Composite KEM – 04 Updates

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

July 2024

ENTRUST

SECURING A WORLD IN MOTION

Interop-breaking changes

KEM Combiner construction changes

- Adjusted the combiner to be compliant with NIST SP800-56C as per https://mailarchive.ietf.org/arch/msg/spasm/nlyQF1i7ndp5A7zzcTsdYF_S9mI/ -- also aligns with X-Wing.
- The combiner is now:

```
KDF(counter || tradSS || mlkemSS || tradCT || tradPK ||
        domSep, outputBits)
```


ENTRUST

KEM Combiner construction changes

- Adjusted the combiner to be compliant with NIST SP800-56C as per https://mailarchive.ietf.org/arch/msg/spasm/nlyQF1i7ndp5A7zzcTsdYF_S9mI/ -- also aligns with X-Wing.
- The combiner is now:

```
KDF(counter || tradSS || mlkemSS || tradCT || tradPK ||
        domSep, outputBits)
```

KDF is now SHA3-256/384/512,
and no longer KMAC.
Aligns with X-Wing and openpgp.

ENTRUST

KEM Combiner construction changes

- Adjusted the combiner to be compliant with NIST SP800-56C as per https://mailarchive.ietf.org/arch/msg/spasm/nlyQF1i7ndp5A7zzcTsdYF_S9mI/ -- also aligns with X-Wing.
- The combiner is now:

```
KDF(counter || tradSS || mlkemSS || tradCT || tradPK ||
        fixedInfo, outputBits)
```

The fixed value “0x00000001” Here purely for compliance with SP.800-56Cr2.

UPDATE: 7/25/2024
Quynh points out that the counter can be omitted when KDF is a single-round hash function.
(thanks @Falko)
I will synch up with Falko.

ENTRUST

KEM Combiner construction changes

- Adjusted the combiner to be compliant with NIST SP800-56C as per https://mailarchive.ietf.org/arch/msg/spasm/nlyQF1i7ndp5A7zzcTsdYF_S9mI/ -- also aligns with X-Wing.
- The combiner is now:

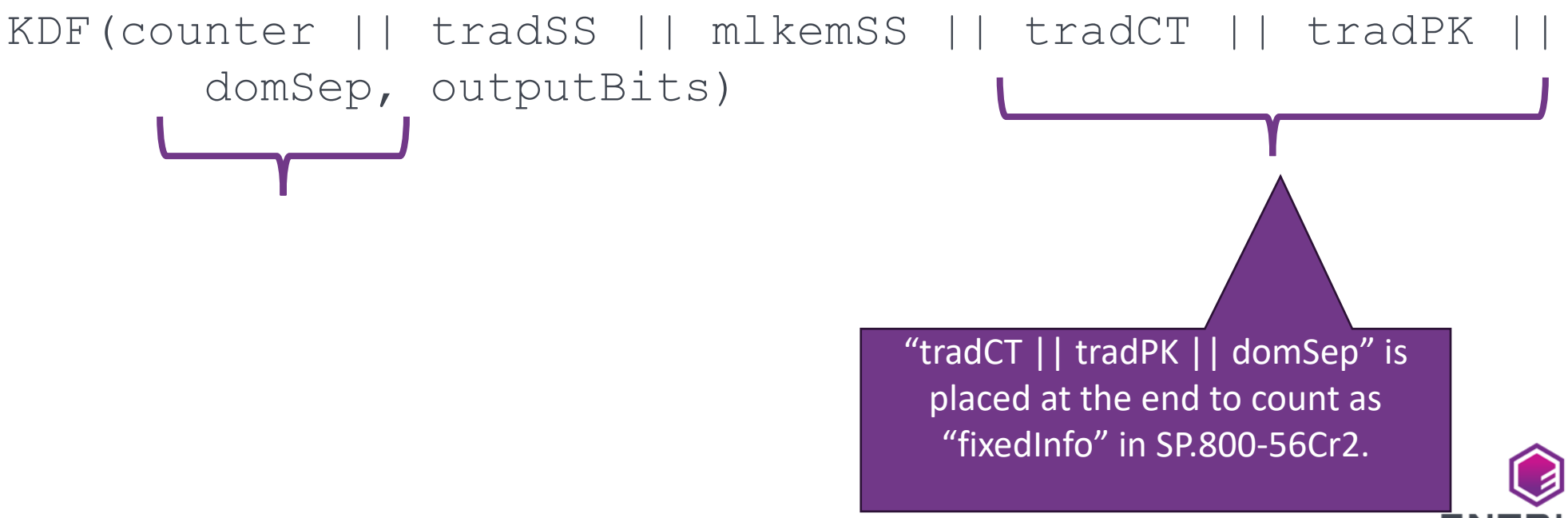
```
KDF(counter || tradSS || mlkemSS || tradCT || tradPK ||
        domSep, outputBits)
```

