Workgroup: COSE
Internet-Draft: draft-ietf-cose-falcon-01
Published: 9 July 2023
Intended Status: Standards Track
Expires: 10 January 2024
Authors: M. Prorock
mesur.io
O. Steele
Transmute
R. Misoczki
Google
M. Osborne
IBM
C. Cloostermans
NXP

JOSE and COSE Encoding for Falcon

Abstract

This document describes JSON and CBOR serializations for Falcon, a Post-Quantum Cryptography (PQC) signature suite.

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

This document registers key types for JOSE and COSE, specifically NTRU.

Key types in this document are specified by the cryptographic algorithm family in use by a particular algorithm as discussed in RFC7517.

This document registers signature algorithms types for JOSE and COSE, specifically FALCON1024 and others as required for use of various parameterizations of the Falcon post-quantum signature scheme.

Note to RFC Editor: FALCON is described and noted as a part of the 2022 PQC Selected Digital Signature Algorithims As a result, this document should not be proceed to AUTH48 until NIST completes paramter tuning and selection as a part of the PQC standardization process.

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 10 January 2024.

Copyright Notice

Copyright (c) 2023 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. 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. Falcon

3.1. Overview

This section of the document describes the lattice signature scheme [Falcon], the "Fast Fourier lattice-based compact signatures over NTRU". Falcon is based on the GPV hash-and-sign lattice-based signature framework introduced by Gentry, Peikert and Vaikuntanathan [GPV08], which is a framework that requires a class of lattices and a trapdoor sampler technique. For the class of lattices, Falcon uses the well-known NTRU lattices, while for the trapdoor sampler, it uses a new fast Fourier sampling technique [DP16]. The underlying hard problem is the short integer solution problem (SIS) over NTRU lattices, for which no efficient solving algorithm is currently known for both classical as well as quantum settings.

The main design principle of Falcon is compactness, i.e. it was designed in a way that achieves minimal total memory bandwidth requirement (the sum of the signature size plus the public key size). This is possible due to the compactness of NTRU lattices. Falcon also offers very efficient signing and verification procedures. The main potential downsides of Falcon refer to the non-triviality of its algorithms and the need for floating point arithmetic support.

The GPV framework, which underpins the Falcon design, is proven to be secure in the (quantum) random oracle model as long as the SIS

problem remains intractable. Falcon requires an adaption of this prove to account for the fact it uses NTRU lattices.

Falcon brings several advantages over other approaches to signature suites:

- Post-quantum secure as long as the NTRU-SIS problem remains intractable.
- Compactness: Falcon aims at minimum signature plus public key sizes. This should be contrasted with hash-based signature schemes (e.g. SPHINCS+), which minimizes public key sizes but suffer from long signatures, and multivariate quadratic schemes, which minimizes signatures sizes but suffers from long public keys. It also offers substantially shorter signatures than other lattice schemes while public keys are about the same size.
- Efficiency: Falcon can produce thousands of signatures per second on a common computer, while verification is up to ten times faster. The operations in Falcon have O(n log n) complexity for degree n.
- Side-channel resistance: Falcon may still have an important limitation regarding side-channel attacks due to the hardness of implementing discrete Gaussian sampling over the integers in constant-time. This gap that may have recently filled, but is under active investigation.

3.2. Core Operations

Core operations used by the signature scheme should be implemented according to the details in [Falcon]. Core operations include key generation, sign, and verify.

3.3. Using FALCON with JOSE

This sections is based on CBOR Object Signing and Encryption (COSE) and JSON Object Signing and Encryption (JOSE)

3.3.1. FALCON Key Representations

A new key type (kty) value "NTRU" (for keys related to the family of algorithms that utilize NTRU based approaches to Post-quantum lattice based cryptography) is defined for public key algorithms that use base 64 encoded strings of the underlying binary material as private and public keys and that support cryptographic sponge functions. It has the following parameters:

- The parameter "kty" MUST be "NTRU".

- The parameter "alg" MUST be specified, and its value MUST be one of the values specified the below table

alg	Description
FALCON512	Falcon with parameter set 512
FALCON1024	Falcon with parameter set 1024

Table 1

*The parameter "pset" MAY be specfied to indicate the 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" "FALCON" one of the described parameter sets "512" or "1024" MUST be specified. Parameter set "512" or above SHOULD be used with "FALCON" 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 "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.

