Workgroup: Internet Engineering Task Force
Individual-Draft: draft-uni-qsckeys-kyber-00
Published: 23 October 2022
Intended Status: Informational
Expires: 26 April 2023
Authors: C. V. Vredendaal, NXP Semiconductors
S. Dragone, IBM Research GmbH
B. Hess, IBM Research GmbH
T. Visegrady, IBM Research GmbH
M. Osborne, IBM Research GmbH
D. Bong, Utimaco IS GmbH
J. Bos, NXP Semiconductors

Quantum Safe Cryptography Key Information for CRYSTALS-Kyber

Abstract

This proposal defines key management approaches for the Quantum Safe Cryptographic (QSC) algorithm CRYSTALS-Kyber which has been selected for standardization by the NIST Post Quantum Cryptography (PQC) process. This includes key identification, key serialization, and key compression. The purpose is to provide guidance such that the adoption of quantum safe algorithms is not hampered with the fragmented evolution of necessary key management standards. Early definition of key material standards will help expedite the adoption of new quantum safe algorithms and at the same time as improving interoperability between implementations and minimizing divergence across standards.

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 26 April 2023.

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

1. Introduction
  1.1. Requirements Language
  1.2. Algorithm Identification
  1.3. Algorithm and Algorithm Parameter Object Identifier
2. Overview of CRYSTALS-Kyber and parameter OIDs
  2.1. Key Formats
  2.2. Public Key Format based on RFC5280
  2.3. Overview of Memo Definitions - PQC Key Formats
3. CRYSTALS-Kyber
  3.1. Algorithm Parameter Identifiers
  3.2. Key Details
  3.3. Private Key Full Encoding
  3.4. Private Key Partial Encoding
  3.5. Public Key Full Encoding
4. Acknowledgements
5. IANA Considerations
6. Security Considerations
7. References
  7.1. Normative References
  7.2. Informative References
Appendix A. Additional Stuff
Authors' Addresses

1. Introduction

QSC algorithms being standardized in the NIST PQC Process have evolved through several rounds and iterations. Keys are neither easily identifiable nor compatible across rounds. It is also expected that algorithms will evolve after final candidates have been selected. The lack of binary compatibility between algorithm versions and variants means that it is important to clearly identify key material. Parallel to the NIST process, industry is evaluating the impact of adopting new PQC algorithms, in particular key management. Here it is important to define and standardize key serialization and encoding formats. Finally, we have seen that many platforms and protocols are very constrained when it comes to the amount of memory or space available for key objects. This makes it important to define and standardize key compression formats. This proposal addresses aspects of key identification, key serialization, and key compression for the future primary NIST PQC KEM standard, CRYSTALS-Kyber. For the other schemes, see draft-uni-qsckeys-dilithium, draft-uni-qsckeys-falcon, draft-uni-qsckeys-sphincsplus and the previous Internet-Draft [draft-uni-qsckeys-01]. This proposal will be updated when the final NIST standard for CRYSTALS-Kyber becomes available.

1.1. Requirements Language

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 RFC 2119 [RFC2119] .

1.2. Algorithm Identification

Algorithm identification is important for several reasons:

* Managing a smooth transition from early adoption algorithm versions to production versions where there is no compatibility.
* Supporting different algorithm versions from different NIST rounds
* Identifying different key serialization strategies
* Identifying compressed and uncompressed keys

The current standardization of quantum safe algorithms does not address the definition of serialization structures for keys. As a result, it has become commonplace for the cryptographic community working on and with these algorithms to define their own approaches. This leads to proprietary and internal representations for key material. This has certain advantages in terms of ease of experimentation while focusing on finding the best-performing QSC algorithms. In terms of longer-term support where algorithm versions change this is a problem. This proposal defines in section 2 a long-term structured key representation format useful to address the goals outlined above.

1.3. Algorithm and Algorithm Parameter Object Identifier

Algorithm and algorithm parameter information shall have ASN.1 type AlgorithmIdentifier as given in [RFC5280] and shall be extended by an pqcAlgorithmParameterName type in the optional parameters field:

```
AlgorithmIdentifier ::=  SEQUENCE {
    algorithm  OBJECT IDENTIFIER, - OID: algorithm and algo parameter
    parameters pqcAlgorithmParameterName OPTIONAL
}
pqcAlgorithmParameterName ::= PrintableString
```

