# Fortiq Overview — Quantum-Resistant Encryption Engine

Fortiq is a lightweight, developer-friendly encryption system designed to bring quantum resistance to file security, software systems, and internal IT processes.  
  
Built using the Kyber512 Key Encapsulation Mechanism (KEM) — one of NIST’s officially selected post-quantum cryptographic standards — Fortiq combines modern asymmetric security with symmetric AES-256 encryption in a fast, auditable, and portable CLI package.

## How It Works

1. Key Generation (Kyber512) — Lattice-based cryptography generates a quantum-resistant keypair.

2. Shared Secret Encapsulation — Secure Kyber-based handshake to generate a shared secret.

3. AES-256-CBC Hybrid Layer — Shared secret is hashed (SHA-256) and used with AES-CBC.

4. File Output — JSON container with encrypted data, Kyber ciphertext, AES IV, and optional vault label.

## Security Properties

|  |  |
| --- | --- |
| Feature | Description |
| Quantum Resistance | Based on Kyber512, NIST PQC standard |
| Symmetric Encryption | AES-256-CBC with SHA-256 derived key |
| Integrity Controls | Vault tracking and key-specific file scoping |
| No Persistent Daemons | All key management is manual and local |
| Stateless CLI | No background processes, clean use |
| Uninstall Support | Built-in key deletion and secure cleanup |

## Technical Specs

• Language: Python 3.8+  
• Dependencies: pqcrypto, pycryptodome  
• Mode: CLI + local file encryption (no network)  
• Key Format: Kyber public/private keypairs  
• Output Format: JSON (hex-encoded ciphertext)

## For Security Teams

• No runtime persistence  
• Easy to audit — under 500 lines of code  
• AES-256 protects against practical attacks  
• Kyber ensures forward secrecy and PQ safety  
• Can be pipelined or wrapped into an API

## Typical Use Cases

• Secure file storage and transport  
• Privacy-first backups or vaults  
• Encrypted messaging systems  
• Developer workflows in regulated environments

## Example Commands

Generate Keys:  
python quantum\_resistant\_kem.py gen-keys --name alice  
  
Encrypt:  
python quantum\_resistant\_kem.py encrypt --infile myfile.txt --outfile myfile.enc.json --keyname alice --vault finance  
  
Decrypt:  
python quantum\_resistant\_kem.py decrypt --infile myfile.enc.json --outfile restored.txt --keyname alice  
  
Uninstall:  
python quantum\_resistant\_kem.py uninstall --keyname alice