Sizes of various key and signature material is as follows

Variable	Paramter Name	Paramter Set	Size
Signature	sig	512	666
Public Key	x	512	897
Private Key	d	512	1281
Signature	sig	1024	1280
Public Key	x	1024	1793
Private Key	d	1024	2305

Table 2

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

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

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

*The "alg" field MUST be present, and it MUST represent the algorith and parameter set.

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

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

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

3.3.2. FALCON 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
NTRU	FALCON512	512
NTRU	FALCON1024	1024

Table 3

3.4. Using FALCON 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
FALCON512	TBD	Falcon with parameter set 512	No
FALCON1024	TBD	Falcon with parameter set 1024	No

Table 4

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

Name	Value	Description	Recommended
NTRU	TBD	kty for NTRU based digital signatures	No

Table 5

4. Security Considerations

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

Care should be taken to ensure "kty" and intended use match, the algorithms described in this document share many properties with other cryptographic approaches from related families that are used for purposes other than digital signatures.

4.1. Falcon specific Security Considerations

Falcon utilizes floating point multiplications as part of fast Fourier transforms in its internal operations. This is somewhat novel and care should be taken to ensure consistent implementation

across hardware platforms. Well tested underlying implementations should be selected for use with JOSE and COSE implementations.

4.2. 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.

4.3. 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 Falcon 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 - care should be given in implementations to preserve the property of uniform sampling in implementation.

4.4. Randomness considerations

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

5. IANA Considerations

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

* Name: "NTRU"
* Description: NTRU family post-quantum signature algorithm key pairs
* JOSE Implementation Requirements: Optional
* Change Controller: IESG
* Specification Document(s): Section 3.1 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): "NTRU"

*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: "d"
*Parameter Description: The private key
*Parameter Information Class: Private
*Used with "kty" Value(s): "NTRU"
*Change Controller: IESG
*Specification Document(s): Section 2 of RFC 8037

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

*Parameter Name: "x"
*Parameter Description: The public key
*Parameter Information Class: Public
*Used with "kty" Value(s): "NTRU"
*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: "FALCON512"
*Algorithm Description: FALCON512 signature algorithms
*Algorithm Usage Location(s): "alg"
*JOSE Implementation Requirements: Optional
*Change Controller: IESG
*Specification Document(s): Section 4.1 of this document (TBD)
*Algorithm Analysis Documents(s): (TBD)

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

*Algorithm Name: "FALCON1024"
*Algorithm Description: FALCON1024 signature algorithms
*Algorithm Usage Location(s): "alg"
*JOSE Implementation Requirements: Optional
*Change Controller: IESG
*Specification Document(s): Section 4.1 of this document (TBD)
*Algorithm Analysis Documents(s): (TBD)

6. Appendix

6.1. General References

*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

[DP16]: Leo Ducas and Thomas Prest. Fast fourier orthogonalization. In Sergei A. Abramov, Eugene V. Zima, and Xiao-Shan Gao, editors, Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC 2016, Waterloo, ON, Canada, July 19-22, 2016, pages 191-198. ACM, 2016. [GPV08]: Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, 40th ACM STOC, pages 197-206, Victoria, BC, Canada, May 17-20, 2008. ACM Press.

6.2. Appendix A. Acknowledgements

We would like to especially thank David Balenson for careful review of approaches taken in this document. We would also like to thank Michael B. Jones for guidance in authoring.

6.3. Appendix B. Document History

-01

*Added Acknowledgements
*Added Document History
*Updated test vectors

-00

*Created draft-ietf-cose-falcon-00 from draft-ietf-cose-post-quantum-signatures-01 following working group feedback