2. Overview of CRYSTALS-Kyber and parameter OIDs

CRYSTALS-Kyber consists of six parameter sets. This memo attributes a name and a placeholder for an OID to the different parameter sets of CRYSTALS-Kyber. The following table gives an overview of the possible OIDs in the algorithm field and possible parameters set names in the parameters field of the AlgorithmIdentifier type. Each name or OID represents a single parameter set of given security. Details can be found in the next section.

```
|=========+=====+==============================================|
| CRYSTALS-Kyber (PQC KEM)                                     |
|=========+=====+==============================================|
| kyber-512-r3                                                 |
|---------+-----+----------------------------------------------|
|         |ASN.1| {..*.. pqc-kem-kyber kyber-512-r3 }          |
|         |dot. |                                              |
|---------+-----+----------------------------------------------|
| kyber-512-90s-r3                                             |
|---------+-----+----------------------------------------------|
|         |ASN.1| {..*.. pqc-kem-kyber kyber-512-90s-r3}       |
|         |dot  |                                              |
|---------------+-----+----------------------------------------|
| kyber-768-r3                                                 |
|---------+-----+----------------------------------------------|
|         |ASN.1| {..*.. pqc-kem-kyber kyber-768-r3 }          |
|         |dot  |                                              |
|---------------+-----+----------------------------------------|
| kyber-768-90s-r3                                             |
|---------------+-----+----------------------------------------|
|         |ASN.1| {..*.. pqc-kem-kyber kyber-768-90s-r3 }      |
|         |dot  |                                              |
|---------+-----+----------------------------------------------|
| kyber-1024-r3                                                |
|---------+-----+----------------------------------------------|
|         |ASN.1| {..*.. pqc-kem-kyber kyber-1024-r3 }         |
|         |dot  |                                              |
|---------+-----+----------------------------------------------|
| kyber-1024-90s-r3                                            |
|---------+-----+----------------------------------------------|
|         |ASN.1| {..*.. pqc-kem-kyber kyber-1024-90s-r3}      |
|         |dot  |                                              |
|=========+=====+==============================================|
```

Figure 1

2.1. Key Formats

The private key format defined is from PKCS#8 [RFC5208] . PKCS#8 PrivateKeyInfo is defined as:

```
PrivateKeyInfo ::=  SEQUENCE {
    version               INTEGER             -- PKCS#8 syntax version
    privateKeyAlgorithm   AlgorithmIdentifier -- see chapter above
    privateKey            OCTET STRING,       -- see chapter below
    attributes            [0]  IMPLICIT Attributes OPTIONAL
}
```

Distributing a PQC private key requires a PKCS#8 PrivateKeyInfo with a joined PQC algorithm and algorithm parameter OID in the algorithm field of AlgorithmIdentifier and a PQC algorithm specific private key object in the privateKey field of PrivateKeyInfo. Both objects are defined in the specific algorithm sections of this document. For an overview see tables above and below.

2.2. Public Key Format based on *[RFC5280]*

RFC5280 subjectPublicKeyInfo is defined in as:

```
SubjectPublicKeyInfo := SEQUENCE {
    algorithm          AlgorithmIdentifier  -- see chapter above
    subjectPublicKey   BIT STRING           -- see chapter below
}
```

Distributing a PQC public key requires a [RFC5480] subjectPublicKeyInfo with a joined PQC algorithm and algorithm parameter OID in the algorithm field of AlgorithmIdentifier and a PQC algorithm specific public key object in the subjectPublicKey field of subjectPublicKeyInfo. Both objects are defined in the specific algorithm sections of this document. For an overview see tables above and below.

2.3. Overview of Memo Definitions - PQC Key Formats

The privateKey field in the PrivateKeyInfo type [RFC5480] is an OCTET STRING whose contents are the value of the private key. The interpretation of the content differs from PQC algorithm to algorithm. The subjectPublicKey field in the subjectPublicKeyInfo type [RFC5480] is a BIT STRING whose contents are the value of the public key. Here also the interpretation of the content differs from PQC algorithm to algorithm.

3. CRYSTALS-Kyber

CRYSTALS-Kyber is an IND-CCA2-secure key encapsulation mechanism (KEM), whose security is based on the hardness of solving the learning-with-errors (LWE) problem over module lattices.

- *Project Website: https://pq-crystals.org/kyber/index.shtml
- *NIST Round 3 Submission: https://csrc.nist.gov/CSRC/media/Projects/post-quantum-cryptography/documents/round-3/submissions/Kyber-Round3.zip

