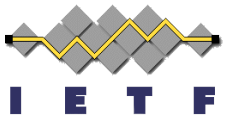
QSC Key Identification and Serialization

draft-uni-qsckeys

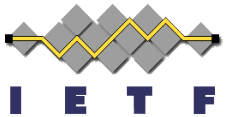


Mike Osborne
IETF 113, Viena
March 21, 2022

What's the deal with PQC ?

- NIST PQC submissions have single serialized structure for keys
  - Resulting from the NIST evaluation API – not design
  - Sounds nice in theory – no parsing of key formats
- Implicit Key structure depends on :
  - Algorithm strength (Parameter sets - 1,2,3,4,5)
  - Algorithm version where algorithm evolves (Round 2/ Round 3/ Final ,..)
  - HW alternatives (SHA2 vs SHAKE) (AES-256)
- The result:
  - Many possible algorithm variants for a single scheme (E.g. Dilithium)
  - An explosion in variants for hybrid schemes
  - Interop testing problems when scheme evolve

What's the deal with PQC ?

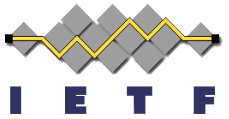


- Nice in theory to tie an algorithm identifier to an implicit key format
  - Removes parsing vulnerabilities
- In practice :
  - Algorithms already being deployed
  - We see that keys for many legacy systems simply DO NOT FIT
  - Getting keys safely to algorithm engines not addressed by algorithm designers
    - Algorithm performance tests conveniently ignore key provisioning times
- Most schemes detail key compression alternatives
  - E.g. Seeds that can be expanded
  - These compression alternatives are required
  - Lets not repeat mistakes made in the past

What are our goals?

- Goal: Make sure that PQC algorithms can be used by the largest number of applications – ***quickly and safely***
  - Learning from past mistakes with ECC in reducing multiple standards
  - Allowing as much interoperability and experience building in parallel to NISTs final standard
- Approach
  - To ensure correct communication key formats are serialized.
  - To recognize the need for key compression and deal with it early in a safe way
  - To identify the best identification algorithm/key identification approach
  - Higher level (than crypto API) considerations:
    - How to store / load the key from key formats (ordering)
    - Optional choices (for performance / size considerations)

Solution direction



- An RFC specifying key formats will help
  - Help manage algorithm versions and compatibility in key formats
  - Help interoperability of both testing and integration
  - Help make choices in future standards clear
  - Help prevent delays in integration and adoption
- Draft RFC “PQC Key Identification and Serialization” is shared with the cryptographic community
- Has received feedback from most scheme authors
- Has already demonstrated mistakes and lack of clarity in some scheme specifications

Next Steps

- Debate parsing complexity tradeoff for structure definitions
- Align with NIST on algorithm OIDs
- Align with ETSI / OASIS SAM / PKCS11 / KMIP TC / …
- Resolve issues around hybrid modes (IP, key serialization)
- Encouraged format for migration
- Adjust Algorithm scope
  - + Alternate Round 4 candidates
  - - Round 3 loosers

Resources

Work Item Repository (Issues, PRs, Details):
https://github.com/Quantum-Safe-Collaboration/qsc-key-rfc

Datatracker: https://datatracker.ietf.org/doc/html/draft-uni-qsckeys-00.html

NIST PQC:
https://csrc.nist.gov/projects/post-quantum-cryptography

Relevant KEM Schemes:
https://pq-crystals.org/kyber/
https://ntru.org/
https://www.esat.kuleuven.be/cosic/pqcrypto/saber/
https://classic.mceliece.org/

Relevant Signature Schemes:
https://pq-crystals.org/dilithium/
https://falcon-sign.info/
https://www.pqcrainbow.org/