Hybrid KEM Specification

April 16, 2024

Abstract

The hybrid key encapsulation mechanisms (KEM) algorithm provides an approach that combines a post-quantum-computing resistant KEM algorithm of Kyber with a classic algorithm in a way that when one of the algorithm is compromised, the security strength of the resulting shared secret is still as strong as the yet uncompromised algorithm.

This combination allows the secure use of a PQC algorithm even though it is yet new and must still receive additional cryptoanalysis. Even though a later cryptoanalysis shows that the PQC algorithm is weak, its weakness is offset by the used classical key agreement algorithm.

Conversely, if the advancements in quantum computers allows breaking the classic key agreement algorithm, its weakness is offset by the used PQC algorithm.

The presented algorithm can be used as a direct drop-in-replacement for a standalone Kyber KEM use case. The only difference is the enlarged sizes of all data points. It is compliant with SP800-56C rev 2, specifically section 2 which permits a hybrid shared secret. In addition, it is compliant to [1] by specifying KMAC as KDF.

The selection of X25519 for the classic key agreement mechanism is arbitrary since the reference implementation provided with leancrypto contained an X25519 implementation. The presented mechanism equally works with ECDH based on NIST prime field curves or Brainpool curves. The algorithm also works with FFC-DH. If a user wants to use those classic algorithms instead, the references to X25519 would need to be replaced by references to the selected classic Diffie-Hellman algorithm.

Contents

1 Hybrid KEM - Kyber & X25519

In addition to the sole use of Kyber KEM, a hybrid mechanism using X25519 can be devised that acts as a drop-in-replacement for Kyber KEM. In this case, a PQC algorithm is merged with a classical key establishment algorithm. The basis is the enhancement of the Kyber KEM encapsulation and decapsulation algorithms as follows.

When using the hybrid KEX algorithm, instead of the sole KEM encapsulation and decapsulation operations, the hybrid variants are used that are outlined in the subsequent subsections. In addition, the Kyber KEX data along with the X25519 data is exchanged in the same manner as outlined for the standalone Kyber KEX. Thus, the KEX operation is not re-iterated here.

The presented algorithm ensures that even if one algorithm is compromised, the resulting shared secret is still cryptographically strong compliant with the strength of the uncompromised algorithm. However, it is to be noted that Kyber may have a cryptographic strength of up to 256 bits when using Kyber 1024. On the other side, the cryptographic strength of X25519 is significantly lower, between 80 and 128 bits, depending on the analysis approach.

1.1 Hybrid KEM Key Generation

As part of the hybrid KEM key generation, the following steps are performed:

1. Generation of the Kyber key pair yielding the Kyber `pk_kyber` and `sk_kyber`.
2. Generation of the X25519 key pair yielding the X25519 `pk_x25519` and `sk_x25519`.

Both public keys and both secret keys are maintained together such that every time the hybrid KEM requires a public key, the Kyber and X25519 public keys are provided. The same applies to the secret keys.

Thus the following holds:

- `pk_hybrid` = `pk_kyber` || `pk_x25519`
- `sk_hybrid` = `sk_kyber` || `sk_x25519`

Both, `pk_hybrid` and `sk_hybrid` are the output of the hybrid KEM key generation operation.

1.2 Hybrid KEM Encapsulation

The hybrid KEM encapsulation applies the following steps using the input of the hybrid KEM public key `pk_hybrid`.

1. Invocation of the Kyber encapsulation operation to generate the Kyber shared secret `ss_kyber` and the Kyber ciphertext `ct_kyber` using the `pk_kyber` public key presented with `pk_hybrid`.

2. Generation of an ephemeral X25519 key pair `pk_x25519_e` and `sk_x25519_e`.

3. Invocation of the X25519 Diffie-Hellman operation with the X25519 public key `pk_x25519` provided via `pk_hybrid` and the ephemeral secret key `sk_x25519_e`. This generates the shared secret `ss_x25519`.

4. Secure deletion of the `sk_x25519_e` ephemeral secret key.

The operation returns the following data:

- Public data: `ct_hybrid` = `ct_kyber` || `pk_x25519_e`
- Secret data: `ss_hybrid` = `ss_kyber` || `ss_x25519`

The data `ct_hybrid` is to be shared with the peer that performs the decapsulation operation.

On the other hand `ss_hybrid` is the raw shared secret obtained as part of the encapsulation operation and must remain secret. It is processed with a KDF as outlined in section 1.4.

