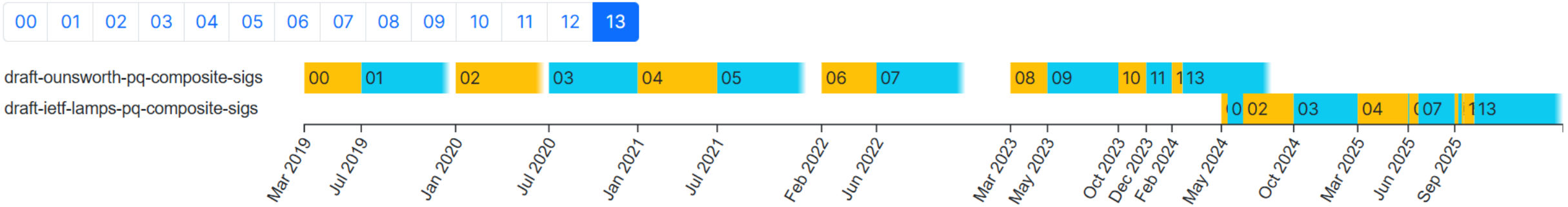
Composite Signature – 13 Updates
Composite KEM - 09 Updates

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

November 2025

Composite Signatures

- Achieved WGLC !!
- Early OID assignments registered!
  - https://www.iana.org/assignments/smi-numbers/smi-numbers.xhtml
- WG State: Submitted to IESG for Publication
- Thanks to everyone who has helped!



Composite KEM

- Currently in WGLC
  - A few minor editorial comments from Russ, Piotr and Dan.
  - SPACESHIPS

```
\./      |
/^\     /-\        |-()-|
```

    - Hackathon update: Cute, but … we switched to ASCII.
    - During the hackathon, the ref. Impl. for LAMPS Composite-KEM and CFRG Concrete Hybrid KEMs successfully performed interop testing on labels: “\.//^\ (0x5C2E2F2F5E5C)”, “MLKEM768-P256”, “MLKEM1024-P384”.
    - That’s it. We’re done. We all agree: no more changes.
  - Authors need to do one more version that includes the labels change, and address a few more github issues.
  - Then close WGLC?

Composite KEM

- Open discussion topic: private key format
- -07 had this:

  `mlkemSeed || tradSK`
- -08 now has this:

  `mlkemSeed || lenTradPK || tradPK || tradSK`
- Reason:
  - You need `tradPK` as input to the Decaps() KEM Combiner; in general you might not be able to derive this from the private key, therefore carry it.
  - But we would like feedback on this from LAMPS.

CMS Usage Documents

- The “Use in CMS” section of both the composite Signatures and composite KEM was removed to get documents through WGLC
- Authors group has started working on the CMS versions of the documents
  - Daniel Van Geest has joined us to help with this work

- **Question**: Since the “Use in CMS” was originally part of Composite ML-DSA and Composite ML-KEM when adopted, can we go ahead and post these as
  - draft-ietf-lamps-cms-composite-sigs
  - draft-ietf-lamps-cms-composite-kem

Bonus Stuff….

Working Implementations – Dynamic Hackathon Results

-- Composite Signatures (Final OIDS!)

✅ = passing all verifiers
◒ = passing some verifiers
○ = not passing any verifiers
Columns represent producers who submitted artifacts. Verifiers are not listed in this table, but are listed in the broken-out tables below.

-	bc	carl-redhound	cht	composite-kem-ref-impl	composite-sigs-ref-impl	corey-digicert	crypto4a	cryptonext	entrust	ossl35	safelogic	seventhsense.ai
id-MLDSA44-RSA2048-PSS-SHA256-cert			◒ 1/3		◒ 1/3			◒ 1/3				
id-MLDSA44-RSA2048-PKCS15-SHA256-cert			◒ 1/3		◒ 1/3			◒ 1/3				
id-MLDSA44-Ed25519-SHA512-cert			◒ 1/3		◒ 1/3			◒ 1/3				
id-MLDSA44-ECDSA-P256-SHA256-cert			◒ 1/3		◒ 1/3			◒ 1/3				
id-MLDSA65-RSA3072-PSS-SHA512-cert			◒ 1/3		◒ 1/3			◒ 1/3				
id-MLDSA65-RSA3072-PKCS15-SHA512-cert			◒ 1/3		◒ 1/3			◒ 1/3				
id-MLDSA65-RSA4096-PSS-SHA512-cert			◒ 1/3		◒ 1/3			◒ 1/3				
id-MLDSA65-RSA4096-PKCS15-SHA512-cert			◒ 1/3		◒ 1/3			◒ 1/3				
id-MLDSA65-ECDSA-P256-SHA512-cert			◒ 1/3		◒ 1/3			◒ 1/3				
id-MLDSA65-ECDSA-P384-SHA512-cert			◒ 1/3		◒ 1/3			◒ 1/3				
id-MLDSA65-ECDSA-brainpoolP256r1-SHA512-cert			◒ 1/3		◒ 1/3			◒ 1/3				
id-MLDSA65-Ed25519-SHA512-cert			◒ 1/3		◒ 1/3			◒ 1/3			◒ 2/4	
id-MLDSA87-ECDSA-P384-SHA512-cert			◒ 1/3		◒ 1/3			◒ 1/3				
id-MLDSA87-ECDSA-brainpoolP384r1-SHA512-cert			◒ 1/3		◒ 1/3			◒ 1/3				
id-MLDSA87-Ed448-SHAKE256-cert			◒ 1/3		◒ 1/3			◒ 1/3				
id-MLDSA87-RSA3072-PSS-SHA512-cert			◒ 1/3		◒ 1/3			◒ 1/3				
id-MLDSA87-RSA4096-PSS-SHA512-cert			◒ 1/3		◒ 1/3			◒ 1/3				
id-MLDSA87-ECDSA-P521-SHA512-cert			◒ 1/3		◒ 1/3			◒ 1/3				

