YEMlqcTHCAv3F8B9ewl70fQlmg6bvdMuoVdVE+p0er7IAmWMRgviIzYv9sKEEQrCmua\  
2qL5xPSbD05KRf8ZAZ2B8LSCDR1nzXrQXZbXBKJivsCVQDuzxrwGE0gqRmpbk4f5GYCG\  
4i/08Knoru+jj f6wVQDYKfyz1QUGRlXHkGUGLxfv03r7UbJugycjV05kbGxhoZkq0q8z\  
ZEpkfvrNoxeotw/z4QpjI8JlY97GDb0mGVHbmdHugjMtVTGhVJFBbPIinmR+emt70+\  
4q0r7ywRxCvt2lziWtpPBwaf/1XDnN5Gesex1gr1YrcTRNmB808b01sxLQmxcTt4eQ0/\

LUkas7qTJ3AQThOfDdtIpkqsthsBFy+WjSQuoXCymRcPi6MlpxJndDF32lCnL1ranV6e\  
F2ST0SYT+NwNDesMzTRmNbHUW5KAhu0k9WABTvcM5ba0Uq6i0a1NsFrcLag+KhxN6HPn\  
oobwJ/EsDi5S7TAl8WrjqIhZ8x6h9eRRXerpa0w/FYk+2MpWByP/98VE12/Ew0qAiPp\  
eLAvUeM0lRkpG64bJsmYtHuNWgcv5Qiy7/eGw9ZpvB3J3G3jxvbynExqdFyDc067EKi\  
5WxDFPuZUjkfKpekNvzQuIrs49BzcRyMt5ndEVE21TPPfZ/R8B7Rxn2LiK+hQc+cc9\  
pEEaWgwA0iMILcp/1CyY6Imd06RHsxwflMH7gej+hN41kaoEghI0l9kMGTlZbq5Pc8Pz\  
6F2LKTBMJWg9o/0blvilMH9EPblcLeF/bR1AZTUD6ZFdi2TxN6Epn3QVqeG/qPm1EBTF\  
Gw1V92m6/08Dd6zi1HPqwKbkHx4F567owoKHaM2imin0yVUpwxoRjruLRHMCB3tn8C4\  
ZpFl+sGV3Gip3tKlS7PKQkTqI6DMwxEbdvtdY1sHZagpcLLDisA/yFT4RR2m3VNJR9P\  
6Nx3teqN1eg6RXmD/MlKCdWrLcjZ/6yeIQYwbr9CjItY/tLQX2gtAR1SX0h99UUBVv+Z\  
E03VOZ+Ecsc78lSB9G/6n6CFzlbk/HgAF+cu0yMbGnEM8W3mTUSpS4JBACwk5w0XWNNQ\  
DWVEdgzuLGHpq+hYExDjVZrLELhkh8YgZA+7RXXUZHM/joNOGHUhpUG/bFo3ktnaILCu\  
xs0XMUBDC3VcitFFHsGK1svtcERDFxk1HA8pGa59jT0do6n3wEbnBDU1soKNFtpmcVKE\  
Ul3XpvuoW3BgCwJzBUCWvPs47DJRgGx011bSaEYYlhTVaaShcvzgz46Akq0+Q7TjckDP\  
/8uzssQk0AbuhxWFQpSiBP80Z/U=",

"d": "z7u7GwhsjjnfHH3Nkrs2xvvw020Rcw5ymdlTnhRenjDUBgLGfklHURz5btM5\  
yrI5FQdWk+U2srVuSmfDV7EYG897mUFY35Z0WQ0mZ9XvIOKCh+GFF0k56b5F0Fq6xnV8\  
UDQnFyY2JREUOHdiUjcUNxA1YxR3QiQ0BkE1AUBmFE0AUHZGBzQAU2dxVIgTQRV3U3g4\  
GGiISEYQhHRSWDIBQ2Z3UIIWdSV1EWhwBTYiWGI3VmJVI1UIU2REdUHHBoJ2gRhFUTHy\  
BSQnhBIGI1AoMVB2MCNhUXQiNUGCKHgzUmQxU3dEgBhmQyIQgmFjdxY1dCJgGBSEB4Ij\  
CEJ0MBGIQWRRN3QjRmRSQWQIJgNjcdnMlJhJIU1MlJRd1NmF4dwhHIIdeYYcAhEcLBQ\  
JjESAIbWbQYzYlAIIOcBcoZFcGVkA2SDMCVTBjgzCAAzNnQGYHI1VwJzYxQRckBIZBV4\  
VxZmZiVlYXgHFRNjdEFIYVOFIVdhcnIINEhhIURjg0cxJ0SCIWYUUVcHJzdDUTASciAW\  
UiJAgliOqkiWnJmLcwACZxcHZVJA4EnNnWAZVVoBjNnNEcicTdyEEUHBVFjETETNjd0\  
YUFmFDVXNUcHFVJoE2AlVwFhMzc0dQckMYJUaBJ0JkUBdyd1AnZiJ3hYYkWAgyR1VziD\  
"

BERVh1NQAFIGWIhUZXCEQxIThyR1FIGGNTKAcwdRNBGDYEd2RVEwUDMzWAAhNQEEMYAS\  
hUSCgTJ3VIdXULFBFxFkhVCCZEABJjQCAxFGNkhHQzM1YSSHNLEBEmJkgWZCVwZHRSD\  
GDZzRCQINhE3IghDhSFoBYCHNFVHMxZAZTSGAVMUQkhBKIFRQEVBB0cHNGEiJVVScQQh\  
hzQiBzKBYRY4Q0R2MVUldwVCIkQYEEgFYEREY2NERyFVdHJzdAZHBmhmdSCFIgh1IwGB\  
AxNzFjEoUWFCF4AhZRJSEROEETHxAGYBJTgUcUdFJBOFRmcVYnZUUFcAY1AnZEZBVThs\  
ECBQYBNGeAdzQ3KGhIE3ZiFxQoVCgQBjEFdFcIV0IAaFcgi0iAgAUVQAHEInQWECY1I4\  
E2U0MkAXQSZkdSRRc2I4BjEiSGd4hgQEEDcTRmhFd3MBMmY2gERxNwiISAVkFVETGCcI\  
gYhzRXByGBgzKGVXJUhoGEY1hgFmZWgmEWYwVoISd4VlhId4VUMHgSZXhXUSaCgmJ3Eg\  
