Post-Quantum Manifesto

Philippe Lamontagne

1. Public-key Cryptography Recap

2. How Quantum Mechanics Breaks Encryption

3. Post-Quantum Cryptography

4. Quantum Key Distribution

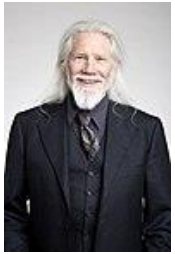
PUBLIC KEY CRYPTOGRAPHY

The Public Key Revolution (’70s)

New Directions in Cryptography

Invited Paper

WHITFIELD DIFFIE AND MARTIN E. HELLMAN, MEMBER, IEEE



A Method for Obtaining Digital Signatures and Public-Key Cryptosystems

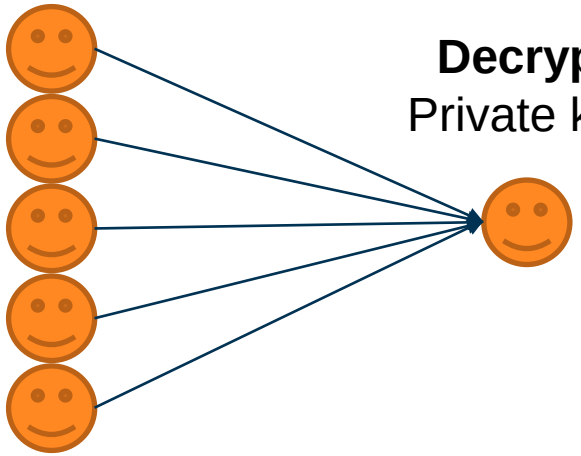
R.L. Rivest, A. Shamir, and L. Adleman*

Credits: Duncan Hull, Ajvol, Ronald L. Rivest, Leonard Adleman

Reminder

Encryption

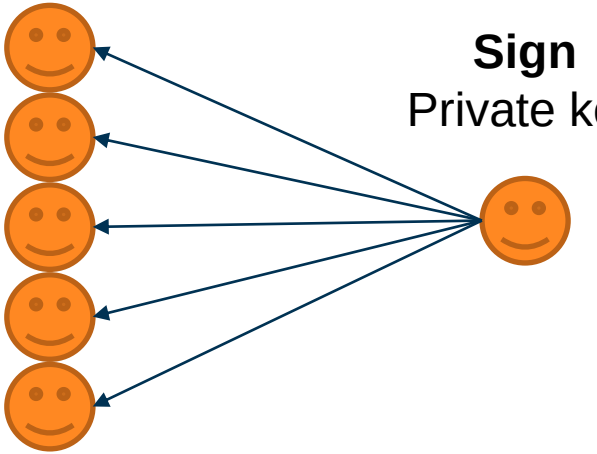
Encrypt
Public key



Decrypt
Private key

Digital Signatures

Verify
Public key



Sign
Private key

Public Key Cryptography

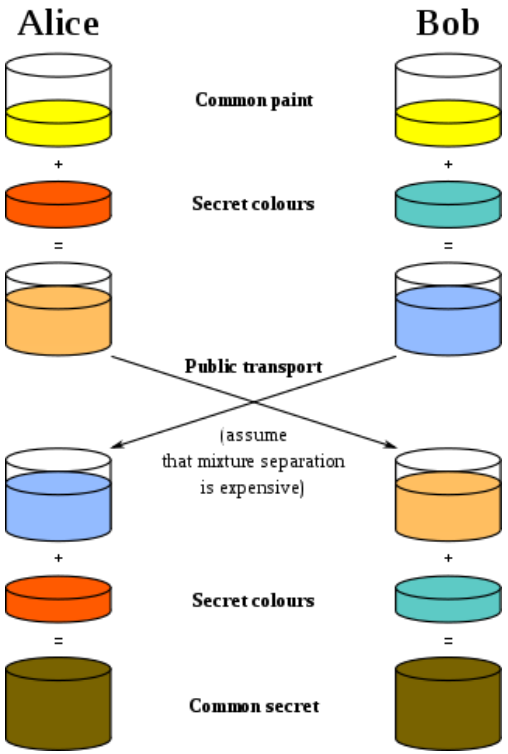
Many wonderful tools

- Public key encryption
- Key exchange
- Digital signatures
- Public key certificates