6.4. Appendix C. Test Vectors

6.4.1. NTRU FALCON512

6.4.1.1. publicKeyJwk

```
========== NOTE: '\' line wrapping per RFC 8792 ==========
{
  "kty": "NTRU",
  "alg": "FALCON512",
  "x": "TUlJRGp6QUhCZ1Vyemc4REJnT0NBNElBQ1g3ZElvVGR5ajZmN3VmME5vbWJP\
ekVtNG84U3JCMytYdWFveTA1cFFjVkpsWUF0VFkzcWsxcUNZcnJOdU9saXI5OC84dHh5\
SDZlRXNCNUxVVXBPbjZQWUtZclZFVjFQM1ZHYmd5UmFKaDNkUldrVUpDWTVTVkEra2JB\
ZUFiM29MVUtMZktTUk9UZGN3MStpL1BkamFNWXpwb2hWOEJqa003REhFTGNqTUhxTlJn\
SkRDdE95RXAvNTBXanl2Vm9XRXZWVTlZemxjamhMeWw5S29XTklZb1UrWUJGMnRqS2ts\
VXlsWlBma0F6c3QwSk5FSW0xWkIxS2xWR3ZpYks2WThuaXM0T2F2OHFVa1l5UGdsTzYr\
cGtJVkdDV09xdHIxclpwZ0JZWm5CbDlJVytrekpuVW5ZbFQ2bDBpVEhtT0MxbFNJZnh6\
SXBQekpFU1pHQldxQmhmb1BKa0tFVktvV0VFVFFseUNBVXlybTM3Qm9oMGZ3cEdweS9W\
NDBnNTFZU2xveTkrbEJiaWhnSUpReXdvdUg4YnVrZGtKRGtOYlZNMklkeERvc0hOVEx4\
M2lYY0RocjZ4eU9jT2w0TmRWYU9CUFdzek9yWGtOSWdUenp5UWNRWW41cFF5MFR1WHd0\
QW1PN2grRUYxQ1NEdlFhSUZkRDl5ejdWTzFyVmdvY1ZiMHc3R0xyMEZoWHVSb0JsNnpp\
cnJRenhJUmFiVjlZamFYekpibC9vdlNKMCtnOEttaUlvUE9LajQwVTBOc3phaVcyNldY\
RmdzNW1rbUxQQTcxR29uQldIQkNua0ZkbzdjM0FJbXlwUjhJQmdObXZvR0RkZXNSUGJW\
cEdNdUJUOUJyMkxBYjFNbnlTRmExK2xxTkVteW9Da21acUhOM2tGaEp3OUFCZDN6QUlm\
Z29ldzhkRVBIRm14SnVOVklpM3FqbUVxU0U4QmhQUUdmRjVMSUhKRkpEQk1rUjgrOHZF\
eG9YMXpsRFZuNzFHMHZkRjBxUGJnV3BKbFZhQ3M1VHlSRlgxR0NZbmJvY2lBR3dKUito\
ekE5SnNKeGJ4TDVBUlNWR3J3Zlg2UEVJM0gvNUpreU1jSEdnNFdoZmJHbzQ4TEdFVlFx\
YVpXUkJHNG96aU00TU9HWGFPTHBZNFJSKzB0TVRnT0lpK0sxUWl1TFBkbjhUK0dJK1c5\
cDFTNTVBU0VHeGRBRFpKZmFSRkZaQll6b1VaeWJtRmt1WmJuUkVGOXh2QzlBbnc3L3Ay\
cG9HUXRwUGpkVlFZc1MxOUVjVWlCQmRhdU1HLzEya1FDTEhZMjlKL0orU2tFR0hKVTFr\
OWNRSlNvUTJRYmtpVm5RaVRWQzMwSzdtbkpKOW14U2NOSURBQUlOTFhUNHVDdU9pR20v\
c1RWMHI5TXJJcjRjTzRNYXBvTTBWbmJZTDA2d25SL2pzTTVPRFhxMVJ5blRMbGFXN29z\
cDA3TUFmQUhDd2V4M1p0bkFNZUFaUVJ4T0Yza1F3QU1reElITE01aThzMmFHTjJtV3FX\
VHR5cGlqMEw4eExtUzBhNXBUR2FWb1lhbnZrV1M1UC9CRjdER3hEUmtxdm1SbERyVXdo\
UjY",
  "use": "sig"
}
```

