Post-Quantum Crypto
Today's defense against tomorrow's quantum hacker gnomes

Christian Paquin

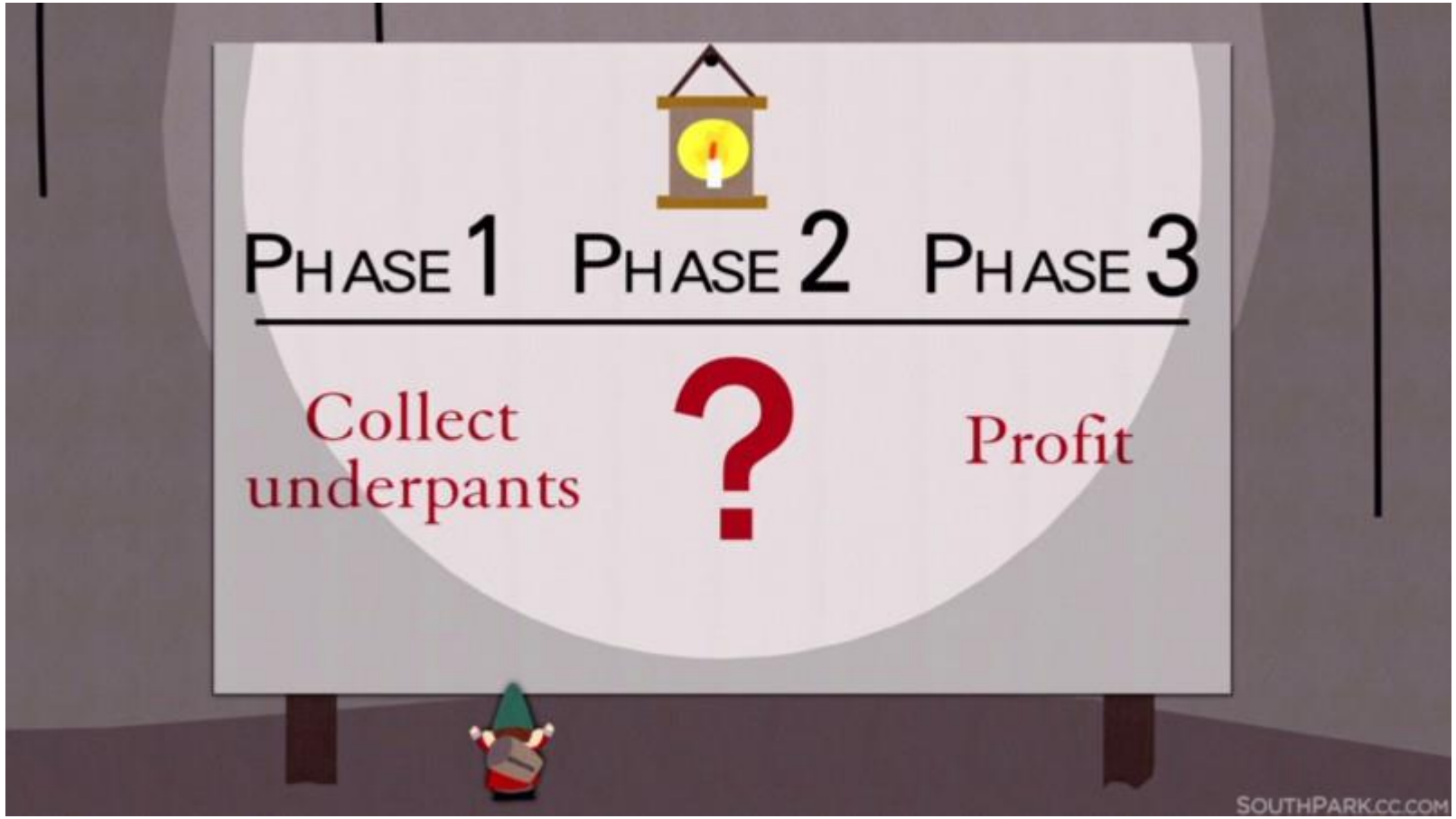
@chpaquin

Principal Program Manager

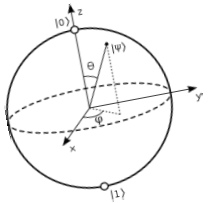
Microsoft Research

nsec 2019

Turns out the underpants business isn't great



The Quantum Revolution



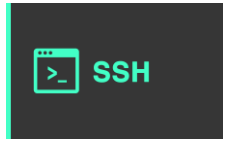
$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$