Trad (RSA, ECC) first so that it is the one that gets FIPS credit under SP.800-56Cr2.

UPDATE: 7/25/2024
I had misread Quynh's on-list message, and actually either order of "tradsSS" and "mlkemSS" is acceptable under SP.800-56Cr2.
So I will reverse this change for consistency with the rest of the document.
(thanks @Falko)

KEM Combiner construction changes

- Adjusted the combiner to be compliant with NIST SP800-56C as per https://mailarchive.ietf.org/arch/msg/spasm/nlyQF1i7ndp5A7zzcTsdYF_S9mI/ -- also aligns with X-Wing.
- The combiner is now:



ENTRUST

KEM Combiner construction changes

- Adjusted the combiner to be compliant with NIST SP800-56C as per https://mailarchive.ietf.org/arch/msg/spasm/nlyQF1i7ndp5A7zzcTsdYF_S9mI/ -- also aligns with X-Wing.
- The combiner is now:

```
KDF(counter || tradSS || mlkemSS || tradCT || tradPK ||
      domSep, outputBits)
```

KEM Combiner construction changes

- Adjusted the combiner to be compliant with NIST SP800-56C as per https://mailarchive.ietf.org/arch/msg/spasm/nlyQF1i7ndp5A7zzcTsdYF_S9mI/ -- also aligns with X-Wing.
- The combiner is now:

```
KDF(counter || tradSS || mlkemSS || tradCT || tradPK ||
      domSep, outputBits)
```

The X-wing paper proves that this is sufficient to achieve desired binding properties when trad is an ECDH variant. Early review by Douglas Stebila, Dierdre Connolly, suggest that this should also hold for RSA-OAEP, but a proof is needed.

Inclusion of tradCT and tradPK, but not mlkemCT and mlkemPK aligns with X-Wing.

Changes affecting Implementation Interoperability

- Specified the fixedInfo domain separators as the DER encoded object identifiers.
  - This will require early OID allocation by IANA so that we can roll the domain separator values.

Composite KEM AlgorithmID	Domain Separator (in Hex encoding)
id-MLKEM512-ECDH-P256	060B6086480186FA6B50050201
id-MLKEM512-ECDH-brainpoolP256r1	060B6086480186FA6B50050202
id-MLKEM512-X25519	060B6086480186FA6B50050203

ENTRUST

RSA-KEM -> RSA-OAEP

- Replaced RSA-KEM [RFC5990] with RSA-OAEP.
- Added a section "Promotion of RSA-OAEP into a KEM".

-04 has a typo here.

```
RSAOAEPKEM.Encaps(pkR):
    shared_secret = SecureRandom(ss_len)
    enc = RSA-OAEP.Encrypt(pkR, shared_secret)

    return enc, shared_secret
```

- Should be straightforward, but requires cryptographic review as to whether this provides the desired IND-CCA2 and PK/CT binding properties.
  - Thanks to Peter C, Sophie Schmieg, Deirdre Connolly, Douglas Stebila, Falko, and others for providing comments so far.

ENTRUST

DHKEM

- Removed references to I-D.ounsworth-lamps-cms-dhkem since we'll just inline a simplified version of RFC9180's DHKEM.

```
DHKEM.Encaps(pkR):
  skE, pkE = GenerateKeyPair()
  shared_secret = DH(skE, pkR)
  enc = SerializePublicKey(pkE)

  return enc, shared_secret
```

- Simplifications from 9180 DHKEM: don’t need to bind pkR, enc, or use 9180’s labelledExtract() for a domain separator because we do all of that at the Combiner level.

