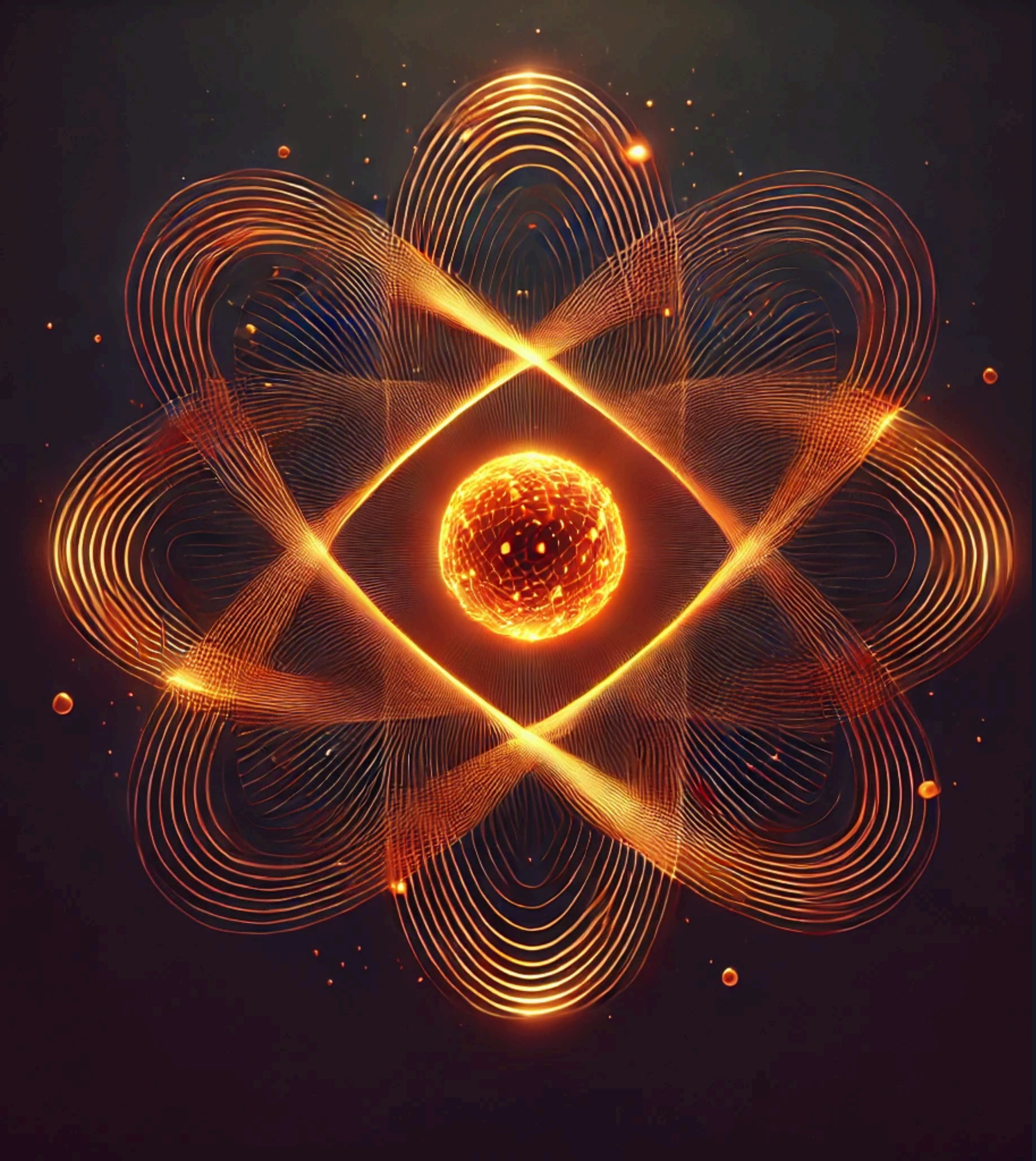

The current state of Post Quantum Cryptography

Kartik Kulkarni

13-10-2024
MiniDebConf24, Cambridge

Session Plan

- High level overview of PQC
- Understand the terms around PQC
- Choosing the appropriate algorithms
- Plan for migration
- Questions and Discussion

The Quantum Threat

- Quantum computers are getting better!
    - IBM condor (1121 physical qubits)
- Asymmetric cryptography at threat
    - Shor's algorithm will reduce prime factorisation problem from exp to poly time
    - RSA / ECC at risk
- Symmetric cryptography at half security level
    - Grover's search will allow quadratic boost for brute forced search
    - AES-256 / SHA-512 will be equiv to 128/256 bit strength