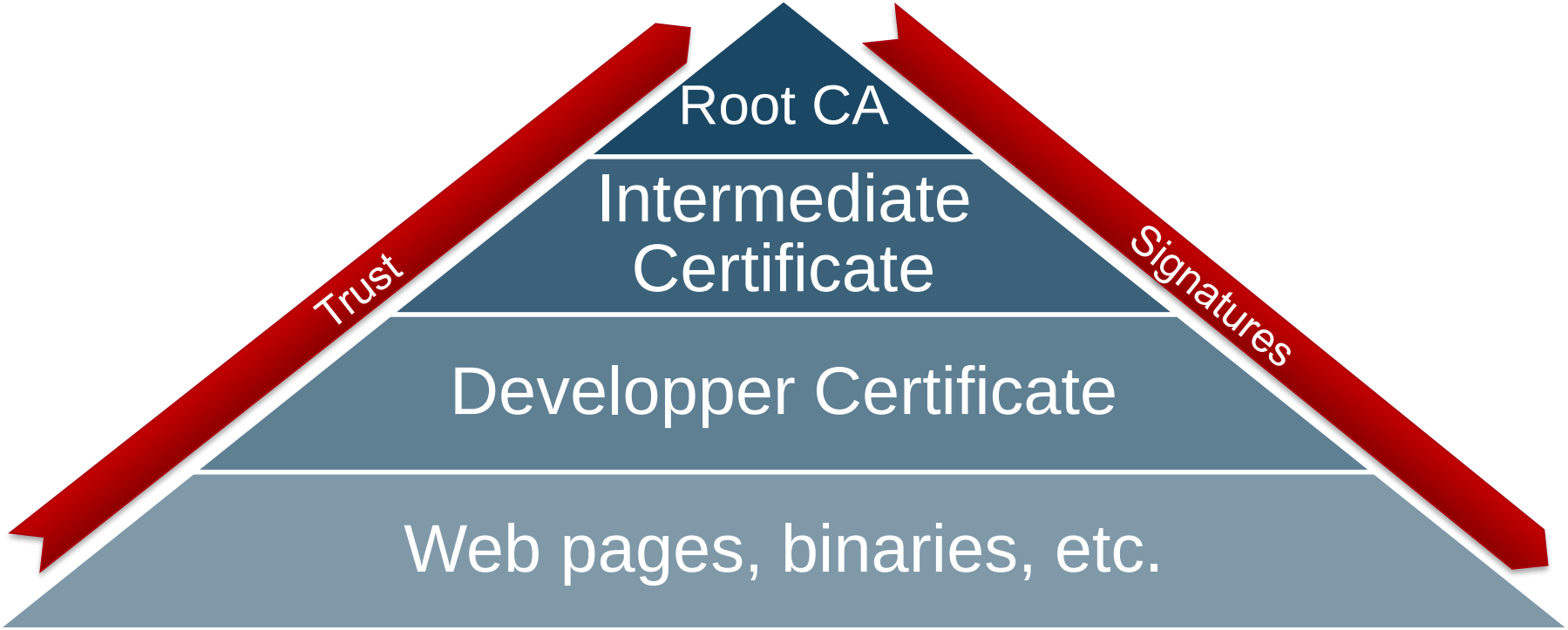
Assumes “one-way functions”

- Factoring (RSA) $N = p \cdot q$
- Discrete logarithm (DH, ECC)

$$x \quad such\ that \quad b^x \equiv a \pmod{n}$$



The Public Key Infrastructure



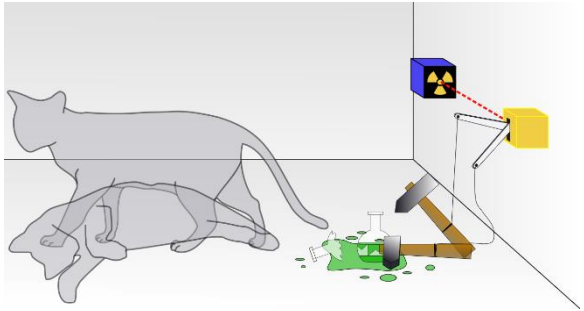
QUANTUM COMPUTING

Incomplete History of Quantum Mechanics

- ‘20s – ’30s: emerged as a new formalism of physics.
- 1935: Einstein dismisses QM (incomplete theory).
- 1964: John Stewart Bell “proves” Einstein wrong.
- 1983: Wiesner’s quantum money.
- 1984: Bennet and Brassard invent quantum key distribution.
- 1996: Peter Shor is factoring and discrete-logging on quantum computers.