ENTRUST

Other misc. interop changes

- In the "Use in CMS > Underlying Components" section, the MLKEM768 combinations were lifted from id-aes192-Wrap to id-aes256-Wrap because the latter is believed to have better general adoption.
  - Thanks Dan van Geest.

ENTRUST

New OIDs?

- Since we've broken interop, we should have issued all new OIDs, but as we're not aware of any mature implementations, we decided to be lazy.
  - (We can start rev'ing OIDs with the next version)

ENTRUST

Non-interop-breaking changes

Misc editorial changes

- Since all combinations use ML-KEM; changed the document title to "Composite ML-KEM".
- Added an appendix "Fixed Component Algorithm Identifiers" -- not finished, needs more work.
  - The idea is to provide explicitly what the AlgorithmIDs (including parameters) of the components should be, should an implementation need them, for example to call into its crypto library. Providing them explicitly avoids ambiguity and implementation errors.

ENTRUST

CFRG KEM Combiners

- Mike Ounsworth participated in the CFRG KEM Combiners design team, which recently produces guidance to CFRG:
  - Thread: [CFRG] KEM combiners design team output
    - https://mailarchive.ietf.org/arch/msg/cfrg/CwrVvm-J7o85TEWkG9RJxZwfXDY/
  - It calls for a CFRG document with:
    - Studies the security requirements of a KEM combiner,
    - Defines 3 explicit instantiations:
      - (I) a hybrid of P-256 and ML-KEM-768,
      - (II) a hybrid of X25519 and ML-KEM-768, and,
      - (III) a hybrid of P-384 and ML-KEM-1024.

ENTRUST

CFRG KEM Combiners

- Anyway, for now I've just removed reference to draft-ounsworth-cfrg-kem-combiners so that we don't end up in a downref situation if we want to get this LAMPS doc out before CFRG has finished.

- Given that this LAMPS document – particularly the RSA part – is ahead of CFRG and published research, how do we want to proceed?

ENTRUST

Open Questions

Binding public keys

- Following the security proof in the X-Wing paper, we are now binding the recipient public key into the KDF.
- Question: is it a safe assumption that a crypto module doing a composite decryption will have access to the RSA or ECDH public key?
  - Think smartcard, PKCS#11, etc.
  - Sorta boils down to whether public keys are typically stored inside private keys, or whether decryption routines have access to their own public key certs.
  - Maybe the answer is that when doing composite KEM, pubKey is a mandatory part of the privKey, even for the RSA/ECC component.
  - Need community feedback on this.

ENTRUST

KDF = SHA3 … what about SHA2?

- Deb Cooley and Joe Salowey brought the excellent point that not all crypto libs will have easy access to SHA3 at the combiner level.
- We have chosen SHA3 because that aligns security analysis with X-Wing and openpgp-pqc.
- We could also add HMAC-SHA2 variants, but that will 2x the numbers of algs that we're registering
  - (note: you can't just replace SHA3 with SHA2 without losing security properties; it has to be HMAC-SHA2).

- **Deb and Joe are probably right.**
  Does the WG want us to 2x the size of the list?

ENTRUST

Timing?

- Presumably we want to get this out pretty quickly after FIPS 203? – ie together withraft-ietf-lamps-kyber-certificates and draft-ietf-lamps-cms-kyber.
- Or, since the KEM Combiner is novel cryptography, do we want to delay this for more review?
- Or, do we want to wait entirely for the parallel CFRG KEM Combiners drafts?

- My vote: get this out quickly so that real prod environments can start protecting themselves. If CFRG / academia finds problems with the construction then we'll do a –bis.
  - IE helping people get started on PQC sooner is worth the risk of having to do a –bis later.

ENTRUST

PEM Samples

- We are not aware of any implementation mature enough to provide us with samples for the appendix.
- The OQS team is working on an implementation. Maybe Entrust, or BouncyCastle, or Carl will get there first?

ENTRUST

Other open github issues

- Add back text to not reuse component keys
- Create RSA-4096 combos
- Russ feedback on RSA-OAEP
- Issues with the ASN.1 module
- Synchronize sections and writing style of composite sigs and kems
- ML-KEM public keys should be unwrapped BIT STRINGS with no ASN.1 typing
- Add a new section: explicitly list SPKI AlgIds

ENTRUST

Thank You

entrust.com

© Entrust Corporation