- Quantum computers use the properties of quantum mechanics (entanglement, superposition) to implement algorithms not possible on classical computers
  - Nice math on paper, hard to build in practice
  - Years away, but…
- A lot of investment$
  - All around the world
  - My colleagues are building the full stack: from the chip to the SDK
    https://www.microsoft.com/quantum/

QDK

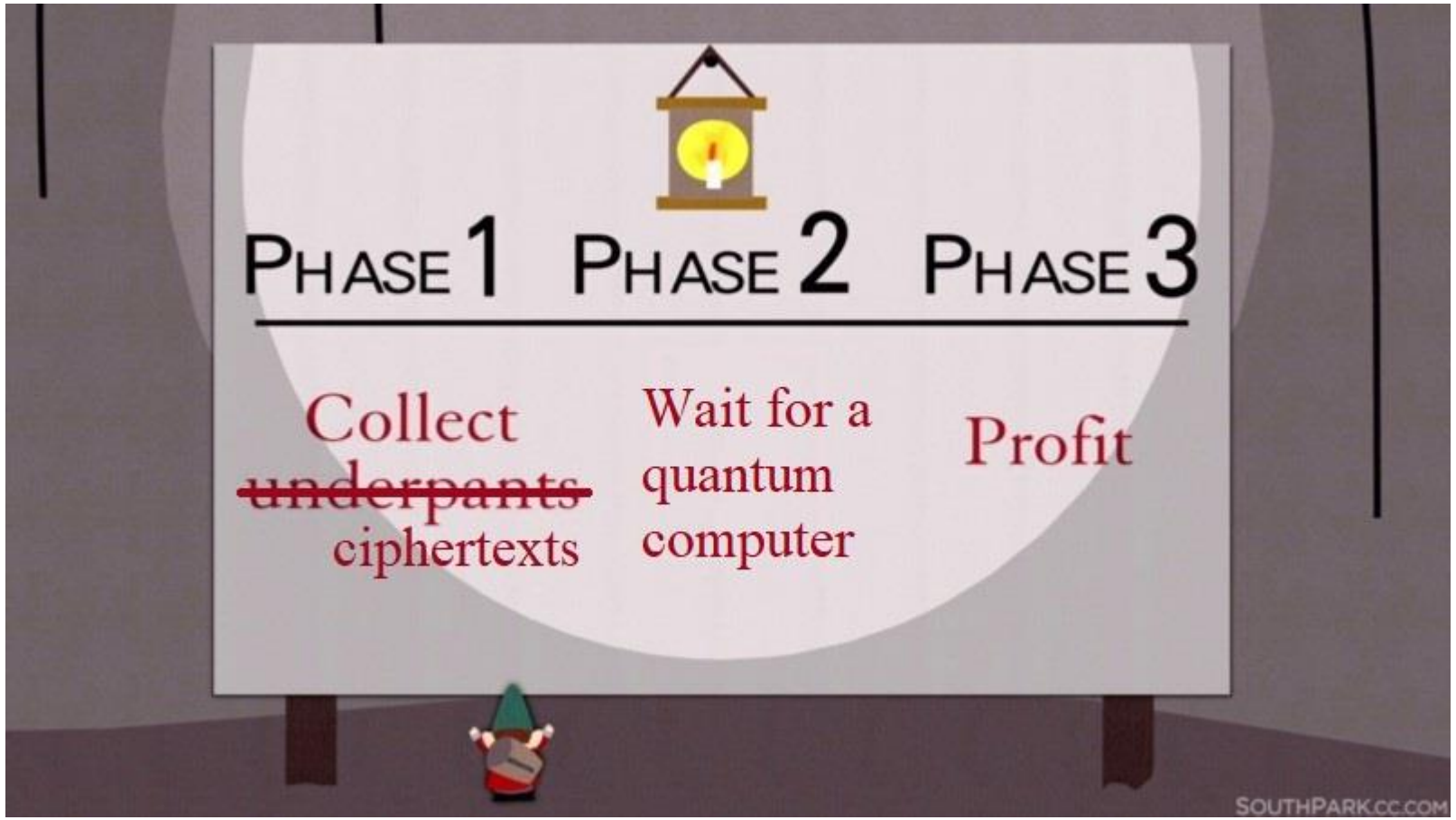
The Quantum Menace

- Quantum is great for computing, but bad news for cryptography!
  - Shor (1994) solves the factoring (breaks RSA) and discreet log (breaks DSA, Diffie Hellman, and elliptic curve variants) problems in polynomial time
  - Grover (1996) speeds up function inversion; need to double the size of hash functions (SHA) and block ciphers (AES)
- Breaks ~~most~~ all the asymmetric crypto in use today




Hacker gnomes have a new business model



Tic toc…

Oh dear!
Oh dear!
I shall be
too late!