MQIzIDIThwRSARQhBFB2QQBjEgIoEHN1BhMCiIIBULZWCHdBMyZFQoQ2UUJXJCFnZyFD\  
RTFUUTQoQmUjB0aHJXJHeDZLJGBCExATEUEDg3BTACHWImN0AWcFB3IUgBEARxVUREDx\  
QzhVBEBBF4UgcRhCg4gUcoRkdYCAIUQBRUJUyJfjEBYUhsBjIyV2cXBYckeIiMic1N4ED\  
gDUGVSCHHCYURXcQB10Dg4hDJFJ3Jld2gyYnQYclRjE3VwcQNXJBYN0GhYFoU2dHATAw\  
OEeBUGQjJHcDgUJTRXBxJ3IAEjEXhXeEEemghIAAGdjUVBndnI2FCAVcyV4cxIyZ2dYE1\  
ZEiHcYJYaCcXQXJCaER1coIDRWJkcsNhEVF3JGOCQkYWYkcndRA1QTh0MFgzIyVocYJY\  
cwIAFRNiE1VAUECBI3MzQiZkUhr1cid1NSAXFXN0gUNnRCV0ISNmYnB3NIZyMIQWEVvz\  
ElEohnQyKBNoY1cCdmdjVYiGhVULA0CHMTZIURBgR4aIJwJQJLIDMYhwNGNwiGcIBUDa\  
NXMBETUGICJyMkMIN1BAIAB4gEMDUwhndFJkQDEgRRMld4EzhRM0ZhGAYHMGZ4NhEEhn\  
U2I2VicBBXZUFUNoczcmbZBnUEBxUwCdG4RHUXZS0EZogABHRAISVEAZdoQhRmBgEWEY\  
Z4UldgQ2gjgQUjMScgIIFQR3YFcZhTYoB1IiVCYGBAVmVAFIdhRmZ1InhwdTRjJjNhVx\  
IzcWgjFLFCaChLIWUySIaDFwAWV3RAIxg2QWYgAYMUjP7wmwOwPp7Ukl3L1KaLY/6dN4\  
dBr1AYS8JnkVq6pPeBf07ccX95SrVfA07EX7RVEYyhVR9Q0QyEpLBUMcfcfnHCZWKM0o\  
0BF7BXiWMR9BQo4ybtPjGKQ+IZyCKUJVRhZ+uae182qYcBKFMd00zXi08kAa98eUy6SR\  
pPfkPD6D+xXgtJ0FWtYnp1Jy2aIG3HqMiThoSdVivccGkf94gpVWTMeJQsQpgq7dAJiJ\

5JOMQjk7JIHcIzxb4T8sQHZA55MFfvM7Hus/8FUX7NfIN1JRmc2zHL/7kdfCFSwG67iW\  
U4ob2kTwdKzPvOL+d3e+A0E0PihJ4vVJA0jhWm02fIFNvFhNqPh0MSiSkatPGbSVdqQ1\  
PsG6C+1YqMrTM7KFr4hTQM8a3+tA0sImMjXSSPDkVeuJFq1rw642SJJx8yZTXVe8g75D\  
ZTYghbeX5LLzaVkt9mZS7cW16Zy+C3MwnWDRGQ6hUDxYaYJp7SOGJHepcmVV214oD6nw\  
5QprgpGIxVcdXQU00fhKwerYDko0Ij+uqk7NYDv0t8zANphYcE3v+6yVFyYh3eg7DYRJ\  
rIzIcbaG91ySv2iRRC+cWaymH6xuqaHRwZu/p962/u8/c3rITJzCoVc+0bnZ5oItZfBe\  
AYFhLBx7PvPdBULXyCqmtk0tnT/jnaCUVxtGeaIeQmmeM4yPq3d5uWBf0vIyuPmfBSKd\  
Y0NETGlsaoQuqFp0kCmQdMVZKh3UZ8A0jw22LlqaZlrUf0akb0fs7le2HT47KV9y0JHC\  
tec9tjHUeBVmma504AofGcVXLbkqKv+Soax9GooHV0v+uxa8iwjAdTZKtqwKnKDX4jaR\  
+zotCsYi4BuB2JbkjnHG6NL7ubN+aNknwnzZnMKQZIH2Q7vSRyKTM8j90GLq7IP8q2NS\  
oc7iT//eAvb4oF6LaY7qebxQ6R0XCSRrXgpo+pw3ltfuUCuGzAXd4+wMZU3dlXsivhJ\  
PnTEjI/V6GmkRlfZ9XnYfj8SILETWk03dMFJh3LmUwkbRV+C3mL2GzjgQVTkvP82KDBL\  
DAR9iKyPkJnMnK9Ix/StVyJbGAtGp4jHnp+PSjz9ja4qI9jVRjGgIUQhw0DnI0fnplUn\  
Qhz3F9MQXMPSPvFw8M0xkUKsAcxQvxZGb5LkYByZ0Zr0/ipphwnE6zQuOva+8uTyBX\  
B9VR24tUITvlhy7SS6JrULrvTA+D/ZCiQKRx61iF6pU3BoC8fgA9D/AifiQnPz0SI5kx\  
FJfDTz1LWMjULQKBHFvRFLE9eFD0rnwAGx7Ppgpyc/KrLqVmcmj/96TYtoedp/iW4asfY\  
C2vs+GVyxVoumIdFPHJpencWbE/niZnVDaJCih1iqgXzDsI8bENh2B9cutDWX+bsHZSC\  
jSQb9YkGN+MoNiJLXmQHSJDyfPhzWPibds/lpS90ppPWIY+PpL0fzDSGFFWswQ4q5Phc\

```

pLWHx5lw9KSye+T86p6kadnBBTLTyfn0dG7Np09QKQObMN60MnybkVGx5nH9yLJlFmV\
0H+K0VZIKm4UzYV+RYfqXYtMqTQxeQ1U7L7o0H+6viErXuKj5rS3i+r1rdfECAGgCoq\
0mixATHISAHi2eSV5fk3r5xMkKSwwPIRuMt50+kkLRPUoLohTj7G1CnL602xwBdQMTUx\
4Jq5JBWnfB+U4D9n0si1DwikIhpaUyOoBeaWo4iFQiWVLwjeeQvY6zj66l70XsPHjZXg\
uCitsWfp5MYV3cLTkb80uCM/xhp4Y0Edobt6x3k1FD8vbh8g3YAG0Xe/U+Iz3klnpCt2\