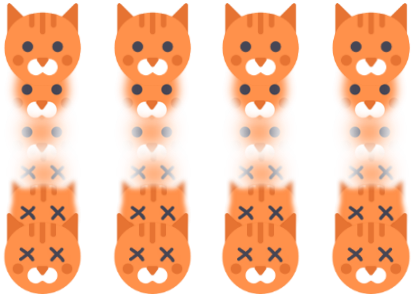
Quantum Computing: Superposition

“Schrödinger’s cat”



If  then  $= 42$

Collapse



$= 16$ cat combinations

Credits: Wikipedia user Dhatfield, tumblr user Igor Canova

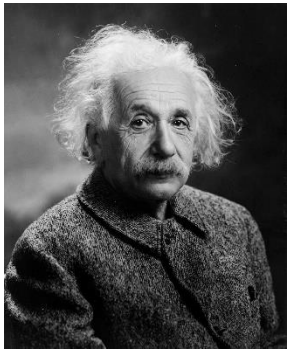
Quantum Computing: Entanglement

Teleporting an Unknown Quantum State via Dual Classical and Einstein-Podolsky-Rosen Channels

Charles H. Bennett,[(1)] Gilles Brassard,[(2)] Claude Crépeau,[(2),(3)]

Quantum Pseudo-Telepathy

Gilles Brassard,[1,*] Anne Broadbent,[1,†] and Alain Tapp[1,‡]



"Spooky action at a distance"

Shor's Algorithm

How Quantum Computers Break Encryption | Shor's Algorithm Explained

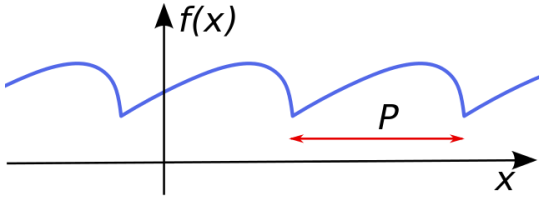
minutephysics • 1.2M views • 5 months ago

Go to http://www.dashlane.com/minutephysics to download Dashlane for free, and use offer code minutephysics for 10% off ...

CC

"To find a factor of an odd number n, given a method for computing the order r of x, choose a random $x \pmod n$, find its order r, and compute $\gcd(x^{r/2} - 1, n)$."

- Order: number r such that $x^r \equiv 1 \pmod n$



- Build superposition of:

$$|x^1 (\bmod n)\rangle + |x^2 (\bmod n)\rangle + \cdots + |x^q (\bmod n)\rangle$$

- Quantum Fourie

Polynomial

Exponential

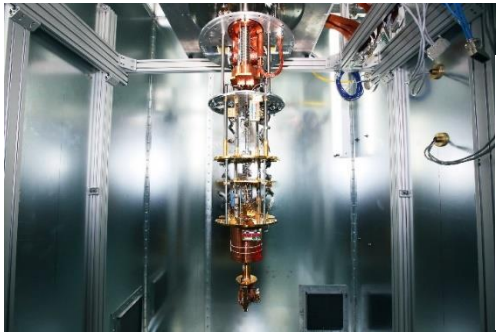
QFT: $O(\log^2 n)$

FFT: $O(n \cdot \log n)$

The Quantum Aftermath

Quantum Algorithm	Impacts
Shor (factoring)	• RSA broken
Shor (discrete logarithm)	• (EC)Diffie-Hellman broken • (EC)DSA broken
Grover (unstructured search) $O(N) \rightarrow O(\sqrt{N})$	• Symmetric Cryptography AES-128 -> AES-256 • Hash functions SHA-256 -> SHA-512

Why aren’t we panicking?