- Michele Mosca (Waterloo):

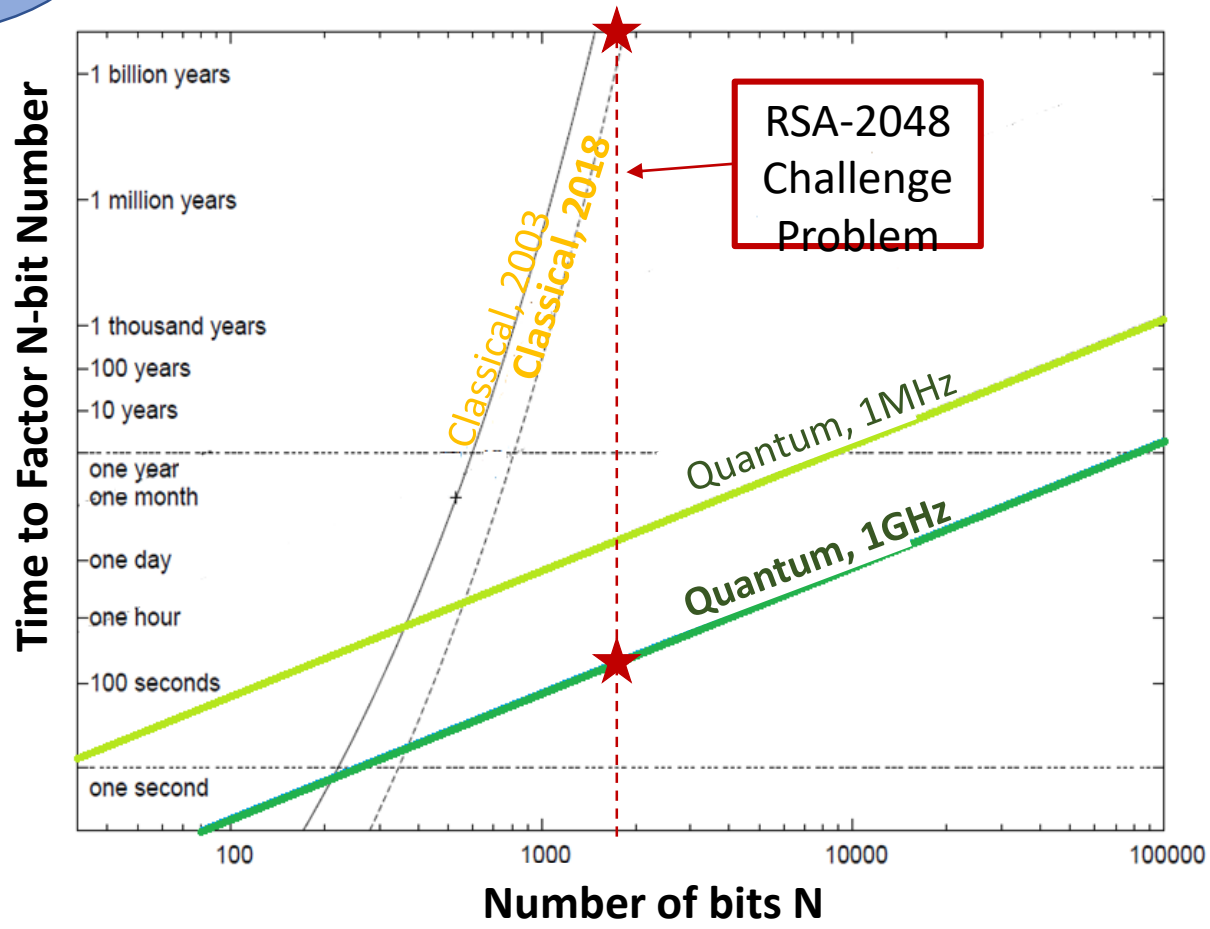
  "1/7 chance of breaking RSA-2048 by 2026, 1/2 chance by 2031" (2015)