6.4.1.2. privateKeyJwk

```
========== NOTE: '\' line wrapping per RFC 8792 ==========
{
  "kty": "NTRU",
  "alg": "FALCON512",
  "x": "TUlJRGp6QUhCZ1Vyemc4REJnT0NBNElBQ1g3ZElvVGR5ajZmN3VmME5vbWJP\
ekVtNG84U3JCMytYdWFveTA1cFFjVkpsWUF0VFkzcWsxcUNZcnJOdU9saXI5OC84dHh5\
SDZlRXNCNUxVVXBPbjZQWUtZclZFVjFQM1ZHYmd5UmFKaDNkUldrVUpDWTVTVkEra2JB\
ZUFiM29MVUtMZktTUk9UZGN3MStpL1BkamFNWXpwb2hWOEJqa003REhFTGNqTUhxTlJn\
SkRDdE95RXAvNTBXanl2Vm9XRXZWVTlZemxjamhMeWw5S29XTklZb1UrWUJGMnRqS2ts\
VXlsWlBma0F6c3QwSk5FSW0xWkIxS2xWR3ZpYks2WThuaXM0T2F2OHFVa1l5UGdsTzYr\
cGtJVkdDV09xdHIxclpwZ0JZWm5CbDlJVytrekpuVW5ZbFQ2bDBpVEhtT0MxbFNJZnh6\
SXBQekpFU1pHQldxQmhmb1BKa0tFVktvV0VFVFFseUNBVXlybTM3Qm9oMGZ3cEdweS9W\
NDBnNTFZU2xveTkrbEJiaWhnSUpReXdvdUg4YnVrZGtKRGtOYlZNMklkeERvc0hOVEx4\
M2lYY0RocjZ4eU9jT2w0TmRWYU9CUFdzek9yWGtOSWdUenp5UWNRWW41cFF5MFR1WHd0\
QW1PN2grRUYxQ1NEdlFhSUZkRDl5ejdWTzFyVmdvY1ZiMHc3R0xyMEZoWHVSb0JsNnpp\
cnJRenhJUmFiVjlZamFYekpibC9vdlNKMCtnOEttaUlvUE9LajQwVTBOc3phaVcyNldY\
RmdzNW1rbUxQQTcxR29uQldIQkNua0ZkbzdjM0FJbXlwUjhJQmdObXZvR0RkZXNSUGJW\
cEdNdUJUOUJyMkxBYjFNbnlTRmExK2xxTkVteW9Da21acUhOM2tGaEp3OUFCZDN6QUlm\
Z29ldzhkRVBIRm14SnVOVklpM3FqbUVxU0U4QmhQUUdmRjVMSUhKRkpEQk1rUjgrOHZF\
eG9YMXpsRFZuNzFHMHZkRjBxUGJnV3BKbFZhQ3M1VHlSRlgxR0NZbmJvY2lBR3dKUito\
ekE5SnNKeGJ4TDVBUlNWR3J3Zlg2UEVJM0gvNUpreU1jSEdnNFdoZmJHbzQ4TEdFVlFx\
YVpXUkJHNG96aU00TU9HWGFPTHBZNFJSKzB0TVRnT0lpK0sxUWl1TFBkbjhUK0dJK1c5\
cDFTNTVBU0VHeGRBRFpKZmFSRkZaQll6b1VaeWJtRmt1WmJuUkVGOXh2QzlBbnc3L3Ay\
cG9HUXRwUGpkVlFZc1MxOUVjVWlCQmRhdU1HLzEya1FDTEhZMjlKL0orU2tFR0hKVTFr\
OWNRSlNvUTJRYmtpVm5RaVRWQzMwSzdtbkpKOW14U2NOSURBQUlOTFhUNHVDdU9pR20v\
c1RWMHI5TXJJcjRjTzRNYXBvTTBWbmJZTDA2d25SL2pzTTVPRFhxMVJ5blRMbGFXN29z\
cDA3TUFmQUhDd2V4M1p0bkFNZUFaUVJ4T0Yza1F3QU1reElITE01aThzMmFHTjJtV3FX\
VHR5cGlqMEw4eExtUzBhNXBUR2FWb1lhbnZrV1M1UC9CRjdER3hEUmtxdm1SbERyVXdo\
UjY",
  "d": "TUlJSWxnSUJBREFIQmdVcnpnOERCZ1NDQ0lZRWdnaUNXZmZoUXZBQS8veGd0\
UWd1ZWZBUU8vUHZ3Ky9mZ2h1L2ZRZXdTUHVBUyt5QWdCdVBnUWZ2Znd2UlBQdi9nZy8v\