WIRED

TOM SIMONITE BUSINESS 09.24.2019 07:00 AM

Google’s ‘Quantum Suprem
Isn’t the End of Encryption

Google said its quantum computer outperformed conventional models. But it will still be years before you can use one for anything practical.

Decoherence & Noise

Yet we should do something about it <u>now</u>.

Time data needs to be secret

Time to transition to PQC

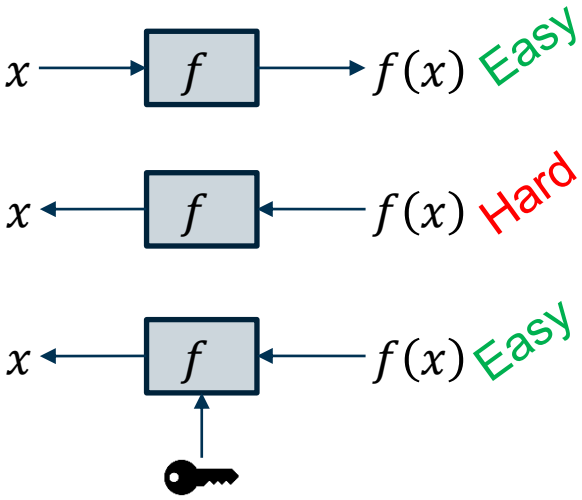
Time to large-scale quantum computer

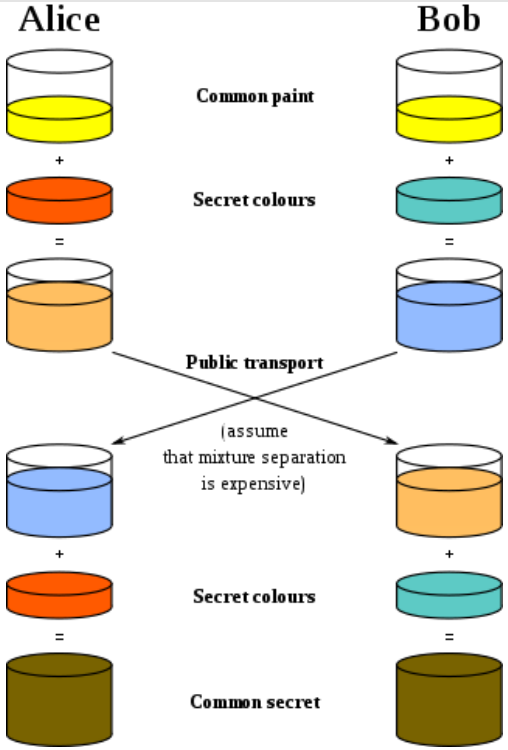
“Store now, decrypt later”

POST-QUANTUM CRYPTOGRAPHY

Finding New Hard Problems

Trapdoor permutation





Lattice-based Cryptography

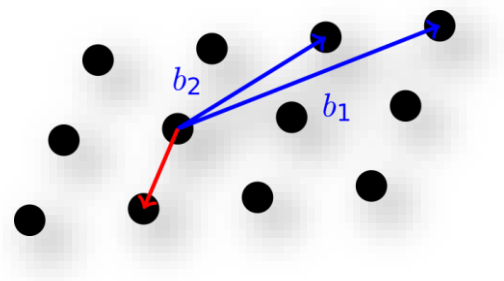
Key generation

- Private key is “good” basis of lattice
- Public key is “bad” basis

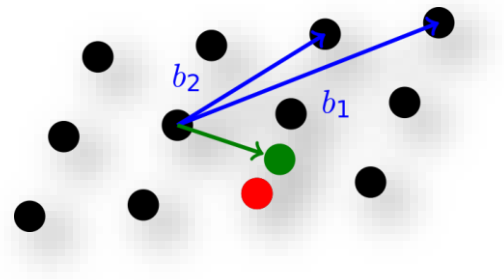
Encryption/Decryption

- Message is lattice point
- To encrypt, add noise vector to a properly chosen lattice point
- To decrypt, find the closest lattice point to noisy vector