3.1. Algorithm Parameter Identifiers

CRYSTALS-Kyber uses OIDs to identify parameters sets.

```
|=========================+======================================|
| kyber-512-r3                                                   |
|=========================+======================================|
| Parameter OID           | {..*.. kyber-512-r3}                 |
|                         | <.>                                  |
| NIST Level Security     | Level 1                              |
|-------------------------|--------------------------------------|
| Parameters              | n= 256,                              |
|                         | k=2                                  |
|                         | q=3329                               |
|                         | eta_1=3                              |
|                         | eta_2=2                              |
|                         | (d_u, d_v)=(10, 4)                   |
|                         | delta=2^{-139}                       |
|=========================+======================================|
| kyber-512-90s-r3                                               |
|=========================+======================================|
| Parameter OID           | {..*.. kyber-512-90s-r3}             |
|                         | <.>                                  |
| NIST Level Security     | Level 1                              |
|-------------------------|--------------------------------------|
| Parameters              | n= 256,                              |
|                         | k=2                                  |
|                         | q=3329                               |
|                         | eta_1=3                              |
|                         | eta_2=2                              |
|                         | (d_u, d_v)=(10, 4)                   |
|                         | delta=2^{-139}                       |
|=========================+======================================|
| kyber-768-r3                                                   |
|=========================+======================================|
| Parameter OID           | {..*.. kyber-768-r3}                 |
|                         | <.>                                  |
| NIST Level Security     | Level 3                              |
|-------------------------|--------------------------------------|
| Parameters              | n= 256,                              |
|                         | k=3                                  |
|                         | q=3329                               |
|                         | eta_1=2                              |
|                         | eta_2=2                              |
|                         | (d_u, d_v)=(10, 4)                   |
|                         | delta=2^{-164}                       |
|=========================+======================================|
| kyber-768-90s-r3                                               |
|=========================+======================================|
| Parameter OID           | {..*.. kyber-768-90s-r3}             |
|                         | <.>                                  |
| NIST Level Security     | Level 5                              |
```

Parameters	n= 256, k=3 q=3329 eta_1=2 eta_2=2 (d_u, d_v)=(10, 4) delta=2^{-164}
kyber-1024-r3	
Parameter OID	{..*.. kyber-1024-r3} <.>
NIST Level Security	Level 5
Parameters	n= 256, k=4 q=3329 eta_1=2 eta_2=2 (d_u, d_v)=(11, 5) delta=2^{-174}
kyber-1024-90s-r3	
Parameter OID	{..*.. kyber-1024-90s-r3} <.>
NIST Level Security	Level 5
Parameters	n= 256, k=4 q=3329 eta_1=2 eta_2=2 (d_u, d_v)=(11, 5) delta=2^{-174}

Figure 2

The '90s' variants listed above differ in the symmetric primitives that are used internally. By default, CRYSTALS-Kyber uses SHAKE-128 as XOF, SHA3 for hashing and SHAKE-256 for PRF and KDF. The '90s' variants use AES256CTR to construct a XOF and a PRF, SHA2 for hashing and SHAKE-256 as KDF. The main advantage of the '90s' variants is that they benefit from the ready availability of hardware AES and SHA2 co-processors. While the parameters listed in the table are the same, the key-pairs will not be compatible with the non-'90s' variants.

3.2. Key Details

Public key. The public-key consists of two parameters:

- *t: encoded vector A*s+e, where A is a public matrix over a constant-sized polynomial ring, s and e are vectors over the same ring.
- *rho: public seed (32 bytes)

The size necessary to hold all public key elements is 12*k*n/8+32 bytes.

Private key. The private key consists of 3 parameters:

- *s: encoded sample from a centered binomial distribution B_{eta_1} (12*k*n/8 bytes)
- *H(pk): hashed public key (32 bytes). CRYSTALS-Kyber uses SHA3-256 as H by default. The '90s' variants use SHA256 instead.
- *z: a nonce (32 bytes)

If the private key is fully populated, it consists of 3 parameters. The size necessary to hold all private key elements accounts to 12*k*n/8+64 bytes, not counting the optional public key. The resulting public key and private key sizes are shown in the following table.

Algorithm OID	Public Key	Private Key	Private Key (partial)
kyber512-r3 / kyber512-90s-r3	800	832	32
kyber768-r3 / kyber768-90s-r3	1184	1216	32
kyber1024-r3 / kyber1024-90s-r3	1568	1600	32

Figure 3

3.3. Private Key Full Encoding

Encoding a CRYSTALS-Kyber private key with PKCS#8 must include the following two fields:

* kyber-(n*k)-r3 in the algorithm field of AlgorithmIdentifier
* KyberPrivateKey in the privateKey field, which is an OCTET STRING.

