draft-ietf-lamps-pq-composite-kems
&
draft-ounsworth-lamps-cms-dhkem

Mike Ounsworth, John Gray, Max Pala,
Jan Klaußner, Scott Fluhrer

LAMPS 118
- draft-ounsworth-pq-composite-keys
  Draft no longer needed;
  will be allowed to expire.
  All Pub Key content folded in
  to composite-kem draft.

draft-ietf-lamps-kyber-certificates

+ draft-ietf-lamps-rfc5990bis
  Provides RSA-KEM

+ draft-housley-lamps-cms-sha3-hash
  Provides id-sha3-256 needed
  internally for RSA-KEM instantiation.
  UNDER DEBATE – see slide below.

+ draft-ounsworth-lamps-cms-dhkem
  Provides KEM wrappers for ECDH
  and EdDH.
  (ie the ECC equiv. of RFC 5990)
Changes affecting Interoperability

- Defined KeyGen(), Encaps(), and Decaps() for a composite KEM algorithm.
- Re-worked wire format and ASN.1 to remove vestiges of Generics.
  - Changed all SEQUENCE OF SIZE (2..MAX) to SEQUENCE OF SIZE (2).
  - Changed the definition of CompositeKEMPublicKey from SEQUENCE OF SubjectPublicKeyInfo to SEQUENCE OF BIT STRING since with complete removal of Generic Composites, there is no longer any need to carry the component AlgorithmIdentifiers.
  - Removed CompositeKEMParams since all params are now explicit in the OID.
- Removed the discussion of KeyTrans -> KEM and KeyAgree -> KEM promotions, and instead simply referenced [I-D.ietf-lamps-rfc5990bis] and [I-D.ounsworth-lamps-cms-dhkem].
- Made RSA keys fixed-length at 2048 and 3072
  - (Added id-MLKEM512-RSA2048-KMAC128 and id-MLKEM768-RSA3072-KMAC256)
Changes affecting Interoperability

• Re-worked section 5.1 & 5.2 (id-MLKEM512-RSA2048-KMAC128 and id-MLKEM768-RSA3072-KMAC256 instantiations) to Reference 5990bis and its updated structures.
  - Removed RSA-KEM KDF params and make them implied by the OID; ie provide a profile of 5990bis.

• Still TODO:
  - Align combiner with draft-ounsworth-cfrg-kem-combiners-04.
Editorial changes

- Refactored to use MartinThomson github template.
- Made this document standalone by folding in the minimum necessary content from composite-keys and dropping the cross-reference to composite-sigs.
- Added a paragraph describing how to reconstitute component SPKIs.
- Added an Implementation Consideration about FIPS validation where only one component algorithm is FIPS-approved.
- Shortened the abstract (moved some content into Intro).
- Brushed up the Security Considerations.
- Made a proper IANA Considerations section.
- Rename "Kyber" to "ML-KEM".
Open design issues
Composite KEM construction

Combiner(ss1, ss2, fixedInfo) = KDF(counter || ss1 || ss2 || fixedInfo, outputBits)

I ran into “deadline issues”; Not yet aligned with draft-ounsworth-cfrg-kem-combiners-04

It should be “ct1 || ss1 || ct2 || ss2”

<table>
<thead>
<tr>
<th>KEM Combiner Name</th>
<th>KDF</th>
<th>outputBits</th>
</tr>
</thead>
<tbody>
<tr>
<td>KMAC128/256</td>
<td>KMAC128</td>
<td>256</td>
</tr>
<tr>
<td>KMAC256/384</td>
<td>KMAC256</td>
<td>384</td>
</tr>
<tr>
<td>KMAC256/512</td>
<td>KMAC256</td>
<td>512</td>
</tr>
</tbody>
</table>
Algs list
(... spoiler: it has grown to split “RSA” into 2048 & 3072)

1. id-MLKEM512-RSA2048-KMAC128
2. id-MLKEM512-ECDH-P256-KMAC128
3. id-MLKEM512-ECDH-brainpoolP256r1-KMAC128
4. id-MLKEM512-X25519-KMAC128
5. id-MLKEM768-RSA3072-KMAC256
6. id-MLKEM768-ECDH-P256-KMAC256
7. id-MLKEM768-ECDH-brainpoolP256r1-KMAC256
8. id-MLKEM768-X25519-KMAC256
9. id-MLKEM1024-ECDH-P384-KMAC256
10. id-MLKEM1024-ECDH-brainpoolP384r1-KMAC256
11. id-MLKEM1024-X448-KMAC256

Note: The full AlgID table in the draft specifies whether each is
- KMAC128/256
- KMAC256/384
- KMAC256/512
## RSA-KEM instantiations -- SHA3 or SHAKE?

<table>
<thead>
<tr>
<th>RSA-KEM Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>id-MLKEM512-RSA2048-KMAC128</td>
<td></td>
</tr>
<tr>
<td>keyDerivationFunction</td>
<td>kda-kdf3 with id-sha3-256</td>
</tr>
<tr>
<td>keyLength</td>
<td>128</td>
</tr>
<tr>
<td>DataEncapsulationMechanism</td>
<td>kwa-aes128-wrap</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RSA-KEM Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>id-MLKEM768-RSA3072-KMAC256</td>
<td></td>
</tr>
<tr>
<td>keyDerivationFunction</td>
<td>kda-kdf3 with id-sha3-384</td>
</tr>
<tr>
<td>keyLength</td>
<td>256</td>
</tr>
<tr>
<td>DataEncapsulationMechanism</td>
<td>kwa-aes256-wrap</td>
</tr>
</tbody>
</table>

### Reasoning:
Since Kyber already needs SHA3, implementations already need it, might as well re-use it here rather than introducing another primitive (SHAKE).

### Problem:
NIST has assigned SHA3 OIDs, but they are not currently in any IETF CMS RFC.

But RFC 8702 registers OIDs for SHAKE128, SHAKE256, which can be used as message digests. – Panos is advocating for this instead.

**Need WG consensus here!**
It’s an -00; needs some work, but I needed something quick to point composite-kems at.

“This document defines a mechanism to wrap Ephemeral-Static (E-S) Diffie-Hellman (DH) and Elliptic Curve Diffie-Hellman (ECDH) such that it can be used in KEM interfaces within the Cryptographic Message Syntax (CMS).”

“This is a sister document to RSA-KEM [RFC5990] and simplifies future cryptographic protocol design by only needing to handle KEMs at the protocol level.”

“This draft follows the DH-Based KEM (DHKEM) construction defined in [RFC9180] whereby the Encapsulate() operation includes the generation of an ephemeral key and the usage of that key against the recipient's static public key.”

Russ gave some initial feedback that I have not yet incorporated.

Fingers crossed that this is straightforward once I get it written out properly.