Zy9DUFJ2L1BnQmZ1dkFBeEFoZGdQQXdmZitRUXdCQVFmL1BBUS9RZml4UGYreC93ZlFQ\
eGhoZ3Z4UHdBUHZmT1JmQTl1L1FTQkFBUXdneUFnUFFmdXdlUXVQd3YrdnhnUFF3Z1BP\
aEFBQkFnQkJBUXZ2UE9SaFEveGUrL0J2d0FnUE9RZkFCd1NmUHZnZnZnaEFoZnZBd2hm\
QUJlUHMvZS8vUWgvZ2ZSZmRRUVAvaGhodndRZk9BZy9TUHdlL1BnUFJQd1FndlF2eFBo\
ZndnQ2dQdnhRdndlLy9mQi9QdmdpZXh3d3dnUU9pZXc5UVB3dndnL2hQUHZOL2d1dy9l\
QWdTQXUvL3doZWdCUGUveEErZ0FldlFSZ3Z2Z1FnUndBTy93UFJPd0JoQkJQd3ZnaFBl\
Z2V2ZS8vdFFoUWR2ZmZ4UCsrZ3UvK3V3T2dBT2hnQWdBQjlmZk9nK0IreC8vUS92UlB3\
dU93dndmdit2UGVRQ0JmUXdnUHdQZnhnUHd3QWZ3Z1Ard2Z3dWZmUmZBUGYvL0FQUXZ3\
Umc5Z2dQUXdBdy8vZ0FoUHdneEFBUXZQQXhndnYvU3hnZy92QVJ3UFAraE93UGd2Qi9Q\
UWZCK1JleWlndmdBeFArL1FmUndlZWd3dndndit3QnlBd09BaEJQeFJBUGV2dWRQdmdo\
ZWdlaFAvT2dBdmdRZmZmZS92ZmZSd1FnUEF3UXZ3dnZ2Z2dndkJCUlBQeFJPL2YveUJB\
QWUvdHdQZ2V1ZkFmZmhQZmd4ZHZoUEJRL3ZBKy9oZi9oZWdRQnZTQVJkdk9RQVNpZndP\
K3dSQkFCZ2Z3Z2dRdnZRd2Q5QXdQL0Nmd3Z4ZmhRaFF3T3dnUndlL1FndGZkZlBPUVBT\
U1J1QWd3UGZBQVF3eFErUVJBd08rQXZRdmYvdy9QTy9RLy9oUWhQd2doQkJCQXQvdnQv\
L0FBd0FoZ0FRUCt2US8vdlFnK2cvd3dPK1JQLy8rZndQZ2cvQUFPZ0FBQWhld2gvQVJ2\
L2dCQUF3ZXZCQUFRL2Z3T3d2dmZQK2hST0FQZ2dmd2h3Z2dTZyt0QXdneEJnZlJSd3Zn\
UFF4UlFCQmh2US93ZlBnaHdlL3dmZi94Z2d4UVF2UC93dmdCaEJBd1EvQi9mQXdBdmdS\
ZmhQd2ZQd3d2aEFCZ3dmdlJCUkJCUVJSZ3dBdnd2Z1JPd0JQdnhBZnYrQS91UHdRUHdQ\
```

```
UnVBQStnQWVnZ2cvd3hmdnd4dndoaFFRaHZ2ZVF3d3cvQS94L09oZnYvZmdEcUV4VHRL\
QTc4OHdRTUh2ejNLZmJ5MXlNK0VTUU9LUHdJQ2Z6MkN3bnlIQ2dQRWU0TStnci83aEU0\
SnZtL0VBTUYrK1hUeWZILy9DTUJKUS8rR2hEYzVQRWRKZVFtQ2ZjaENQM2IwZ3I3RVF2\
eC9OWWQ5ZUxnendiZy9RZnEyeUlBNmpEd04rYjkwZWozL1JiL0IrWUUrdS9XTU9IQzNB\
VGcrL0w0NTk4VTdmOFA1Tm5qK2U3VkhpM3M2UWNVNE8wYitzN1UrQW9KRUFiajlRTCs2\
Tm5iOGUvbEY5Mngzd0l1K056WS92SFpFeGNNSkRIOUIvbjUrT1B0Nmd3Q0N0MEhHZlBw\
OGVudC91TWZDQW56NmhyckRnRFkyZXNpQ0FYNFNQME5JQ24zMlBzZzRRd2JJai82TlFJ\
TEFPd1NBQXphSjlMRkN4RWdFaG9MNHRQNkVpc2pIQUVZQVA3QThTN3dHZnNRQlFrcUVR\
TUdKd0pBKyt2NXBlNGtMd3dXRmdUM0FncmU4L29JRGczOUZRSUlBZlVLL3huY0Z6dnhE\
UXI4eWdIOTcrUUZHdjBIUFBBQTlFUVQ4T2NCNEFVWkRQYitDdVFETXdrRTZCL0k0Zi9X\
