Comparison of hybrid KEM drafts across WGs
## Comparison

<table>
<thead>
<tr>
<th>Draft</th>
<th>Purpose</th>
<th>Combiner</th>
</tr>
</thead>
<tbody>
<tr>
<td>cfrg-kem-combiners</td>
<td>Abstract construction. Secure in “worst case”.</td>
<td>KDF(ct1, ss1, ct2, ss2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>* ct and ss must be length-encoded if not constant length.</td>
</tr>
<tr>
<td>tls-hybrid-design</td>
<td>Ephemeral-ephemeral key exchange embedded within the TLS handshake.</td>
<td>ss1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>transcript(including ct1, ct2) --&gt; HKDF-EXTRACT</td>
</tr>
<tr>
<td>cfrg-hpke-xyber768d00</td>
<td>Add X25519Kyber768 to HPKE (RFC 9180).</td>
<td>ss1</td>
</tr>
<tr>
<td>lamps-composite-kem</td>
<td>Public keys for KEM X.509 certificates. For use anywhere that needs X.509 encryption certs.</td>
<td>cfrg-kem-combiners with KMAC{128,256}</td>
</tr>
<tr>
<td></td>
<td>CMS, S/MIME – typically one-shot KEM for CEK key wrapping.</td>
<td></td>
</tr>
<tr>
<td>openpgp-pqc</td>
<td>OpenPGP</td>
<td>cfrg-kem-combiners with KMAC256</td>
</tr>
<tr>
<td></td>
<td>Long-lived public keys. One-shot KEM for CEK key wrapping.</td>
<td></td>
</tr>
<tr>
<td>jose-hybrid-encrypt</td>
<td>JOSE / COSE</td>
<td>cfrg-kem-combiners with KMAC256</td>
</tr>
<tr>
<td></td>
<td>Long-lived public keys. One-shot KEM for CEK key wrapping.</td>
<td></td>
</tr>
</tbody>
</table>
### Hybrid Algorithm Alignment

* This is close to aligned, but not perfect. … missing P521??

<table>
<thead>
<tr>
<th>X</th>
<th>ML-KEM-512</th>
<th>ML-KEM-768</th>
<th>ML-KEM-1024</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA</td>
<td>lamps-composite-kem</td>
<td>lamps-composite-kem</td>
<td></td>
</tr>
<tr>
<td>ECDH-P256</td>
<td>lamps-composite-kem jose-hybrid-encrypt</td>
<td>tls-hybrid-design lamps-composite-kem openpgp-pqc</td>
<td></td>
</tr>
<tr>
<td>ECDH-Brainpool-P256</td>
<td>lamps-composite-kem</td>
<td>lamps-composite-kem openpgp-pqc</td>
<td></td>
</tr>
<tr>
<td>X25519</td>
<td>lamps-composite-kem jose-hybrid-encrypt</td>
<td>tls-hybrid-design crfg-hpke-xyber768d00 lamps-composite-kem openpgp-pqc jose-hybrid-encrypt</td>
<td></td>
</tr>
<tr>
<td>ECDH-P384</td>
<td></td>
<td>jose-hybrid-encrypt *</td>
<td>lamps-composite-kem openpgp-pqc</td>
</tr>
<tr>
<td>ECDH-Brainpool-P384</td>
<td></td>
<td></td>
<td>lamps-composite-kem openpgp-pqc</td>
</tr>
<tr>
<td>X448</td>
<td></td>
<td></td>
<td>lamps-composite-kem openpgp-pqc</td>
</tr>
</tbody>
</table>
Aside: SP 800-56Cr2 shared secret combiners

“this Recommendation permits the use of a “hybrid” shared secret of the form $Z' = Z \parallel T$”

- HASH(), HMAC-hash() or KMAC#()
- $\text{counter} \parallel Z \parallel \text{FixedInfo}$
  - counter starts at 0x00000000; if using HASH() on large input then you need to iterate and increment $\text{counter}$. 

- draft-ounsworth-cfrg-kem-combiners is the direct instantiation of this (ie trivial FIPS-compliance).
- tls-hybrid-design and cfrg-hpke-xyber should be compliant with some explaining.
**draft-ounsworth-cfrg-kem-combiners**

**Authors:** Mike Ounsworth, Aron Wussler, Stavros Kousidis

**Purpose:**
Defines an abstract KDF construction, compatible with NIST SP 800-56Cr2, suitable for combining two or more KEMs such that the overall combined KEM will be IND-CCA2 so long as at least one of the component KEMs is.