  "1/6 chance within 10 years" (2017)

- Simon Benjamin (Oxford):

  "maybe 6-12 years if someone is willing to go Manhattan project"

- My colleagues estimate 2030



We need *quantum-safe* alternatives soon: *post-quantum cryptography!*

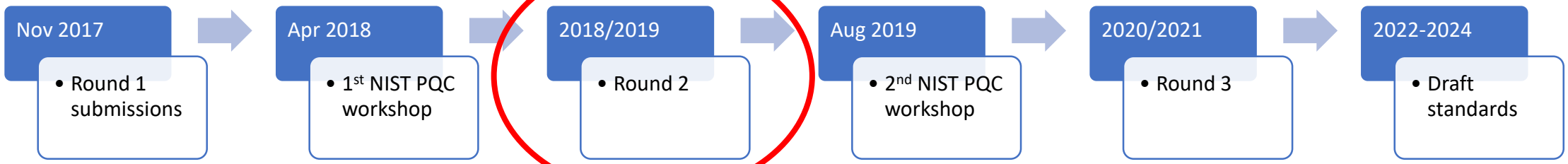
Why PQC today?

- Capture now, decrypt later
- Updating standards is looooooong
  - TLS, SSH, IKE, PKI, S/MIME, …
- Unknown impact on code base
  - Longer key/message/sig sizes
  - Slower running times
  - Code agility

Do your apps protect data that needs to be kept secret for more than 10 years?

NIST competition

- The National Institute of Standards and Technologies (NIST) started the process to standardize Post-Quantum Cryptography



- Looking for signatures, encryption, and key establishments schemes
  - Five levels, corresponding to breaking AES-128/192/256 and SHA-256/384
- 26 submissions remaining in round 2 (from 69 in round 1)
  - 9 signature schemes, 17 KEM/encryption schemes
- https://csrc.nist.gov/projects/post-quantum-cryptography

Many new proposals

- From various math families
  - Lattices, error-correcting codes, multivariate systems, hash functions, isogenies, zero-knowledge proofs
- Our submissions
  - FRODO (KEM)
    - Learning With Error problem
    - https://frodokem.org/
  - SIKE (KEM)
    - Supersingular Isogeny elliptic curves
    - https://sike.org/
  - Picnic (sig)
    - Zero-knowledge proofs, hash, and block ciphers
    - https://microsoft.github.io/Picnic/
  - qTesla (sig)
    - Ring Learning with Error problem
    - https://qtesla.org

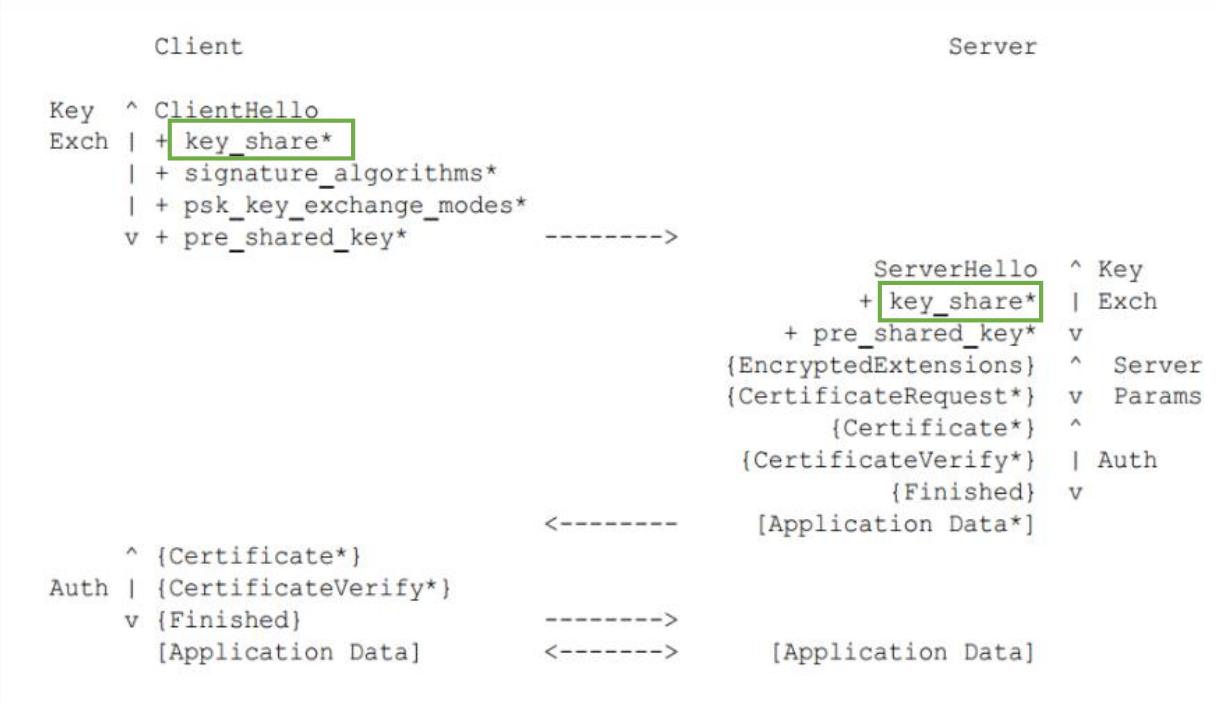
I can’t deal with all that PQC!?!

