Fundamentals

Key Encapsulation Mechanism (KEM)

A Key Encapsulation Mechanism is a cryptographic protocol used to securely transmit encryption keys over an unsecured network. It enables one party to generate a secret key and encapsulate it into a ciphertext, which can then be sent to another party.

Only the intended recipient, who possesses the appropriate private key, can decapsulate the ciphertext to retrieve the secret key.

Digital Signature Algorithm (DSA)

The Digital Signature Algorithm is a standard for digital signatures, which are used to verify the authenticity and integrity of digital messages or documents.

When a message is signed using DSA, it produces a signature that is unique to both the message and the sender's private key. Recipients can use the sender's public key to verify that the signature is valid and that the message has not been tampered with.

This process ensures that the message genuinely comes from the claimed sender and remains unaltered during transmission.

Module lattice

A module lattice is a mathematical structure consisting of points in a multidimensional space arranged in a regular, repeating pattern.

In the context of cryptography, lattices are used because they involve hard mathematical problems that are difficult to solve without specific information. The complexity of these problems, especially in high-dimensional lattices, provides a strong foundation for secure cryptographic systems.

Crystals-Kyber

Official website	Kyber (pq-crystals.org)
Main implementation	pq-crystals/kyber (github.com)
Standard name	Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM)

Crystals-Dilithium

Official website	Dilithium (pq-crystals.org)
Main implementation	pq-crystals/dilithium (github.com)
Standard name	Module-Lattice-Based Digital Signature Algorithm (ML-DSA)

General PQC libraries

Currently, there are no pure-Python implementations of Post-Quantum Cryptographic (PQC) algorithms. Instead, they are usually implemented in C.

This means for Python we need to:

- Use C versions of PQC algorithms
- Install C compilers
- Use tools like CFFI to integrate the C code into Python projects.

Additionally, the compiled binaries are specific to each platform, making it challenging to develop and deploy across different systems without recompiling the C code for each one.

Library	Language	Link	Description
quantcrypt	Python	quantcrypt · PyPI	Pre-compiled binaries
pypqc	Python	pypqc · PyPI	Partial Python bindings