R0Q2lGQa0JmL4nbQr3tqTLoXv4szaErFp/Xw05Cnt9DsBzN5DNrmfF6EDcfVf/hn8v9a\
wrg6Rfv8Jpys1YFpwLanhb3Wz+x1yaDsa54IdlF0FnyBxv8GppbFrMpVfx/nLAXGIocc\
WjcRKs0tBJUW/IoXeKOMPUD1wHR4dqUCEXsoexHJiNe5sH+akr6UID0bF70hhupBoiY9\
AzVXi5zXf2VdafyQrkGfKz4BEUkiqcaajHr1CF9ZJ+Mjdmfr3z0xyCmCAWir5ZL0BXDj\
T7sYCV3QjCz4a2mGvee9IxC9kSLapCq90UMAxnTLjJGQM/dlpgjDsjsCZX5wKdsnMs79\
60Z75BGD0C1dDINj4f5kHZmwwcmw/04mi/1RPBUABXse3Up3eJQOX2haZPqmY0+2PZTF\
exku9pETHtKcfSdRe1oJLmLb34JSogRmNp1eBxakcIL09huiFVtGVZng/pC/ryoJ/T9q\
9w4aV5H+4u2dHc29Vb77SasxCdRH0sDaLaPpesRXsrdJwbiz0gzRlIx+83o07NuhE+C\
kKf07cZMrFm8r8g1MlzDiFrTf3RTusMtiW6CVlVuTROpZFngqar5yeYPprpSELTQHswz\
U5AaY5Qd8tbky5ec+2/QkX0+cdyWhQUuBRpibwpRpD3x1yTgT4E91cwTFpvSLk54ZHf+\
D3EsZf0PYMN6d4jVdh9iv+0tCebnfMqP65wY26YBopSLtCXXb1anULRpLzPzRq99yKnt\
FM7gK1XnBAZoZBBqCyZw90HWmttIFWcmL4Wd5BxF9uZh2Y8gtcN8UKWHv43tsNBa7j/T\
ikIBSkIVI/6EQvyPW4YTdyz2V8RKHN5XcdpdWFaVhgSJMCI6Bm0Lwenhkmal7Sd247q\
uCtEow8qh+w7Jk4SxrmvJxd5sBnvz150KEaHPeWNNJW00bWEDT+0ZzzD8vMN1/GkbbB3\
s7UfcJXZbRu7HtQ+wHIbLBKVstX3hMonra+k6wS9KPhcAaC3IjZ7ZApSedKk1sW1SuDg\
l48YW2/cyS3LvmISQn9KPWK7yEpNqnV0vurn3ZF0G00eDjSXUjI+xIrRia5GQ1yb3lma\
nJnf2PdHcMmVr0wu4lMGno7a14nMRdnXkBU8bV0p8wF6Toz59hBJ3a/F+mP4/a19Ixra\
wiVVeEPgoi9QQ9NcLgQEFCoskA+EpcLK0FxV2rYI9JFNF/nDxP5nmGtnkmLfalo+pleH\
CJYS00TGKQr6X+Y65N0llx5nNwsnWkIUkCodoSt4Givdoe/S9JNiU8tW+jTBae2hNr9c\
glErCNKDYe1+T+Ldyr9rfOKm9LKnyTBsodgF4KI/hFh9Iv/i55DTwtqjpN0eQnPTB3/6\
+7KzTfSE9il5UMcP3zKKC2mAQvtyYxF3k0m24ZTwPs2LAPJkr/xtPH3BnGE/UfUDmVDS\
TBp9m049Nh9oDZvI4HKsY8auiyENk0ys67F9GTHh0YM0FgHyP5qk4/IR5YC3lnq7xx6i\
owebEJAY63htMytq+xd3cJyZR0lWBU0qvSpd/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": "z7u7GwhsjjnfHH3Nkrs2xvvw020Rcw5ymdlTnhRenjDdr00+nfXRVUZVy9q1\5zDn77zTgrIskM3WX8bqslc+B1fq12iA/wxD2jc1d6j+YjKCtkGH260R7vc0YC2ZiMzW\zG17yebt7JkmjRbN1N+u/2fAKFLuziMcLNP6WLoWbMqxoC2X00VNAWX3QjXrCcGU23Nr\imtdmWz5NrP43E592Sctt5M+SVlfgQeYv8pHmtkQknE8/jr7TrgNpuiV7nXmhWHTMJ4I\zoGXgq43odFFthboEdKNT/enyu+VvUGoIJ6cN8C/1B6o1WLYHEaL0BEIFFbAiAhZ/vnf\cUYMaVPqsDJuETsjetcE32kGCD7Jkume2t068DLiHb/2Z2JX8mkcbxFI6KrmXiRxxQj9\9LVn1fEzdf3Vfpcs/C3omsFGqmTpLDK+AvW/SWVkdI2NKq7hL/AyxlW2u2cqVERQZUTS\Z+ic6V8kZfxr3gRMnH0KuF5BtjleZ/yVvqqPjwP0ZegCKEL2Gd8duhcUde7CR55pil1o\UXy5AwgCcZTdEcJn10P0bGoots9T19gw1x4vnZCQUKVPZuZ1gIkGqDUYXS0lcNTjCMs\miFEmn0ZvB88jxULpb1vl9HoQ3ocM2oZu4AZRt9G/L07MwcuioFCWtAIau+2gqNan/Z\AS10l0j2N0LLtAa0xoF+Ctzscrt0ZMYGHmoQ9daHkpUvEq0c08hDtLp1nq3lQIIIfROQ\jcnS9vNKBu87COBjukZD+L8vV4zy8FN059MCSb9UCLwz2xvfdI1js9/J7hTGaVec8VPx\md42yPFRGw5Na1oefm8vW49EDmevc8AjAtwDirRBDfv9pX3+5S+M6jhteSLYvpKJXQT1\zs1379KvIHwkn9VHpA+PiUUw9TgF6x8xWEGSNl0o1Vn1xtM3givehjYxJ5p5/kBEFZI\DCyFzstAirJ2GadNhae+P1JFZzJWnX5jaLwzldquZwF3yTzNho4sgBA+fKqiXcgn2nw1\ vz0Dkbxr6cMaUool0eFScU1nAz1Z39W64Ltt2nEuYsORx/ht2RzJxxFc21X3nLeEDFCe\NkNDxQFBSfpZjKKgJtXEx23mp+CbBVMrbagsLnzsAGLYbnroVmATU5Iqr6LgYBpuFs+N\Rkq7ZXh6CZPukMGQbc0GuNw06NBuuMNHir5ayGk1ZBiW82C7Nu0hs2pLcgNqWMtt1+LW\8R96KyoSc784ZYAZ40QqvoySwmXQPBRTJ+wB0sVpGBLTxdY9Gw3pXeXN5nao340d2ZA\7YEMlqcTHCAv3F8B9ewl70fQlmg6bvdMuoVdVE+p0er7IAmWMRgviIzYv9sKEEQrCmua\2qL5xPSbD05KRf8ZAZ2B8LSCDR1nzXrQXZbXBKJivsCVQDuzxrwGE0gqRMpbk4f5GYCG\4i/08Knoru+jjf6wVQDYKfyz1QUGRlXHkGUGLxfv03r7UbJugycjV05kbGxhoZkq0q8z\ZEpkefvrrNoxeotw/z4QpjI8JlY97GDb0mGVHbmdHugjMtVTGhVJFBbPIinmR+emt70+\4q0r7yWRxCvt2lziWtpPBwaf/1XDnN5Gesex1gr1YrcTRNmB808b01sxLQmxcTt4eQ0/\LUkas7qTJ3AQTh0fDdtIpkqsthsBFy+WjSQuoXCymRcPi6MlpxJndDF32lCnL1ranV6e\F2ST0SYT+NwNDesMzTRmNbHUW5KAhu0k9WABTvcM5ba0Uq6i0a1NsFrcLag+KhxN6HPn\oobwJ/EsDi5S7TAl8WrjqIhZ8x6h9eRRXerpa0w/FYk+2MpWByP/98VE12/Ew0qAiPp\elAvUeM0lRkpG64bJsmYtHuNWgcv5Qiy7/eGw9ZpvB3J3G3jxvbyExqdFyDc067EKi\5WxDFPuZUjkfKpekNvzQuIrs49BzcRyMt5ndEVE21TPPfZ/R8B7Rxbn2LiK+hQc+cc9\pEEaWgwa0iMILcp/1CyY6Imd06RHsxwflMH7gej+hN41kaoEghI0l9kMGTlZbq5Pc8Pz\6F2LKTBMJWg9o/0blvilMH9EPblcLeF/bR1AZTUD6ZFdi2TxN6Epn3QVqeG/qPm1EBTF\Gw1V92m6/08Dd6zi1HPqwKbkHx4F567owoKHaM2imin0yVUpwxoRJRulRHMCB3tn8C4\ZpFl+sGV3Gip3tKlS7PKQkTqI6DMwxEbdrvtDy1sHZagpcLLDisA/yFT4RR2m3VNJR9P\6Nx3teqN1eg6RXmD/MlKCdWr1cjZ/6yeIQYwbr9CjItY/tLQX2gtAR1SX0h99UUBVv+Z\E03VOZ+Ecsc78lSB9G/6n6CFzlbk/HgAF+cu0yMbGnEM8W3mTUSpS4JBACwk5w0XWNNQ\DWVEdgzuLGHpQ+hYExDjVZrLELhkh8YgZA+7RXXUZHm/joNOGHUhpUG/bFo3ktnaILCu\xs0XMUBDC3VcitFFHsGK1svtcERDFxk1HA8pGa59jT0do6n3wEbnBDU1soKNFtpmcVke\

U13XpvuoW3BgCwJzBUCWvPs47DJRgX011bSaEYYlhTVaaShcvzgz46Akq0+Q7TjckDP\ /8uzsSQk0AbuhxWFQpSiBP80Z/U=",

"d": "z7u7GwhsjjnfHH3Nkrs2xvvw020Rcw5ymdlTnhRenjDUBg16FklHURz5btM5\ yrI5FQdWk+U2srVuSmfDV7EYG897mUFY35Z0WQ0mZ9XvIOKCh+GFF0k56b5F0Fq6xnV8\



UDQnFyY2JREUOHdiUjcUNxA1YxR3QiQ0BkE1AUBmFEOAUHZGBzQAU2dxVIgTQRV3U3g4\  
GGiISEYQhHRSWDIBQ2Z3UIIWdSV1EWhwBTYiWGI3VmJVI1UIU2REdUhHBoJ2gRhFUThy\  
BSQnhBIGI1AoMVB2MCNhUXQiNUGCKHgzUmQxU3dEgBhmQyIQgmFjdxY1dCJgGBSEB4Ij\  
CEJ0MBGIQWRRN3QjRmRSQWQIJgNjcdnMlJhJIU1MlJRd1NmF4dwhHIIdeYYcAhEcLBQ\  
JjESAiBwBQYzYlAIIOcBcoZFcGVkA2SDMCVTBjgzCAAzNnQGYHI1VwJzYxQRckBIZBV4\  
VxZmZiVLYXgHFRNjdEFIYVOFIVdhcnIINEhhIURjg0cxJ0SCIWYUUVcHJzdDUTASciAW\  
UiJAglIoQkIwNjMlcwACZxcHZVJA4EnNnWAZVVoBjNnNEcicTdyEEUHBVFjETETNjd0\  
YUFmFDVXNUcHFVJoE2AlVwFhMzc0dQckMYJUaBJ0JkUBdyd1AnZiJ3hYYkWAgYR1VziD\  
BERVh1NQAFIGWIhUZXCEQxIThyR1FIGGNTKAcwdRNBGDYEd2RVEwUDMzWAAhNQEEMYAS\  
hUSCgTJ3VIdXULFBFxFkhVCCZEABJjQCAxFGNkhHQzM1YSSHNLEBEmJkgWZCVwZHRSD\  
GDZzRCQinHe3IghDhSFoBYCHNFVHMxZAZTSGAVMUQkhBKIFRQEVBB0cHNGEiJVVScQQh\  
hzQiBzKBYRY4Q0R2MVUlDWVCikQYEEgFYEREY2NERyFVDHJzdAZHBmhmdSCFIgh1IwGB\  
AxNzFjEoUWFCF4AhZRJSEROEETHxAGYBJTgUcUdFJBOFRmcVYnZUUFcAY1AnZEZBVThS\  
ECBQYBNGeAdzQ3KGhIE3ZiFxQoVCgQBjEFdFcIV0IAaFcgI0iAgAUVQAHEInQWECY1I4\  
E2U0MkAXQSZkdSRRc2I4BjeiSGd4hgQEEDcTRmhFd3MBmY2gERxNwiISAVkFVETGCCi\  