OPEN QUANTUM SAFE

- C library created to simplify integration of PQC into applications
- Multi-org dev team

- Master branch (for integration) and NIST branch (for experimentation)
- Shipped integrations with OpenSSL (TLS 1.2, 1.3) and OpenSSH
- Language wrappers (C++, C#, Python, upcoming: Java)
- https://openquantumsafe.org/

TLS integration



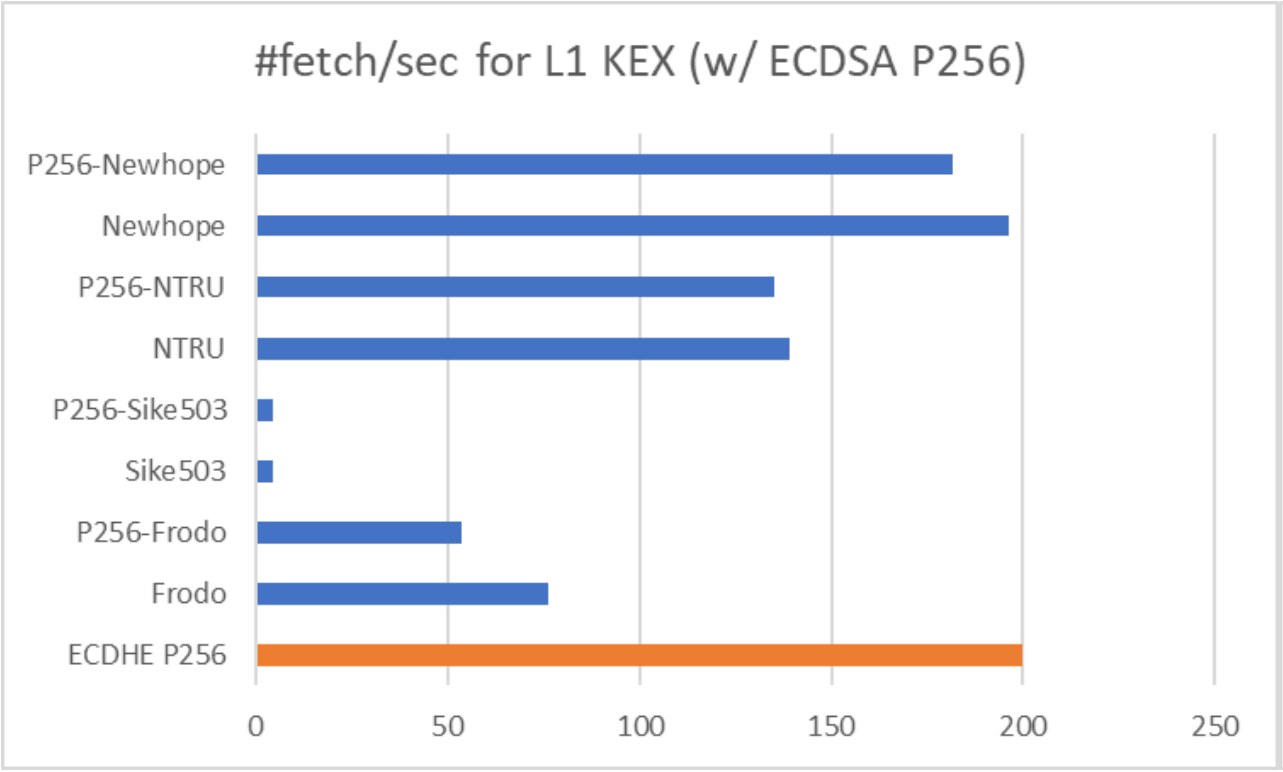
- Integrated OQS in OpenSSL 1.0.2 (TLS 1.2) and 1.1.1 (TLS 1.3)
- Added PQC key exchange (KEX) and authentication (1.1.1 only)
- Supports hybrid mode
  - Combines classical and PQC (today's security + quantum-proof)
- Tested with apache and nginx
- https://github.com/open-quantum-safe/openssl/wiki/PQC-integration-into-TLS-1.3