Shortest Vector Problem



Closest Vector Problem



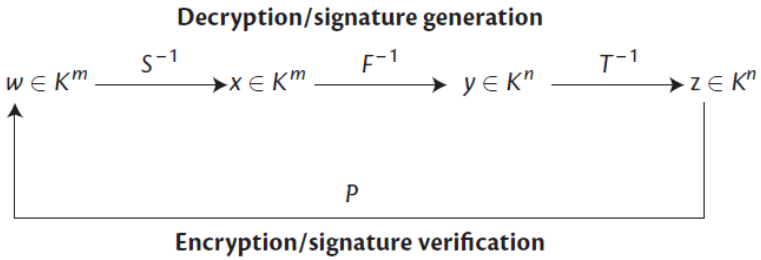
Multivariate Cryptography

Hardness of inverting systems of multivariate quadratic polynomials

1. Choose F easy to invert
2. Hide F with S and T easy to invert
3. Public key P is hard to invert, but easy knowing private key
   (S, F, T).

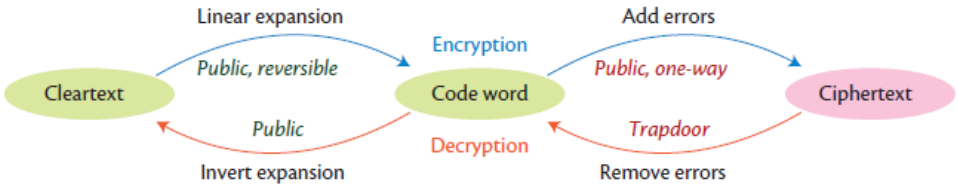
$$F(x, y) = 2xy + 4x^2 + 2y + 7$$

$$P = S \circ F \circ T =$$



Code-based Cryptography

- Hardness of decoding a random linear error correcting code
- Private key is an efficient decoder for an ECC
- Public key is “random” generator matrix for the same ECC
- Encrypt by adding errors. Decrypt by correcting.

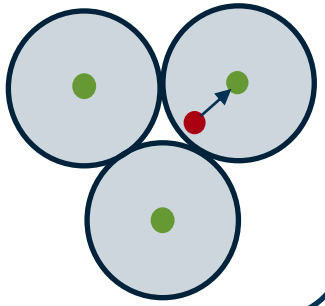


Error Correcting Code

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

$$w = 011 \mapsto c = 01110$$

Decoding



Hash-based Signatures

- Lamport one-time signatures
- Merkle signature scheme
  - Stateful
  - 2^N one-time signatures with a single public key
- PRNG for one-time key generation

sign $m = 101$:

$$K_{priv} = \begin{pmatrix} x_{0,1} & x_{0,2} & x_{0,3} \\ x_{1,1} & x_{1,2} & x_{1,3} \end{pmatrix}$$

$$\implies \sigma = (x_{1,1}, x_{0,2}, x_{1,3})$$

verify signature $\sigma = (x_1, x_2, x_3)$ for $m = 101$:

$$K_{pub} = \begin{pmatrix} y_{1,0} & y_{2,0} & y_{3,0} \\ y_{1,1} & y_{2,1} & y_{3,1} \end{pmatrix}$$

$$\implies \begin{aligned} H(x_1) &= y_{1,1} \\ H(x_2) &= y_{2,0} \\ H(x_3) &= y_{3,1} \end{aligned}$$

ALGORITHMS AND WHERE TO FIND THEM

PQC Standardisation (NIST)

- 2016 • Launched at PQCrypto 2016
- 2017 • Round 1 (69 accepted submissions)
- Now • Round 2 (26 algorithms)
- 2020/2021 • Round 3 begins
- 2022/2024 • Draft standards

Type	PKE/KEM	Signature
Lattice	• CRYSTALS-KYBER[4] • FrodoKEM[5] • LAC • NewHope[6] • NTRU (merger of NTRUEncrypt and NTRU-HRSS-KEM)[7] • NTRU Prime[8] • Round5 (merger of Round2 and Hila5, announced 4 August 2018)[9] • SABER[10] • Three Bears[11]	• CRYSTALS-DILITHIUM[4] • FALCON[12] • qTESLA[13]
Code-based	• BIKE[14] • Classic McEliece • HQC[15] • LEDAcrypt (merger of LEDAkem[16] and LEDApkc[17]) • NTS-KEM[18] • ROLLO (merger of Ouroboros-R, LAKE and LOCKER) [19] • RQC[20]	
Hash-based		• SPHINCS+[21]
Multivariate		• GeMSS[22] • LUOV • MQDSS[23] • Rainbow
Supersingular Elliptic Curve Isogeny	• SIKE[24]	
Zero-knowledge proofs		• Picnic[25]