gYhzRXByGBgzKGVXJUhoGEY1hgFmZWgmEWYVVoISd4Vlhid4VUMHgSZXhXUSaCgmJ3Eg\  
MQIzIDIThwRSARQhBFB2QQBjEgIoEHN1BhMCiIIBULZWCHdBMZYFQoQ2UUJXJCFnZyFD\  
RTFUUTQoQmUjB0aHJXJHeDZLJGBCExATEUEDg3BTACHWImN0AWcFB3IUGBEARxVUREdX\  
QzhVBEBBF4UgcRhCg4gUcORkdYCAIUQBRUJUyJfjEBYUHSBjIyV2cXBYckeimic1N4ED\  
gDUGVSCHhCYURXcQB10Dg4hDJFJ3Jld2gyYnQYcLRjE3VwCQNXJBYn0GhYFoU2dHATAw\  
OEeBUGQjJHcDgUJTRXBxJ3IAEjEXhXeEEemghIAAGdjUVBndnI2FCAVcyV4cxIyZ2dYE1\  
ZEiHcYJYaCcXQXJCaER1coIDRWJkcSNhEVF3JGOCQkYWYkcndRA1QTh0MFgzIyVocYJY\  
cwIAFRNiE1VAUECBI3MzQiZkUhr1cid1NSAXFXN0gUNnRCV0ISNmYnB3NIZyMIQWEVVz\  
ElEohnQyKBNoY1cCdmdjVYiGhVULA0CHMTZIURBgR4aIJwJQJLIDMYhwNGNwiGcIBUDa\  
NXMBETUGICJyMkMIN1BAIAB4gEMDUwhndFJkQDEgRRMld4EzhRM0ZhGAYHMGZ4NhEEhn\  
U2I2VicBBXZUFUNoczcmbZBnUEBxUWCdG4RHUXZS0EZogABHRAISVEAZdoQhRmBgEWE\  
Z4U1dgQ2gjgQUjMScgIIFQR3YFczhTYoB1IiVCYGBAVmVAFIdhRmZ1InhwdTRjJjNhVx\  
IzcWgjFLFCaChLIWUySIaDFwAWV3RAIxg2QWYgAYMUjP7wmwOwPp7Ukl3L1KaLY/6dN4\  
dBr1AYS8JnkVq6pPeBf07ccX95SrVfA07EX7RVEYyhVR9Q0QyEpLBUMcfcfnHCZWKM0o\  
0BF7BXiWMR9BQo4ybtPJGKQ+IZyCKUJVRhZ+uae182qYcBKFMd00zXi08kAa98eUy6SR\  
pPFkPD6D+xXgtJ0FWtYnp1Jy2aIG3HqMiThoSdVIvccGkf94gpVWTMeJQsQpgq7dAJiJ\  
5JOMQjk7JIHcIzxb4T8sQHZA55MFfvM7Hus/8FUX7NfIN1JRmc2zHL/7kdfCFswG67iW\  
U4ob2kTwdKzPvOL+d3e+A0E0PihJ4vVJA0jhWm02fIFNvFhNqPh0MSiSkatPGbSVdqQ1\  
PsG6C+1YqMrTM7KFr4hTQM8a3+ta0sImMjXSSPDkVeuJFq1rw642SJJx8yZTXVe8g75D\  
ZTYghbeX5LLzaVkt9mZS7cW16Zy+C3MwnWDrGQ6hUDxYaYJp7SOGJHepcmVV214oD6nw\  
5QprgpGIxVcdXQU00fhKweryDko0Ij+uqk7NYDv0t8zANphYcE3v+6yVFyYh3eg7DYRJ\  
rIzIcbaG91ySv2iRRC+cWaymH6xuqaHRwZu/p962/u8/c3rITJzCoVc+ObnZ5oItZfBe\  
AYFhLBx7PvPdBULXyCqmtk0tnT/jnaCUVxtGeaIeQmmeM4yPq3d5uWBf0vIyuPmfBSKd\  
Y0NETGlsaoQuqFp0kCmQdMVZKh3UZ8A0jw22LlqaZlrUf0akb0fs7le2HT47KV9y0JHC\  
tec9tjHUEBVmma504AofGcVXLbkqKv+Soax9GooHV0v+uxa8iwjAdTZKtqwKnKDX4jaR\  
+zotCsYi4BuB2JbkjnHG6NL7ubN+aNKnwnzZnMKQZih2Q7vSRYKTM8j90GLq7IP8q2NS\  
oc7iT//eAvb4oF6LaY7qebxQ6R0XCSRrXgpo+pw3lftuUCuGzAXd4+wMZU3dlXsivhJ\  
PnTEjI/V6GmkRlfZ9XnYfj8SILETWk03dMFJh3LmUwkbRV+C3mL2GzjgQVTkvP82KDBL\  
DAR9iKyPkJnMnK9Ix/StVyJbGAtGp4jHnp+PSjz9ja4qI9jVRjGgIUQhw0DnI0fnplUn

```
Qhz3F9MQXMPLSPvFw8M0xkUKsAcxQvxZGb5LkYByZ0Zr0/ipphwnE6zQuOva+8uTyBX/\
B9VR24tUIItvlhy7SS6JrULrvTA+D/ZCiQKRx61iF6pU3BoC8fgA9D/AifiQnPz0SI5kx\
FJfDTz1LWMjUlQKBHFvRFLE9eFD0rnwAGx7Pgpyc/KrLqVmcmj/96TYtoedp/iW4asfY\
C2vs+GVyxVoumIdFPHJpencWbE/niZnVdaJCih1iqgXzDsI8bENh2B9cutDWX+bsHZSC\
jSQb9YkGN+MoNiJLXmQHSJDyfPhzWPibdS/lpS90ppPWIY+PpL0fzDSGFFWswQ4q5Phc\
pLWHx5lw9KSye+T86p6kadnBBTLTyfn0dG7Np09QKQObMN60MnybkVGx5nH9yLJlFlmV\
0H+K0VZIKm4UzYV+RYfqqXYtMqTQxeQ1U7L7o0H+6viErXuKj5rS3i+r1rdfECAGgCoq\
0mixATHISAHi2eSV5fk3r5xMkKSwwPIRuMt50+kkLRPUoLohTj7G1CnL602xwBdQMTUx\
4Jq5JBWnfB+U4D9n0si1DwikIhpaUyOoBeaWo4iFQiWVLwjeeQvY6zj66l70XsPHjZXg\