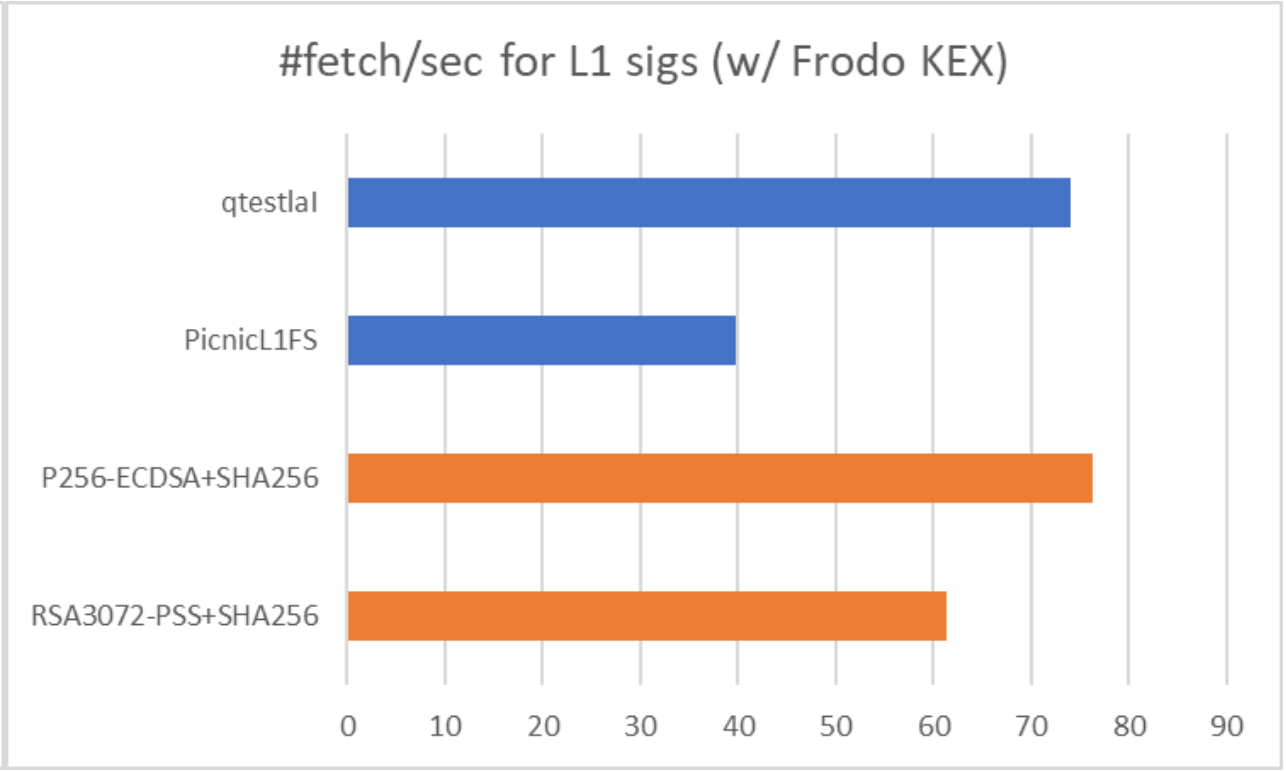
Hybrid scenarios

- TLS 1.3 KEX, two approaches
  - Naïve: define combo schemes and concatenate the data (currently implemented)
  - Multiple key shares (classical and PQC) both updating the master secret
    - State machine already supports hybrid keys, for PSK + ECDHE
    - PQC proposals: draft-whyte-qsh-tls13-06, draft-schanck-tls-additional-keyshare-00
- PKI, need to convey a classical and PQC signature
  - Hybrid signature scheme (currently implemented)
  - Convey two certs
  - TLS PQC cert extension
  - X.509 extension for an extra PQC key
  - Bindel, Herath, McKague, Stebila; Transitioning to QR PKI

