ML-KEM for TLS 1.3

draft-connolly-tls-mlkem-key-agreement

https://datatracker.ietf.org/doc/draft-connolly-tls-mlkem-key-agreement/
https://github.com/dconnolly/draft-connolly-tls-mlkem-key-agreement

A pure-PQ ciphersuite for TLS 1.3

- No purely post-quantum ciphersuites
- Fills in the other side of draft-ietf-tls-hybrid-design
- Needed because there are no documents that describe KEM-only key agreement in TLS
- If PQ-only works for your applications, clean key agreement, no hybrid duplicate shares or mixing and matching logic
- ML-KEM-1024 supports FIPS users who need to comply with the CNSA 2.0 draft
- I want to be able to do it 🐴

New NamedGroups: MLKEM512, MLKEM768, MLKEM1024

```
enum {

      ...,

       /* ML-KEM Key Agreement Methods */
       mlkem512(0x0200),
       mlkem768(0x0201),
       mlkem1024(0x0202)

      ...,

} NamedGroup;
```

Codepoints allocated:

512	MLKEM512	Y	N	[draft-connolly-tls-mlkem-key-agreement-03]	FIPS 203 version of ML-KEM-512
513	MLKEM768	Y	N	[draft-connolly-tls-mlkem-key-agreement-03]	FIPS 203 version of ML-KEM-768
514	MLKEM1024	Y	N	[draft-connolly-tls-mlkem-key-agreement-03]	FIPS 203 version of ML-KEM-1024

Client sends encaps key, server replies with ciphertext

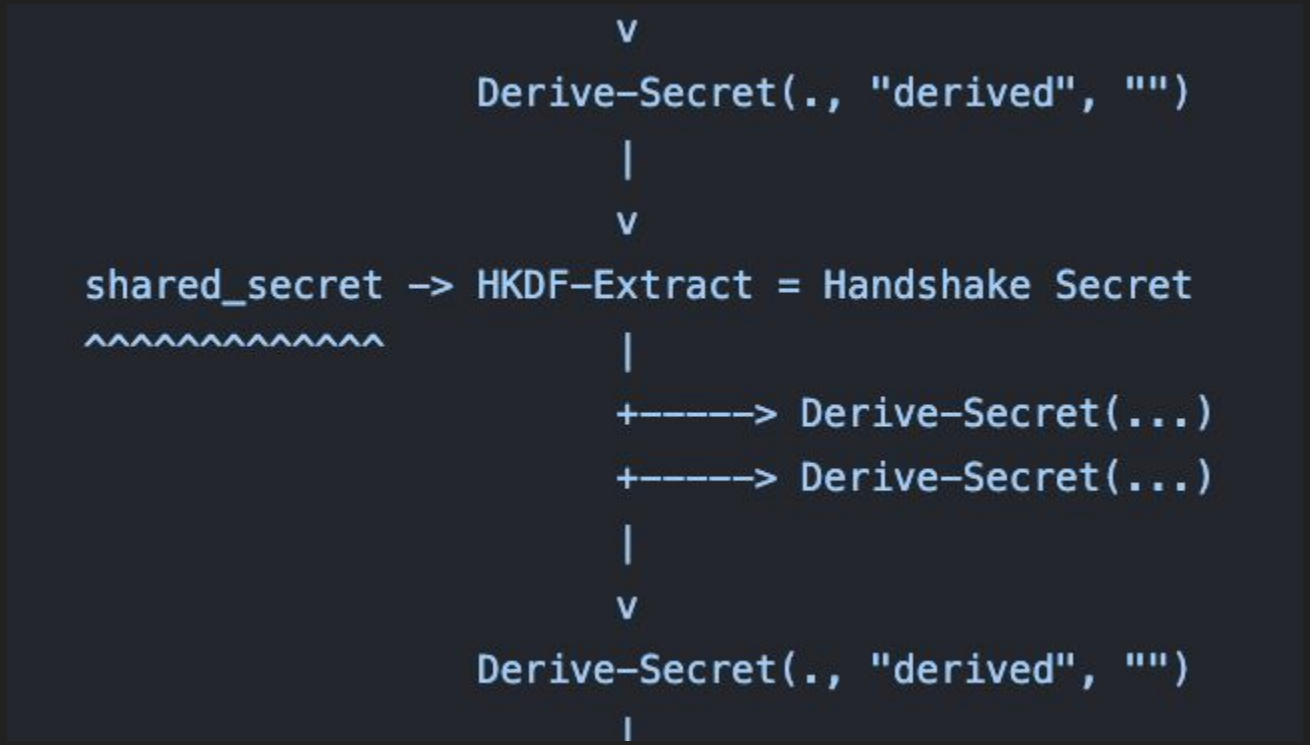
```
struct {
    NamedGroup group;
    opaque key_exchange<1..2^16-1>;
} KeyShareEntry;
```