uCitsWfp5MYV3cLTkb80uCM/xhp4Y0Edobt6x3k1FD8vvh8g3YAG0Xe/U+Iz3klnpCt2\
ROQ2lGQa0JmL4nbQr3tqTLoXv4szaErFp/Xw05Cnt9DsBzN5DNrmfF6EDcfVf/hn8v9a\
wrg6Rfv8Jpys1YFpwLanhb3Wz+x1yaDsa54IdlF0FnyBxv8GppbFrMpVFx/nLAXGIocc\
WjCRKs0tBJUW/IoXeKOMPUD1wHR4dqUCEXsoexHJiNe5sH+akr6UID0bF70hhupBoiY9\
AzVXi5zXf2VdafyQrkGfKz4BEUkiqcaajHr1CF9ZJ+Mjdmfr3z0xyCmCAWir5ZL0BXDj\
T7sYCV3QjCz4a2mGvee9IxC9kSLapCq90UMAxnTLjJGQM/dlpgjDsjsCZX5wKdsnMs79\
60Z75BGD0C1dDINj4f5kHZmwwcmw/04mi/1RPBUABXse3Up3eJQOX2haZPqmY0+2PZTF\
exku9pETHtKcfSdRe1oJLmlB34JSogRmNp1eBxakcIL09huiFVtGVZng/pC/ryoJ/T9q\
9w4aV5H+4u2dHc29Vb77SasxCdRH0sDaLaPpesRXsrdJwbiz0gzRlIx+83o07NuhE+C\
kKf07cZMrFm8r8g1MlzDiFrTf3RTusMtiW6CVlVuTROPFngqaR5yeYPprpSELTQHSwz\
U5AaY5Qd8tbky5ec+2/QkX0+cdyWhQUuBRpibwRpD3x1yTgT4E91cwTFpvSLk54ZHf+\
D3EsZf0PYMN6d4jVdh9iv+0tCebnfMqP65wY26YBopSLtCXXb1anUlRPlzPzRq99yKnt\
FM7gK1XnBAZoZBBqCyZw9OHwmttIFWcm14Wd5BxF9uZh2Y8gtcN8UKWHv43tsNBa7j/T\
ikIBSkIVI/6EQvyPW4YTdyz2V8RKHN5XcdpdWFAvhgSJMCI6Bm0Lwenhkmal7Sd247q\
uCtEow8qh+w7Jk4SxrmvJxd5sBnvz150KEaHPeWNNJW00bWEDT+0ZzzD8vMN1/GkbbB3\
s7UfcJXZbRu7HtQ+wHIb1BKVstX3hMonra+k6wS9KPhcAaC3IjZ7ZApSedKk1sW1SuDg\
l48YW2/cyS3LvmISQn9KPWK7yEpNqnV0vurn3ZF0G00eDjSXUjI+xIrRia5GQ1yb31ma\
nJnf2PdHcMmVr0wu4LMGno7a14nMRdnXkBU8bV0p8wF6Toz59hBJ3a/F+mP4/a19Ixa\
wiVVeEPgoi9QQ9NcLgQEFCoskA+EpcLK0FxV2rYI9JFNF/nDxP5nmGtnkmlFaLo+pleH\
CJYS00TGKQr6X+Y65N0llx5nNwsnWkIUkCodoSt4Givdoe/S9JNiU8tW+jTBae2hNr9c\
glErCNKDYe1+T+Ldyr9rf0Km9LKnyTBsodgF4KI/hFh9Iv/i55DTwtqjpN0eQnPTB3/6\
+7KzTfSE9il5UMcP3zKKC2mAQvtyYxF3k0m24ZTwPs2LAPJkr/xtPH3BnGE/UfUDmvDS\
TBp9m049Nh9oDZvI4HKsY8auiyENk0ys67F9GTHh0YM0FgHyP5qk4/IR5YC3lnq7xx6i\
owebEJAy63htMytq+xd3cJyZR0lWBU0qvSpd/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](#) (#name-sign). 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:

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpzZW50b3kiLCJ1eWciOiJ0bmV0ZX

SXTigJlZIGEGZGFuZ2Vyb3VzIGJlc2luZXNzLCBGcm9kbywgZ29pbmcgb3V0IH  
lvdXIGZG9vci4gWW91IHNOZXAgaG9hZCwgYW5kIGlmIHlvdSBk  
b24ndCBZZWVwIHlvdXIGZmVldCwgdGhlcmXigJlZIG5vIGtub3dpbmcd2hlcm  
UgeW91IG1pZ2h0IGJlIHNOZXB0IG9mZiB0by4

cu22eBqkYDKgIlTpzDXGvaFfz6WGoZ7fUDcfT0kk0y42miAh2qyBzk1xEsnk2I  
pN6-tPid6VrklHkqsGqDqHCdP608TTB5dDDItllVo6\_10LPpcbUrhiUSMxbbXU  
vdvWXzg-UD8biiReQFlfz28zGWVsdI NAUf8ZnyPEgVFfn442ZdNqiVJRmBqrYRX  
e8P\_ijQ7p8Vdz0TTrxUeT3lm8d9shnr2lfJT8ImUjvAA2Xez2Mlp8cBE5awDzT  
0qI0n6uiP1aCN\_2\_jLAeQTlqRHtfa64QQSUmFAAjVKPbByi7xho0uT0cbH510a  
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.\n\nYou step onto the road, and if you don't keep your feet, there's\n\nno knowing where you might be swept off to.",