Expected time to break classical algorithms

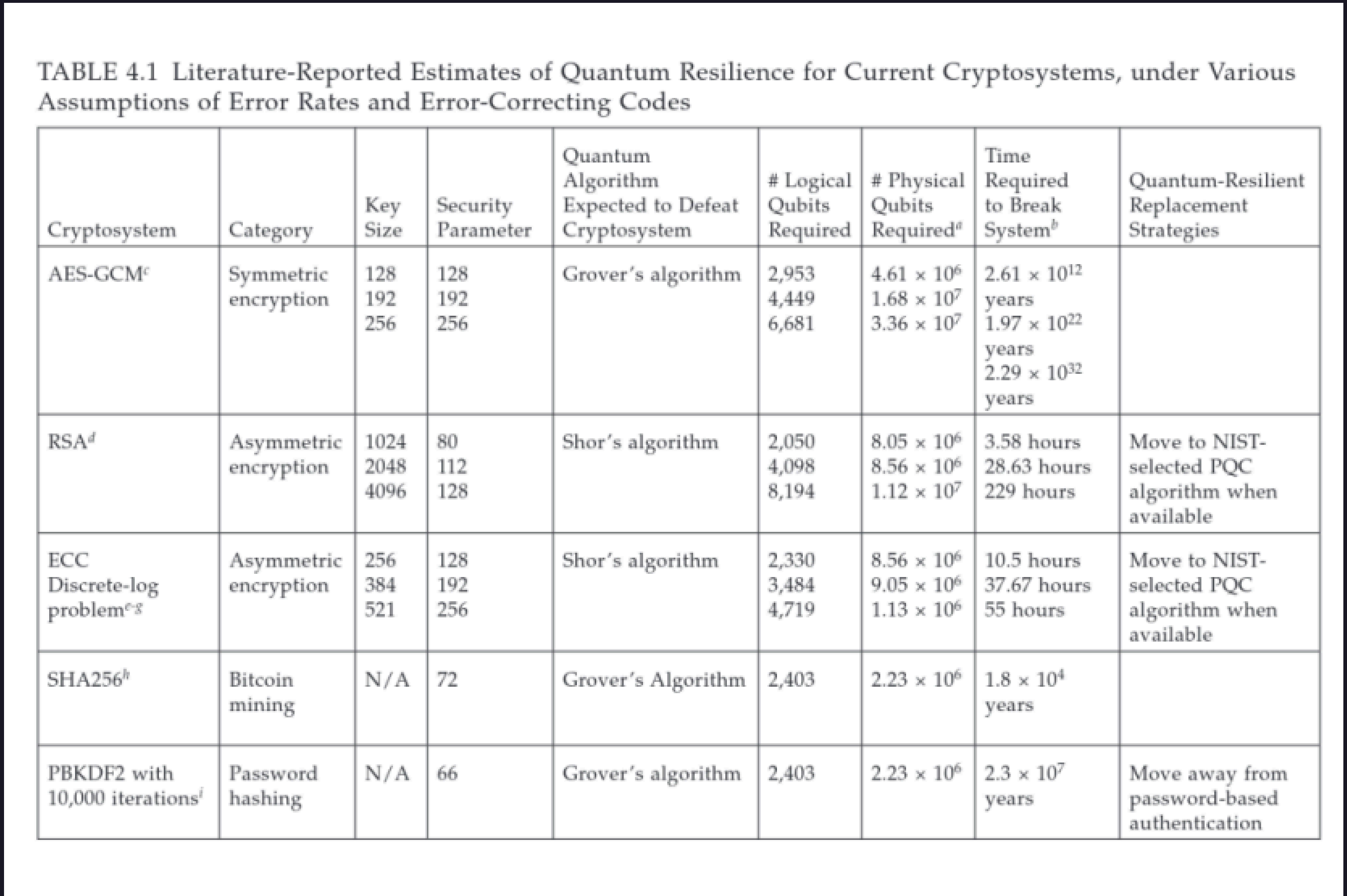

TABLE 4.1 Literature-Reported Estimates of Quantum Resilience for Current Cryptosystems, under Various Assumptions of Error Rates and Error-Correcting Codes

Cryptosystem	Category	Key Size	Security Parameter	Quantum Algorithm Expected to Defeat Cryptosystem	# Logical Qubits Required	# Physical Qubits Required[a]	Time Required to Break System[b]	Quantum-Resilient Replacement Strategies
AES-GCM[c]	Symmetric encryption	128 192 256	128 192 256	Grover's algorithm	2,953 4,449 6,681	4.61×10^6 1.68×10^7 3.36×10^7	2.61×10^{12} years 1.97×10^{22} years 2.29×10^{32} years	
RSA[d]	Asymmetric encryption	1024 2048 4096	80 112 128	Shor's algorithm	2,050 4,098 8,194	8.05×10^6 8.56×10^6 1.12×10^7	3.58 hours 28.63 hours 229 hours	Move to NIST-selected PQC algorithm when available
ECC Discrete-log problem[e-g]	Asymmetric encryption	256 384 521	128 192 256	Shor's algorithm	2,330 3,484 4,719	8.56×10^6 9.05×10^6 1.13×10^6	10.5 hours 37.67 hours 55 hours	Move to NIST-selected PQC algorithm when available
SHA256[h]	Bitcoin mining	N/A	72	Grover's Algorithm	2,403	2.23×10^6	1.8×10^4 years	
PBKDF2 with 10,000 iterations[i]	Password hashing	N/A	66	Grover's algorithm	2,403	2.23×10^6	2.3×10^7 years	Move away from password-based authentication