These are transmitted in the `extension_data` fields of `KeyShareClientHello` and `KeyShareServerHello` extensions:

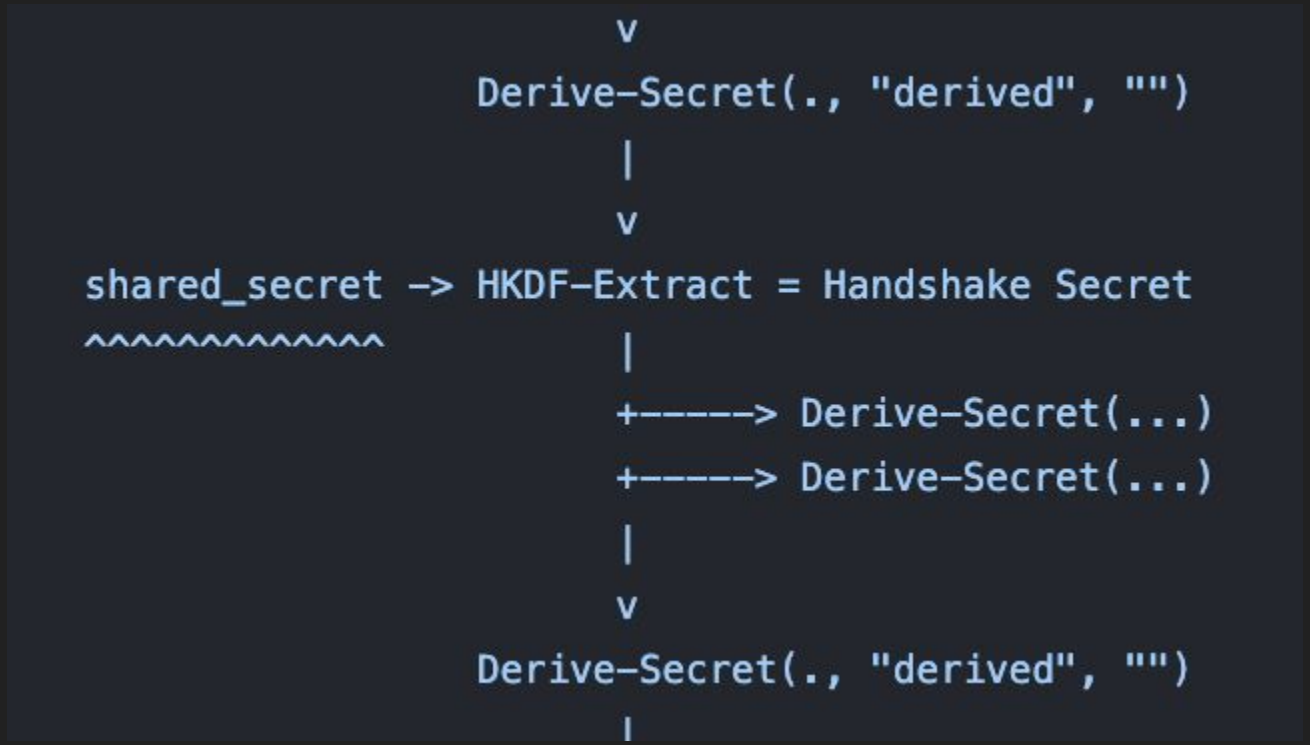
```
~~~
    struct {
        KeyShareEntry client_shares<0..2^16-1>;
    } KeyShareClientHello;

    struct {
        KeyShareEntry server_share;
    } KeyShareServerHello;
~~~
```

KEM shared secret is input to Handshake Secret derivation



KEM shared secret is input to Handshake Secret derivation



Etc

- Added MLKEM-512 per request
- Aligned thrown errors with draft-kwiatkowski-tls-ecdhe-mlkem
- Preliminary implementation done for rustls fork
- Cited by CNSA 2.0 TLS Profile draft: draft-becker-cnsa2-tls-profile/
- DTLS: Y
- Recommended: N
- Make ephemeral a MUST, not an implied SHOULD, by 8446?
  - Related: https://github.com/post-quantum-cryptography/draft-kwiatkowski-tls-ecdhe-mlkem/pull/25
- 🎉

ML-KEM for TLS 1.3

draft-connolly-tls-mlkem-key-agreement

https://datatracker.ietf.org/doc/draft-connolly-tls-mlkem-key-agreement/
https://github.com/dconnolly/draft-connolly-tls-mlkem-key-agreement