**Construction:**
\[
ss = \text{KDF}(\text{counter} \mid\mid k_1 \mid\mid ... \mid\mid k_n \mid\mid \text{fixedInfo}, \text{outputBits})
\]
Where:
\[
k_i = (\text{ct}_i \mid\mid ss_i) \quad -- \text{if constant-length}
\]
OR
\[
k_i = (\text{ct}_i \mid\mid \text{rlen}(\text{ct}_i) \mid\mid ss_i \mid\mid \text{rlen}(ss_i))
\]

**Instantiations:**
KDF = KMAC128, KMAC256, SHA3-256, or SHA3-512.

*Does not* provide concrete instantiations of component KEMs.
This document focuses on hybrid ephemeral key exchange in TLS 1.3

Construction:

ss1 || ss2 --> HKDF-EXTRACT
transcript(including ct1, ct2) --> HKDF-EXTRACT

Also compatible with NIST SP 800-56Cr2.
This is very close to being an instantiation of cfrg-kem-combiners; includes ss’s and ct’s in “cascading” mode, compared to cfrg-kem-combiner’s “concatenated” mode.
Forgoes length-encoding because X25519, SecP256r1 and Kyber are fixed-length.

Instantiations:

X25519Kyber768Draft00, SecP256r1Kyber768Draft00
This memo defines X25519Kyber768Draft00, a hybrid post-quantum KEM, for HPKE (RFC9180).

In short, X25519Kyber768Draft00 is the parallel combination of DHKEM(X25519, HKDF-SHA256) [RFC9180] [RFC7748] and Kyber768Draft00 [KYBER].

Construction:

ss1 || ss2

OK to not include ct’s because both DHKEM(X25519) and Kyber768Draft00 already mix in ct’s. OK to not include KDF at this layer because HPKE (RFC9180) will subsequently pass the ss through HKDF … so it still compliant with NIST SP 800-56Cr2.

Instantiations:

X25519Kyber768Draft00
This memo defines X25519Kyber768Draft00, a hybrid post-quantum KEM, for HPKE (RFC9180).

In short, X25519Kyber768Draft00 is the parallel combination of DHKEM(X25519, HKDF-SHA256) [RFC9180] [RFC7748] and Kyber768Draft00 [KYBER].

Construction:

ss1 || ss2

OK to not include ct’s because both DHKEM(X25519) and Kyber768Draft00 already mix in ct’s.

OK to not include KDF at this layer because HPKE (RFC9180) will subsequently pass the ss through HKDF … so it still compliant with NIST SP 800-56Cr2.

Note:
No longer true if we’re talking about ML-KEM-ipd!

So does that mean the protocol now needs to commit the ciphertext?
This document defines Post-Quantum / Traditional composite Key Encapsulation Mechanism (KEM) algorithms suitable for use within X.509 and PKIX and CMS protocols.

Construction:

cfrg-kem-combiners with KMAC128 or KMAC256, matching the security level of component KEMs.

Instantiations:

Given the wide range of applications of X.509 encryption certificates, a broader set of instantiations is provided (see next slide).
It’s the permutations of:

ML-KEM

X

{ RSA, ECDH-SecP, ECDH-brainpool, ECDH-Edwards }
draft-wussler-openpgp-pqc

Authors: Stavros Kousidis, Falko Strenzke, Aron Wussler

Purpose:
Provide OpenPGP with composite public-key encryption based on ML-KEM in combination with elliptic curve cryptography because OpenPGP has already deprecated RSA.

Construction:
cfrg-kem-combiners with KMAC256.

Instantiations:
ML-KEM-768_x25519Kem, ML-KEM-1024_x448Kem,
ML-KEM-768_ecdhKemNISTP-256, ML-KEM-1024_ecdhKemNISTP-384,
ML-KEM-768_ecdhKemBrainpoolP256r1, ML-KEM-1024_ecdhKemBrainpoolP384r1
Purpose:
“This document provides a construction for hybrid key exchange in JOSE and COSE.”
Note: this is a competing proposal to getting it “for free” from HPKE-xyber.

Construction:
cfrg-kem-combiners with KMAC256.

Instantiations:
x25519-ES_kyber512, secp384r1-ES_kyber768, x25519-ES_kyber768, secp256r1-ES_kyber512
Summary

Combiner construction

- cfrg-kem-combiners is intended to be “overkill”.
  - lamps-composite-kem, openpgp-pqc, jose-hybrid-encrypt directly chain to cfrg-kem-combiners.
  - Open question: can cfrg-kem-combiners safely be further relaxed?
- tls-hybrid-design and crfg-hpke-xyber768d00 relax the combiner construction from cfrg-kem-combiners because their respective protocols (TLS and HPKE) already include ct’s and an HKDF.
  - These constructions are all cryptographically equivalent.

Supported algorithm combinations

- Algs are more aligned than it may appear at first glance; we are very close to:
  lamps-composite-kem ⊃ openpgp-pqc ⊃ jose-hybrid-encrypt ⊃
  tls-hybrid-design ⊃ crfg-hpke-xyber768d00
  (where ⊃ is “superset”)


**Summary**

**Instantiating crfg-kem-combiners**

`crfg-hpke-xyber768d00` provides a really nice template for how to “instantiate` crfg-kem-combiners:

3. Construction

```python
...  
return concat(ss1, ss2)
```

4. Security Considerations

Identify and justify the difference from crfg-kem-combiners.

“In the present case, DHKEM(X25519, -) and Kyber768Draft00 already mix in the respective cipher texts into their shared secrets.

Furthermore, in HPKE, the shared secret is never used directly, but passed through HKDF (via KeySchedule), and thus we can forgo the call to HKDF as well.”