NE5VVkpBQVZMdzdXemhZaC92eitIT1A4OXh6TSt2enpJN2ZaS0R4SENBY04vTlFBOE5I\
Yk5SUGY4UG4xN1FvTTlBVU16UkFXTGYyc0JPcjVEUjhjMmcwbDYvQU1HK1hvNnVidThp\
aTg2aW4xSmZFWkMvQVR5T3ZNNVBEeDN2bjE3dlFIQU5EMUsvUGUrL2Y2L3ZjSXRSTGw1\
d2tORGhBWjZBOEdFK25xK3luK3lmZnBDUHNsQXZmYjZ2TDYrd250Nmc4VTIvamwvdkxM\
SFBMakU4b0E5QWM0Q1g3ZElvVGR5ajZmN3VmME5vbWJPekVtNG84U3JCMytYdWFveTA1\
cFFjVkpsWUF0VFkzcWsxcUNZcnJOdU9saXI5OC84dHh5SDZlRXNCNUxVVXBPbjZQWUtZ\
clZFVjFQM1ZHYmd5UmFKaDNkUldrVUpDWTVTVkEra2JBZUFiM29MVUtMZktTUk9UZGN3\
MStpL1BkamFNWXpwb2hWOEJqa003REhFTGNqTUhxTlJnSkRDdE95RXAvNTBXanl2Vm9X\
RXZWVTlZemxjamhMeWw5S29XTklZb1UrWUJGMnRqS2tsVXlsWlBma0F6c3QwSk5FSW0x\
WkIxS2xWR3ZpYks2WThuaXM0T2F2OHFVa1l5UGdsTzYrcGtJVkdDV09xdHIxclpwZ0JZ\
Wm5CbDlJVytrekpuVW5ZbFQ2bDBpVEhtT0MxbFNJZnh6SXBQekpFU1pHQldxQmhmb1BK\
a0tFVktvV0VFVFFseUNBVXlybTM3Qm9oMGZ3cEdweS9WNDBnNTFZU2xveTkrbEJiaWhn\
SUpReXdvdUg4YnVrZGtKRGtOYlZNMklkeERvc0hOVEx4M2lYY0RocjZ4eU9jT2w0TmRW\
YU9CUFdzek9yWGtOSWdUenp5UWNRWW41cFF5MFR1WHd0QW1PN2grRUYxQ1NEdlFhSUZk\
RDl5ejdWTzFyVmdvY1ZiMHc3R0xyMEZoWHVSb0JsNnppcnJRenhJUmFiVjlZamFYekpi\
bC9vdlNKMCtnOEttaUlvUE9LajQwVTBOc3phaVcyNldYRmdzNW1rbUxQQTcxR29uQldI\
QkNua0ZkbzdjM0FJbXlwUjhJQmdObXZvR0RkZXNSUGJWcEdNdUJUOUJyMkxBYjFNbnlT\
RmExK2xxTkVteW9Da21acUhOM2tGaEp3OUFCZDN6QUlmZ29ldzhkRVBIRm14SnVOVklp\
M3FqbUVxU0U4QmhQUUdmRjVMSUhKRkpEQk1rUjgrOHZFeG9YMXpsRFZuNzFHMHZkRjBx\
UGJnV3BKbFZhQ3M1VHlSRlgxR0NZbmJvY2lBR3dKUitoekE5SnNKeGJ4TDVBUlNWR3J3\
Zlg2UEVJM0gvNUpreU1jSEdnNFdoZmJHbzQ4TEdFVlFxYVpXUkJHNG96aU00TU9HWGFP\
THBZNFJSKzB0TVRnT0lpK0sxUWl1TFBkbjhUK0dJK1c5cDFTNTVBU0VHeGRBRFpKZmFS\
RkZaQll6b1VaeWJtRmt1WmJuUkVGOXh2QzlBbnc3L3AycG9HUXRwUGpkVlFZc1MxOUVj\
VWlCQmRhdU1HLzEya1FDTEhZMjlKL0orU2tFR0hKVTFrOWNRSlNvUTJRYmtpVm5RaVRW\
QzMwSzdtbkpKOW14U2NOSURBQUlOTFhUNHVDdU9pR20vc1RWMHI5TXJJcjRjTzRNYXBv\
TTBWbmJZTDA2d25SL2pzTTVPRFhxMVJ5blRMbGFXN29zcDA3TUFmQUhDd2V4M1p0bkFN\
ZUFaUVJ4T0Yza1F3QU1reElITE01aThzMmFHTjJtV3FXVHR5cGlqMEw4eExtUzBhNXBU\
R2FWb1lhbnZrV1M1UC9CRjdER3hEUmtxdm1SbERyVXdoUjY",
  "use": "sig"
}
```