Efficiency

- **Larger key sizes**
- **Larger signatures**
- **More complex operations***

Algorithm	Type	Public Key	Private Key	Signature
NTRU Encrypt[36]	Lattice	6130 B	6743 B	
Streamlined NTRU Prime	Lattice	1232 B		
Rainbow[37]	Multivariate	124 KB	95 KB	
SPHINCS[19]	Hash Signature	1 KB	1 KB	41 KB
SPHINCS+[38]	Hash Signature	32 B	64 B	8 KB
BLISS-II	Lattice	7 KB	2 KB	5 KB
GLP-Variant GLYPH Signature[10][39]	Ring-LWE	2 KB	0.4 KB	1.8 KB
New Hope[40]	Ring-LWE	2 KB	2 KB	
Goppa-based McEliece[14]	Code-based	1 MB	11.5 KB	
Random Linear Code based encryption[41]	RLCE	115 KB	3 KB	
Quasi-cyclic MDPC-based McEliece[42]	Code-based	1232 B	2464 B	
SIDH[43]	Isogeny	751 B	48 B	
SIDH (compressed keys)[44]	Isogeny	564 B	48 B	
3072-bit Discrete Log	**not PQC**	384 B	32 B	96 B
256-bit Elliptic Curve	**not PQC**	32 B	32 B	65 B

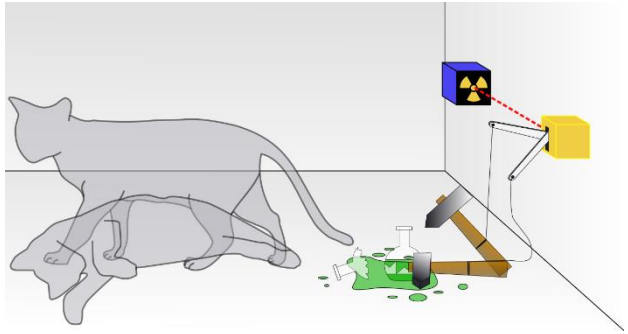
Where to get implementations

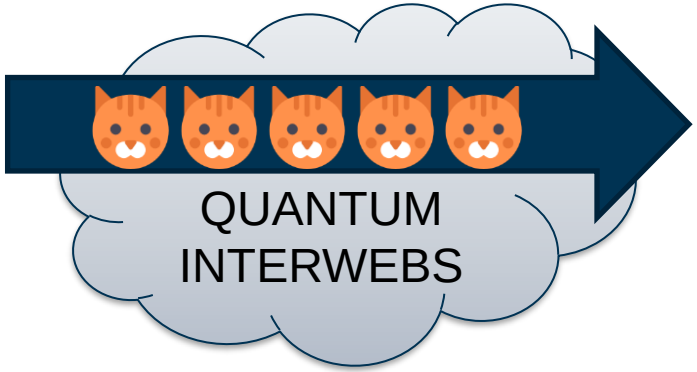
OPEN QUANTUM SAFE https://openquantumsafe.org/

- **liboqs: open-source C implementation**
  - Wrappers for C#, C++, Python, Go
- **Common API for key encapsulation and signatures**
- **OpenSSL and OpenSSH forks**
- **Uses of liboqs:**
  - Microsoft OpenVPN fork, Utimaco HSM, Mullvad VPN client

QUANTUM KEY DISTRIBUTION

Quantum Key Distribution





QKD	PQC
p2p fiber optic & single-photon	Existing infrastructure
Unconditional security	Hard math problems
Authenticated classical channel	Public key distribution

RECAP

- Public key crypto based on hard math problems.
- Easy for quantum computers, but they don’t exist just yet.
- Still important to think of replacing RSA and DH.
- Lots of quantum-resistant alternatives (standards soon).
- QKD also exists.

THANK YOU

Philippe Lamontagne • Research Officer
Philippe.Lamontagne2@cnrc-nrc.gc.ca