# CFRG @ IETF-120

## KEM combiners design team output

https://mailarchive.ietf.org/arch/msg/cfrg/CwrVvm-J7o85TEWkG9RJxZwfXDY/

# Who to blame

- Aron Wussler
- Bas Westerbaan
- Deirdre Connolly
- Mike Ounsworth
- Nick Sullivan
- Stephen Farrell

# Going in...

- DT wasn’t tasked with producing “the answer” but with setting requirements for how the RG might produce answer(s)
- Based on RG list and DT discussion
- Valid but conflicting requirements, no single answer will do
- Lack of clarity on binding and KEM security properties
- Many possibe choices
- Some people don’t want to wait for the heat-death of the universe

# Output...

The DT recommend CFRG produce a “Hybrid PQ/T Key Encapsulation Mechanisms” document meeting the following requirements:

# “Hybrid PQ/T Key Encapsulation Mechanisms” document (1/4)

(A) Identify which KEM security properties are IETF-relevant, and provide a terse overview of those security properties (eg. IND-CCA, LEAK-BIND-K-PK, HON-BIND-K-CT, etc), as well as security properties unique to hybrid KEMs (component key material reuse between hybrid and non-hybrid uses or between multiple hybrids, one component is malicious while the other is honest, etc) with reference to literature, and put into context with real-world attacks. From that, give guidance on a sensible baseline.

# “Hybrid PQ/T Key Encapsulation Mechanisms” document (2/4)

(B) Provide a terse overview of well-reviewed techniques that are options to safely produce the concrete combinations in (C), and which security properties are achieved given those of the constituents.

# “Hybrid PQ/T Key Encapsulation Mechanisms” document (3/4)

(C) Provide an initial number of explicit PQ/T hybrid KEMs using techniques from (B) that reach the baseline set in (A), and should include:

(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.

These hybrids should be accompanied by pseudocode and test vectors.

# “Hybrid PQ/T Key Encapsulation Mechanisms” document (4/4)

- This list includes two options at the ~128-bit security level (due to current implementation/deployment trends) and one at a higher level.
- The DT would be happy for the RG to omit C(I) above should there not be significant implementations for which C(II) and C(III) are hard. The DT did not attempt to survey implementations to determine this.
- There is demand for other hybrid variants that either use different primitives (RSA, NTRU, Classic McEliece, FrodoKEM), parameters, or that use a combiner optimized for a specific use case. The DT recommends the work outlined in (C) is done in a first document, and other use cases could be covered in subsequent documents.

# Possible Next Step

- If/as RG/chairs agree with the above, chairs might find some victims willing to write that document and try get it done soon