6.4.1.3. jws

```
========== NOTE: '\' line wrapping per RFC 8792 ==========
eyJhbGciOiAiRkFMQ09ONTEyIiwgImt0eSI6ICJOVFJVIiwgInR5cCI6ICJKV1QifQ.e\
yJtZXNzYWdlIjogImhlbGxvIHdvcmxkIn0.OfsYG7sdSy2rsww2Np5ZwWxpr6hZrNHLu\
svsFtb8KcK2mrC5BRYQw1Z_pao6qWJj46wkiBlqgcFtqCNFj0L0DmMcuqx6DMjOcp9GE\
fqadmfIgivqlkOY9WSfS71K7GIw4T8Z1av5U_dlKjYWKwZqzK75nznsspTnDqRbTETK3\
OMIjicJ2-3VwjU0HGQXYyiJZo_OPOc__yyyJU7BfdlLMneUj9KZqQDdkloLBmeprdY5w\
ihsFcvlAUJ52IYzf-nmaAs1wzOoanjUJVyNFN4KAX58OpF47MrNnsLOmAQuAlrn6Uo0J\
h7YoqySUhtOKv8oY2rHU2Rb4JYqcYQdVA0qTibErkitJvqYAzdTeJBOVgtmVHDCRIbeF\
4SfL-TpZfsJadNDYUx2G9uGyqadwlyNS7tVjZIhGo807mfHfUi4btjCYZTd1zA0SFZFl\
tFCZivD7pxuCw-Ykf85ayy9sFu_W_UgTRC038I82YfbwdnbPH99Rx5ZGP1ySNGeuUuf3\
YxLT0PQDFLhW_jTZhDZCwh1Yj3UcQiDDLsYpdDYq1ipeB1KLm6mxaBqRRqjV2xcgpTyV\
aQNXin_aPHwJqoOlUy45E2sY482TS97mN64zee0uPM7FFIpVK3Pfx18i9lpuRS9RKEdV\
u0MKT4ZJKMHwEqZCEQFJfhrPvEdjblK93tj1xYs8auvLE2VUZo-nbEckuS7sdeDN9BS7\
Bf4-0EbQx-ML78mw5Clix2o49udTF-PU6xOkM83ZEFI5q4oV3LeF29Xs_Z1kl5uWKf2F\
nwKtz0hVTPKfZ9iPso6ukO3-OQ87EGFyHaf5zdXgXLwxFEwQh-WE3XmfboyVfL4yUkXv\
RUmNfpniJfXbMA8SD0mZp8ysA
```

7. Normative References

[Falcon] Fouque, P., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., and Z. Zhang, "Fast-Fourier Lattice-based Compact Signatures over NTRU", 2017, <https://falcon-sign.info/>.

[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>.

[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>.

8. 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