  "signature": "2As8T1AHenWzLuTojcAYFDnT05n4bmDGIWenHqoXVizL7311HtVg\n\n7PEJHYmpc1fIvFNrm0xJt0asD5bQk3ZY8WuEQDUjsn4j+zbyob8MPQI5u3p5ZkqLLhG\n\n6Q8p1q0Hd5voY4a78vNxFJpYsETc0BECAft196z5hml2VjuDBqI7W4ju/iDKambJIDz\n\nNLYgYinNyPcHjlfBP7aCf0qGBAQrWuVgrAkdeM+uH6djaXW25+FeU4Lg1u0IBPrcj\n\nZJ04M07j7BmiuHJDB74QG/ifVqnvr4z2a1MWHjjR7nPPr2CIKpuRthSpNWYVTRSN3mM\n\nv0GjVLyaghJpmUmewhja0Ci3iP7c59vKatGYiLPPEapsbn7ypIo1Bod/R2PZR0zeool\n"
```

\d9k30VmGsVLkJ40EIFn\A8epv+bJISApZWGuU6NBP8vr4UB2D9DRd8zwvd/vI0BWdq\  
\nfglX4x18lWe8Tnd+21UC9n4zUb+KQlo9RR14VfXE0t9g5a0IzCWjAN+Oz8vqJ/ZwgH\  
\zZotZNF+nZehTFPcPLM3dpoUkEI391VH3QQ6VTYfbMW9wGJ6UnylxZFEzNCnMFF9Qhs\  
\7Xehy4yEDgJBFYIvbTRCFD+EbZbWQAnLKsm7UXXBR7HdUsJMhTkwdffGziWJBTf1UsG\  
\tqaCF1bvXgbcCSe0XGhc0QkQKwwj3kNwY9/hnhH1bn7kyySqaI+W4Ph3pKwRb38sCS\  
\Gb3ryptI8zez0JR+lClWnu18noJjGincZq7jCGMiMCRFpzUV6rpY/FiM26IpZ8M9ShF\  
\BHsTN7KGpyIqG6Yc3Gz0J/4ir7V3I3wYguK7iBUiuTM+OKwxtM75carZJPX/2lkn2Hh\  
\TC+JVb2/yaHS2oDrLCwQosNhA/cB/cm+YmYgHc3KQwrZ/3Axr6weUSrWJ8qV+vJ5QKK\  
\a1+CLrkVGJX6vh+ps1NB5EV9yyMhBXAbBJZ3K+dGed3G7Vj/qF2kbnIULSiEP1f6LsH\  
\XASuVLTU6qG0rTaCMYKwaAc5R0wAgyZmPXuOUwyCtNFb0+S73uX5/N2drrUPdXiURW+\  
\luFKCtaNimU8myoz6YkoY234kz8pedST8eqBZAioe8HeYEKtZSAyYov4YfLgkqHqJG6\  
\ycD2uA33kwnMim+jg/hIrWAIYYP9R90KECTvFR877RtfFgfn3+tZjWlmsxHZ5pNsTIA\  
\dNR+VmNpouZkQ0dgHuFLztyAnCaumL38LHYHFHj2boa0zYsMGw8WtpEQ3+BNgoanNax\  
\dJ5THRRmhvMS3EDwanERimsZ6ZjdK8uchuVhytNiikvBwEFWYIyoK9uUMBoEfDje4DX\  
\wIAefXYCqPK8eXhL+9qDLxADlDQbCu+Ey3whX/r4r2Q6l+34HpRrn3g5ok+Gt0/3ni9\

\dYiIYpcXYfhMGDoXJLZ3IMkK7L6e5u4/Wye7lot2B5ekSGRrkLkju+bTIkppxbTU4Pi\  
\n40qbD91sRzw2/GzZmJsfcFaKbj5dhoNWyh5cZr1PqsxMI5EdXSxJ69VWf8e+h4iPoB\  
\YS1JnUjhicVWslpA1rdAvTkAsVY8rC22e09Hxzbbk/E7bt3iLDpekbbQAghZ31AwDv5\  
\KEG72bBbXIYHzPvhJzrlS2LR0XKTJVd8tAx0SdxQD0t8tE2eKpmWZ38MJfRIxt2Rzol\  
\p+bPkrR++pMLRrpViekVpZl/tlEojImN05rLqxZhLxvZ0yDfcT37jc1oqire527/Y9L\  
\3k894eHNYcXxjb0LGGPDeLuTSEX+afHZLNbd93Qa5VTmLwsPxEW/Erua6nXUrAR/87P\  
\0gIyce3h3sl5jzCXsQm/iODgyn7PTEo5ksQCFRPiyXq5xgiXGKGGkqTGg280hdbby+lN\  
\DPnNHU2J0F6GLTqwk3qGbBLzDGIMR2sePGpxZ/pecoX7yn5bT0f4iY10CyLo5nEgSeb\  
\JdBjH0ZU+QodLRN0cnenLmP1oNK2yCuT9uIALWH9C1CLhBiEOfoIs9/r1W1XHPiPsX7\  
\c21w+B1IPfzUX1cVdndnNo4XdHl9CH1tYJDLr8LfeuYnz+bnafLqEUryTc8zU4A+qB\  
\SIDDJefCbmdsTrdqzGT2J89MKVi0ogy3qJzyt3jo04xq+Q30Gjb0FJikyJEqUm8BmX\  
\d3ctGfzsEr+5w7fDRco40/tDQUSH0q0W0sPkhueLLqKDziJXwhPQI42miVN2A4+OAS4\  
\f2uTgpDNn1gIfH2+d0CkBJlhZeA1Tgrp8FHQxca05kut6cTLrL7CSBqINa7Khe1zyXa\  
\PZG/tXUk+iv0BYT92b7CRNmglqhE0G8V3q3QrB6EePYa1WxRQ7ij4rRcQWcj66A1hZ5\  
\KjDUVJh+02cZTFrv97wM/im3vb3dbiSxAiQExSa2KATfLI2oS+y7RlRNJ+9nF/vTaFc\  
\0H0dKfmuJAUKAcyk/h0Quvdaf9jxEcstj95mva+HkIqPuFifidlvGiafKr4fHZryp1h\  
\g7QUtDRU2a4BRfzcLz6PK0BFV3xVI7qoQbKEqQyldv8mZRd0LBRKprxHW7PdUqutH2V\  
\GEmZ4UuCYXT11UweBx2W9lHrQX+xaKAjTu6oLYIOvmFVCUr4mCrYRcLZnzw0RcsqIL4\  
\G88x8r5aeilL4lsQZ03kNotR4n0qzFVRU2+EX07QJFm+NKxB7aRZ5oH+dSy+Ye6aMeG\  