When a CRYSTALS-Kyber public key is included in the distributed PrivateKeyInfo, the PublicKey field in KyberPrivateKey is used (see description of KyberPublicKey below). The ASN.1 encoding for a CRYSTALS-Kyber private key is defined as follows:

```
KyberPrivateKey ::= SEQUENCE {
    version     INTEGER {v0(0)}   -- version (round 3)
    s           OCTET STRING,     -- sample s
    publicKey   [0] IMPLICIT KyberPublicKey OPTIONAL,
                                  -- see next section
    hpk         OCTET STRING      -- H(pk)
    nonce       OCTET STRING,     -- z
}
```

3.4. Private Key Partial Encoding

The partially populated parameter set uses of the fact that some parameters can be regenerated. In this case, only the initial seed 'd' (nonce) is stored and used to regenerate the full key. Partially encoded keys use the same ASN.1 structure as the fully populated keys, simply with the regenerated fields set to EMPTY. Compared to the approach of a single definition and setting the regenerable fields as OPTIONAL, this approach significantly simplifies the processing of ASN.1 frames and validation of the partial encoding. The ASN.1 format for the partially populated versions is the same as for the fully populated version. The ASN.1 encoding for this variant (z replaced by d) is defined as follows:

```
KyberPrivateKey ::= SEQUENCE {
    version     INTEGER {v0(0)}   -- version (round 3)
    s           OCTET STRING,     -- EMPTY
    publicKey   [0] IMPLICIT KyberPublicKey OPTIONAL,
                                  -- see next section
    hpk         OCTET STRING      -- EMPTY
    nonce       OCTET STRING,     -- d
}
```

3.5. Public Key Full Encoding

The vector 't' is encoded using the function Encode_12, defined as the inverse of Decode_12 as defined in Algorithm 3 of the CRYSTALS-Kyber round 3 specification. The size of t is 12*k*n/8 bytes. The seed 'rho' is a 32 byte OCTET STRING.

```
KyberPublicKey ::= SEQUENCE {
    t           OCTET STRING,
    rho         OCTET STRING
}
```

4. Acknowledgements

This template was derived from an initial version written by Pekka Savola and contributed by him to the xml2rfc project.

This document is part of a plan to make xml2rfc indispensable.

5. IANA Considerations

This memo includes no request to IANA.

6. Security Considerations

Any processing of the ASN.1 private key structures, such as base64 en/decoding shall be performed in "constant-time", meaning without secret-dependent control flow and table lookups. The ASN.1 structures in this document are defined with fixed tag-lengths. The purpose is to prevent side-channel leakage of variable lengths during DER parsing. Any DER parsing of the private key ASN.1 key structures shall be performed with these fixed lengths.

7. References

7.1. Normative References

[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>.

[RFC5208] Kaliski, B., "Public-Key Cryptography Standards (PKCS) #8: Private-Key Information Syntax Specification Version 1.2", BCP 14, RFC 5208, DOI 10.17487/RFC5208, May 2008, <hhttps://www.rfc-editor.org/info/rfc5208>.

[RFC5280] Cooper, D., "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", BCP 14, RFC RFC5280, DOI 10.17487/RFC5280, May 2008, <hhttps://www.rfc-editor.org/info/rfcRFC5280>.

[RFC5480] Turner, S., "Elliptic Curve Cryptography Subject Public Key Information", BCP 14, RFC RFC5480, DOI 10.17487/RFC5480, May 2009, <hhttps://www.rfc-editor.org/info/rfc5480>.

7.2. Informative References

[draft-uni-qsckeys-01]
Vredendaal, C. V., Dragone, S., Hess, B., Visegrady, T., Osborne, M., Bong, D., and J. Bos, "Quantum Safe Cryptography Key Information", Work in Progress, Internet-Draft, draft-uni-qsckeys-01, 12 May 2022, <https://www.ietf.org/archive/id/draft-uni-qsckeys-01.txt>.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, DOI 10.17487/RFC2629, June 1999, <https://www.rfc-editor.org/info/rfc2629>.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <https://www.rfc-editor.org/info/rfc3552>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, DOI

10.17487/RFC5226, May 2008, <https://www.rfc-editor.org/info/rfc5226>.

Appendix A. Additional Stuff

This becomes an Appendix.

Authors' Addresses

Christine van Vredendaal
NXP Semiconductors
High Tech Campus 60
5656 AE Eindhoven
Netherlands

Email: cvvrede@gmail.com

Silvio Dragone
IBM Research GmbH
Saeumerstrasse 4
CH-8803 Rueschlikon
Switzerland

Email: sid@zurich.ibm.com

Basil Hess
IBM Research GmbH
Saeumerstrasse 4
CH-8803 Rueschlikon
Switzerland

Email: bhe@zurich.ibm.com

Tamas Visegrady
IBM Research GmbH
Saeumerstrasse 4
CH-8803 Rueschlikon
Switzerland

Email: tvi@zurich.ibm.com

Michael Osborne
IBM Research GmbH
Saeumerstrasse 4
CH-8803 Rueschlikon
Switzerland

Email: osb@zurich.ibm.com

Dieter Bong

Utimaco IS GmbH
Germanusstrasse 4
52080 Aachen
Germany

Email: dieter.bong@utimaco.com

Joppe Bos
NXP Semiconductors
High Tech Campus 60
5656 AE Eindhoven
Netherlands

Email: joppe.bos@nxp.com