Current state of PQC

- NIST has standardised 3 algorithms
    - with more almost there
- Govt are publishing their guidelines for migration
- Prototype libraries are becoming production grade

Can we migrate everything to PQC today?

Current state of PQC

- NIST has standardised 3 algorithms
  - with more almost there
- Govt are publishing their guidelines for migration
- Prototype libraries are becoming production grade

Can we migrate everything to PQC today? NO

- But we are close

Types of Cryptography

Traditional / Classical

- Prime number factorisation
- Discrete Log

PQC / Quantum safe

- Lattice based
- Code based
- Hash based
- …

PQC Algorithms

Lattice based

- ENCRYPT
  - FIPS 203, ML-KEM, Kyber
- SIGN
  - FIPS 204, ML-DSA, Dilithium
  - (DRAFT FIPS 206) FALCON

Hash based

- SIGN
  - FIPS 205, FN-DSA, SPHINCS+

Code based

- ENCRYPT
  - (Round 4) Classical McEliece
  - (Round 4) BIKE - Bit Flipping Key Encapsulation
  - (Round 4) HQC - Hamming Quasi-Cyclic

Hybrid PQC

- ML-KEM with ECC
- ML-KEM with RSA
- ML-DSA with RSA Sign / ECDSA
- FN-DSA with RSA Sign / ECDSA

Issues

- For packet size of 1500 bytes

Algorithms	PublicKey size	CipherText size	Fits in a packet?
RSA-2048	256 bytes	256 bytes	Yes
Ed25519	32 bytes	64 bytes	Yes
Kyber768	1184 bytes	1088 bytes	Yes
Dilithium2	1312 bytes	2420 bytes	No
Falcon-512	897 bytes	666 bytes	Yes
McEliece-8192	1357824 bytes	14120 bytes	No

Issues

- Many tools and APIs don't accept large keysizes
  - Kyber768 just about fits in a packet
- Lot of bandwidth overhead so networks can get clogged
- More computationally intensive
- Some algorithms require fast floating point arithmetic for good performance
- Even worse if you want a hybrid solution

Libraries for prototyping

- PQ Code Package (WIP: production grade) - Linux Foundation
- Liboqs
  - Wrapper for many different prototype algorithms
  - OpenSSL3 oqs-provider
- BouncyCastle
- Individual reference implementations from NIST submissions

How can you help?

- Find all the places we use asymmetric keys in Debian and slowly start thinking of the sequence of migration
- Think if we need hybrid solution (we probably do) or switching completely to PQC
- Think about various places using certificates
- Think about how we can preserve our web of trust or if we should start over from scratch

How can you help?

- Check if your favourite tools use PQC and test them out
- Update to protocols that support PQC based algorithms
  - TLS 1.3 can supports PQC!
  - OpenSSH 9.9 has hybrid support with `-oKexAlgorithms=mlkem768x25519-sha256`
  - liboqs oqs-provider has OpenSSL3 with PQC algorithms
- Sponsor opensource implementations in your favourite language for security audits
- Help fix bugs in cryto libraries (implementation bugs rather than cryptographic bugs)

References, citations and links

- National Academies of Sciences, Engineering, and Medicine. 2019. Quantum Computing: Progress and Prospects. Washington, DC: The National Academies Press. https://doi.org/10.17226/25196.
- FIPS 203: General encryption, ML-KEM, CRYSTALS-Kyber, Module Lattice KEM. https://doi.org/10.6028/NIST.FIPS.203
- FIPS 204: Digital signatures, ML-DSA, CRYSTALS-Dilithium, Module Lattice, DSA. https://doi.org/10.6028/NIST.FIPS.204
- FIPS 205: Digital signatures, SH-DSA, Sphincs+, Stateless Hash Based DSA. https://doi.org/10.6028/NIST.FIPS.205
- DRAFT FIPS 206: Digital signature, FN-DSA, FFT (fast-Fourier transform) over NTRU-Lattice-Based Digital Signature Algorithm. https://falcon-sign.info
- https://openquantumsafe.org/liboqs/
- https://github.com/open-quantum-safe/oqs-provider

Questions?

IRC: Count-Dracula
mail@kartikkulkarni.me