TLS 1.3 Perf

Measurements with client/server on localhost (no network delay)



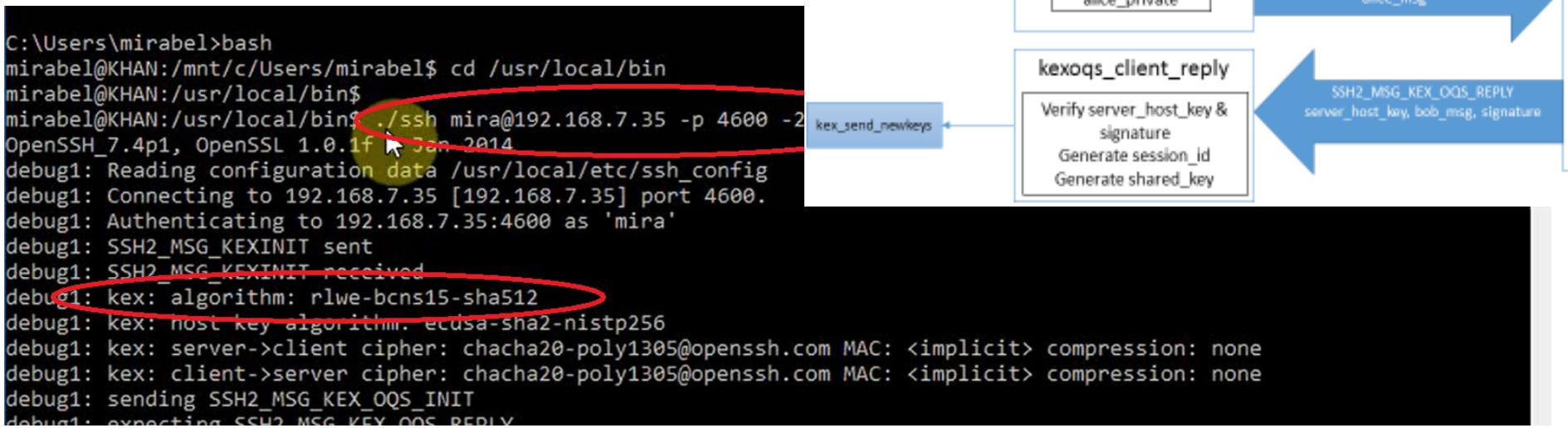


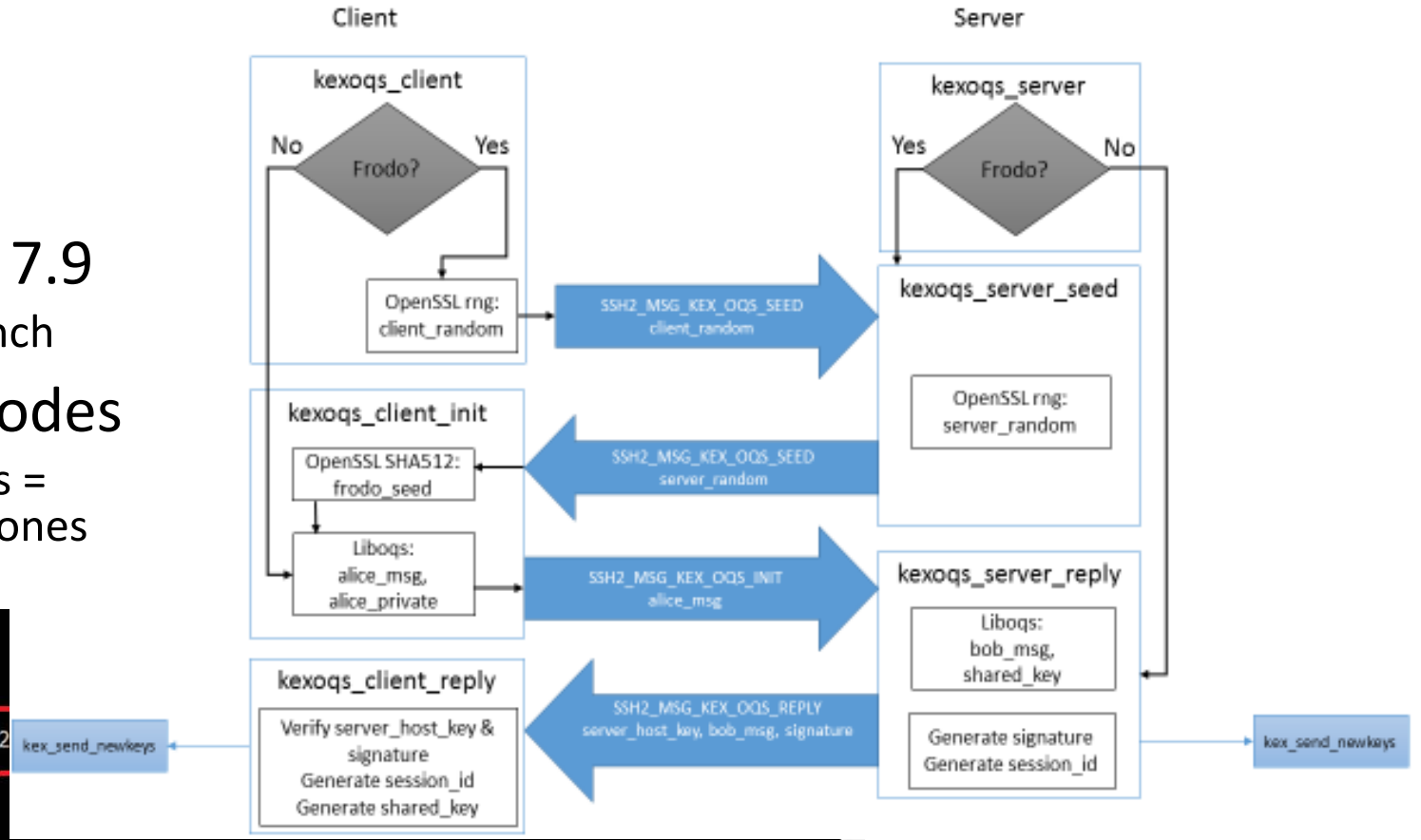
Classical PQC

July 15th built of OQS/OpenSSL
Azure Standard D4s v3 VM, Ubuntu OS

SSH integration