\Epv491LU0LVnZMBP2eUhoEo0gimmZGtUobjRdLuYyNiJfJzVkjwF3gYQtY59zb+46N\  
\SzvWUqpFUG80Vswns8GNAQ5hfLoH80GGohT+UvoqvpTEXhiAAfstT/EQRHLZrYpXHJI\  
\YaICW+6uo9ixL0oWkfi0HLYaXyNkaFKHQ5ZbPaP45dbWq/dqXdrRe2YU8AqdjCxyyz0\  
\lyZR6zH9wHj0k1AI0HvnKZ/B2v4bS8YAtnZ1zgKb0vM4qqSIFETfr8N4yIteumHEznP\  
\prD7Gr6W2VCS/0FXnQt5y0QC8z4ffrnggwPjczfzCRsknktQB1q6Cx8KU0ipf+Rh0vs\  
\HnNN3qJZmxz6YCvo2M7fxJtyRvm34UEVaj8QKXrmzX70Y9rDl6wEhhvSThaeq4dcfAC\  
\vczGXWgCLB10gl+Iz6hVDTgCx7bC2BQ2oHtzSDc+v/UuJewvVaIL9tn4CtMZU86f3Zc\



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 algorithm 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 assurance - 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 implementations 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 (<https://datatracker.ietf.org/doc/html/rfc8812#section-3>)

### [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 persistence - the implementation 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 (<https://eprint.iacr.org/2018/821.pdf>) through side channel mechanisms.

#### 6.3. Randomness considerations

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

Internet-Draft

post-quantum-signatures

March 2022

## 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:

Prorock, et al.

Expires 6 September 2022

[Page 24]

Internet-Draft

post-quantum-signatures

March 2022

- \* 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](#) (<https://tools.ietf.org/html/rfc7515>)
- \* JSON Web Encryption (JWE) - [RFC7516](#) (<https://tools.ietf.org/html/rfc7516>)
- \* JSON Web Key (JWK) - [RFC7517](#) (<https://tools.ietf.org/html/rfc7517>)
- \* JSON Web Algorithms (JWA) - [RFC7518](#) (<https://tools.ietf.org/html/rfc7518>)

- \* JSON Web Token (JWT) - [RFC7519](https://tools.ietf.org/html/rfc7519) (<https://tools.ietf.org/html/rfc7519>)
- \* JSON Web Key Thumbprint - [RFC7638](https://tools.ietf.org/html/rfc7638) (<https://tools.ietf.org/html/rfc7638>)
- \* JWS Unencoded Payload Option - [RFC7797](https://tools.ietf.org/html/rfc7797) (<https://tools.ietf.org/html/rfc7797>)
- \* CFRG Elliptic Curve ECDH and Signatures - [RFC8037](https://tools.ietf.org/html/rfc8037) (<https://tools.ietf.org/html/rfc8037>)
- \* CRYSTALS-Dilithium - Dilithium (<https://www.pq-crystals.org/dilithium/data/dilithium-specification-round3-20210208.pdf>)

## [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

Prorock, et al. Expires 6 September 2022 [Page 26]

---

Internet-Draft post-quantum-signatures March 2022

- [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](mailto:osb@zurich.ibm.com)

Christine Cloostermans  
NXP  
Email: [christine.cloostermans@nxp.com](mailto:christine.cloostermans@nxp.com)