1.3 Hybrid KEM Decapsulation

The hybrid KEM decapsulation applies the following steps using the input of the hybrid KEM secret key `sk_hybrid` and the public data resulting from the hybrid KEM encapsulation operation `ct_hybrid`.

1. Invocation of the Kyber decapsulation operation to generate the Kyber shared secret `ss_kyber` by using `ct_kyber` present in `ct_hybrid` and the Kyber secret key `sk_kyber` found in `sk_hybrid`.

2. Invocation of the X25519 Diffie-Hellman operation with the X25519 secret key `sk_x25519` provided via `sk_hybrid` and the ephemeral public key `pk_x25519_e` provided via `ct_hybrid` which returns the shared secret `ss_x25519`.

The operation returns the following data:

- Secret data: `ss_hybrid` = `ss_kyber` || `ss_x25519`

The data of `ss_hybrid` is the raw shared secret obtained as part of the encapsulation operation and must remain secret - it is the same data as calculated during the encapsulation step. It is processed with a KDF as outlined in section 1.4.

1.4 Hybrid KEM Shared Secret Derivation

To obtain a shared secret of arbitrary length that can be used as key material, a key derivation function is used allowed by [2] section 2:

- The chosen and KDF is based on [1].

- In addition, the input to the KDF is formatted such that the entire hybrid KEM construction is compliant with [2] assuming that Kyber KEM is the approved algorithm and X25519 provides an auxiliary key agreement mechanism. Thus, section 2 of [2] with its requirement `Z' = Z || T` is fulfilled by defining the "standard" shared secret `Z` is provided by Kyber and that the auxiliary shared secret `T` is provided by X25519.

Considering that Kyber uses SHAKE / SHA-3 in its internal processing, the selected KDF is KMAC256 as defined in [1]. KMAC is invoked as follows:

```
KMAC256(K = ss_hybrid,
        X = ct_hybrid,
        L = requested SS length,
        S = "Kyber X25519 KEM SS")
```

When considering the structure of `ss_hybrid` and `ct_hybrid`, the KDF operates on the following specific data:

```
KMAC256(K = ss_kyber || ss_x25519,
        X = ct_kyber || pk_x25519_e,
        L = requested SS length,
        S = "Kyber X25519 KEM SS")
```

The KMAC customization string `S` is selected arbitrarily and can contain any string including the `NULL` string.

The result of the KDF is intended to be usable as key material for other cryptographic operations. That derived key material now contains the individual security strengths of both, Kyber and X25519. Both algorithms are used such that any security break of either algorithm will not impact the strength of the resulting shared secret of the respective other. By concatenating the individual shared secret values as input into the KDF, the result of the KDF will have the security strength of one algorithm even if the respective other algorithm is broken.

1.5 Hybrid KEX Algorithm

Using the hybrid KEM algorithm outlined in the preceding subsections, the hybrid KEX algorithm as specified in the documentation of the secure connection approach can be obtained by the following considerations: use of the Kyber KEX approach outlined at the beginning, but apply the following changes:

1. Replace all occurrences of `pk` with `pk_hybrid`.
2. Replace all occurrences of `sk` with `sk_hybrid`.
3. Replace all occurrences of `ss` with `ss_hybrid`.
4. Replace all occurrences of `ct` with `ct_hybrid`.

5. Replace all invocations of the Kyber standalone functions (key generation, encapsulation, decapsulation) with their respective hybrid variants outlined above.

This implies that the hybrid KEM as well as the hybrid KEX algorithm are usable as a direct drop-in-replacement for the standalone Kyber algorithm use case. The only difference is that the resulting data is larger as it contains the X25519 data as well.

This approach ensures that both mechanisms contributing to the hybrid KEX algorithm, Kyber and X25519, independently also provide the implied authentication. That means that even if one algorithm is considered to be compromised, not only the shared secret strength is preserved by the unencumbered other algorithm, but also the implied authentication is still guaranteed with the unencumbered algorithm.

References

[1] Lily Chen. *NIST SP 800-108r1, Recommendation for Key Derivation Using Pseudorandom Functions*. Revistion 1 edition, August, 2022.

[2] Richard Davis Elaine Barker, Lily Chen. *NIST Special Publication 800-56C Recommendation for Key-Derivation Methods in Key-Establishment Schemes*. Revistion 2 edition, August, 2020.