In the PQ Hackathon, Crypto4A, CryptoNext, Safelogic, OpenSSL, Entrust and CHT are working together.
We expect the above table to solidify with interoperability testing results!

Working Implementations – Dynamic Hackathon Results

-- Composite KEM

✅ = passing all verifiers
◒ = passing some verifiers
○ = not passing any verifiers
Columns represent producers who submitted artifacts. Verifiers are not listed in this table, but are listed in the broken-out tables below.

-	bc	carl-redhound	cht	composite-kem-ref-impl	composite-sigs-ref-impl	corey-digicert	crypto4a	cryptonext	entrust	ossl35	safelogic	seventhsense.ai
id-MLKEM768-RSA2048-SHA3-256-cert				✅ 1/1				✅ 1/1				
id-MLKEM768-RSA3072-SHA3-256-cert				✅ 1/1				✅ 1/1				
id-MLKEM768-RSA4096-SHA3-256-cert				✅ 1/1				✅ 1/1				
id-MLKEM768-X25519-SHA3-256-cert				✅ 1/1				✅ 1/1				
id-MLKEM768-ECDH-P256-SHA3-256-cert				✅ 1/1				✅ 1/1				
id-MLKEM768-ECDH-P384-SHA3-256-cert				✅ 1/1				✅ 1/1				
id-MLKEM768-ECDH-brainpoolP256r1-SHA3-256-cert				✅ 1/1				✅ 1/1				
id-MLKEM1024-RSA3072-SHA3-256-cert				✅ 1/1				✅ 1/1				
id-MLKEM1024-ECDH-P384-SHA3-256-cert				✅ 1/1				✅ 1/1				
id-MLKEM1024-ECDH-brainpoolP384r1-SHA3-256-cert				✅ 1/1				✅ 1/1				
id-MLKEM1024-X448-SHA3-256-cert				✅ 1/1				✅ 1/1				
id-MLKEM1024-ECDH-P521-SHA3-256-cert				✅ 1/1				✅ 1/1				

Working Implementations – Automated CMS results!

✅ = passing all verifiers
◒ = passing some verifiers
○ = not passing any verifiers
Columns represent producers who submitted artifacts. Verifiers are not listed in this table, but are listed in the broken-out tables below.

-	carl-redhound	cryptonext	ossl35
ML-DSA-44		✅ 1/1	✅ 1/1
ML-DSA-65		✅ 1/1	✅ 1/1
ML-DSA-87		✅ 1/1	✅ 1/1
SLH-DSA-SHA2-128s		✅ 1/1	✅ 1/1
SLH-DSA-SHA2-128f		✅ 1/1	✅ 1/1
SLH-DSA-SHA2-192s		✅ 1/1	✅ 1/1
SLH-DSA-SHA2-192f		✅ 1/1	✅ 1/1
SLH-DSA-SHA2-256s		✅ 1/1	✅ 1/1
SLH-DSA-SHA2-256f		✅ 1/1	✅ 1/1
SLH-DSA-SHAKE-128s		✅ 1/1	✅ 1/1
SLH-DSA-SHAKE-128f		✅ 1/1	✅ 1/1
SLH-DSA-SHAKE-192s		✅ 1/1	✅ 1/1
SLH-DSA-SHAKE-192f		✅ 1/1	✅ 1/1
SLH-DSA-SHAKE-256s		✅ 1/1	✅ 1/1
SLH-DSA-SHAKE-256f		✅ 1/1	✅ 1/1
ML-KEM-512	✅ 1/1	✅ 1/1	
ML-KEM-768	○ 0/1	✅ 1/1	
ML-KEM-1024	✅ 1/1	✅ 1/1	

Thank You