Comparative Demo: RSA vs Kyber key exchange using Python.

The goal of this demo is to show how vulnerable the RSA key exchange method is in post-quantum cryptography, and why the Key Encapsulation Mechanism (KEM) model should be preferred in PQC.

Why is encrypting a symmetric key directly as in RSA vulnerable in the PQC era?

First lets look at how RSA key exchange work:

- Bob generates a random symmetric key (e.g., for RSA)
- He encrypts it with Alice's RSA public key.
- Alice decrypts it with her private key.

In this process the symmetric key is directly exposed to the mathematical structure of RSA.

Why it's vulnerable:

In the near future, quantum computers can run Shor's Algorithm, which can factor large integers as RSA relies on the difficulty of factoring large numbers. Tho, it can efficiently break RSA). Additionally, it can solve logarithms which can break ECC and DH.

This can result in the concept of Harvest Now, Decrypt Later attack. It means that if a quantum attacker captures an RSA-encrypted symmetric key (ciphertext) now, they can store it. Once a large quantum computer exists, they can break RSA and decrypt the symmetric key.

Why PQC KEMS are safe:

KEMs like kyber do not rely on factoring or discrete logs. Instead, they use lattice base problems, which are believed to be hard even for quantum computers. Additionally, the shared secret is derived from mathematical noise, not from a structure that can be algebraically reversed like RSA. Kyber never directly encrypts the symmetric key. Instead, both sides derive it from shared structured randomness, encapsulated in a way that can't be reversed efficiently.

How this process work with kyber:

1. Alice generates a Kyber public/private key pair.
2. Bob, use Alice's public key and does two things:

    - He generates a shared secret (randomly)
    - He created a ciphertext that encapsulates, meaning it hides and protects that shared secret.
3. Bob send the ciphertext to Alice
4. Alice uses her private key to decapsulate the ciphertext and recover the exact same shared secret.

Key exchange Comparison:

Classical RSA

- Slower key generation and encryption/decryption
- Smaller ciphertext but larger keys
- Secure today, but not post quantum safe

Post Quantum Kyber768

- Much faster key generation and operations
- Larger ciphertext and keys
- Resistant to quantum attacks (NIST-selected)

Final Report:

Encapsulation in cryptography refers to the process of generating a shared secret and producing a ciphertext. These two must be binded together to protect the key, so only the holder of the corresponding private key can decapsulate it. Unlike classical encryption, KEMs do not encrypt pre-existing messages or keys. Instead, they derive and protect a shared key in one operation, making them efficient and secure even against quantum adversaries.