- Integrated OQS in OpenSSH 7.9
  - KEX and sig algs from master branch
- Supports PQC and hybrid modes
  - Shared secret and auth signatures = concatenation of classical & PQC ones

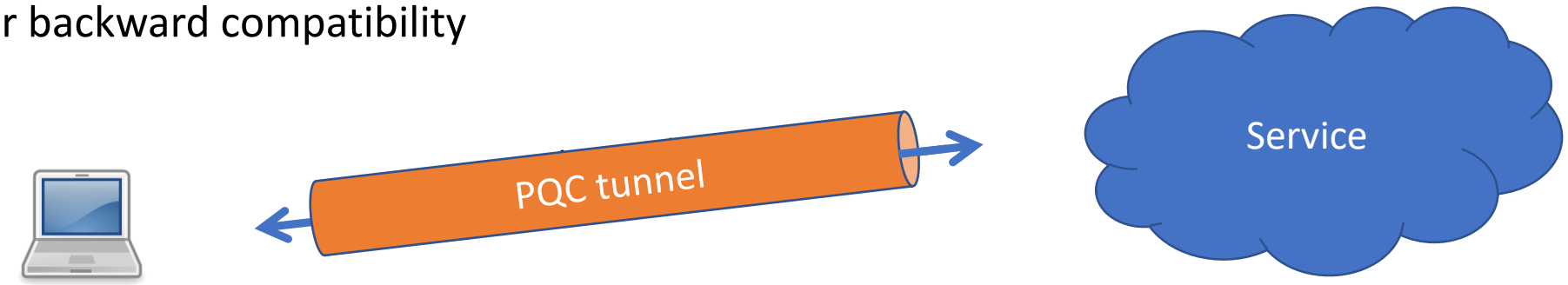




https://github.com/open-quantum-safe/openssh-portable

OpenVPN

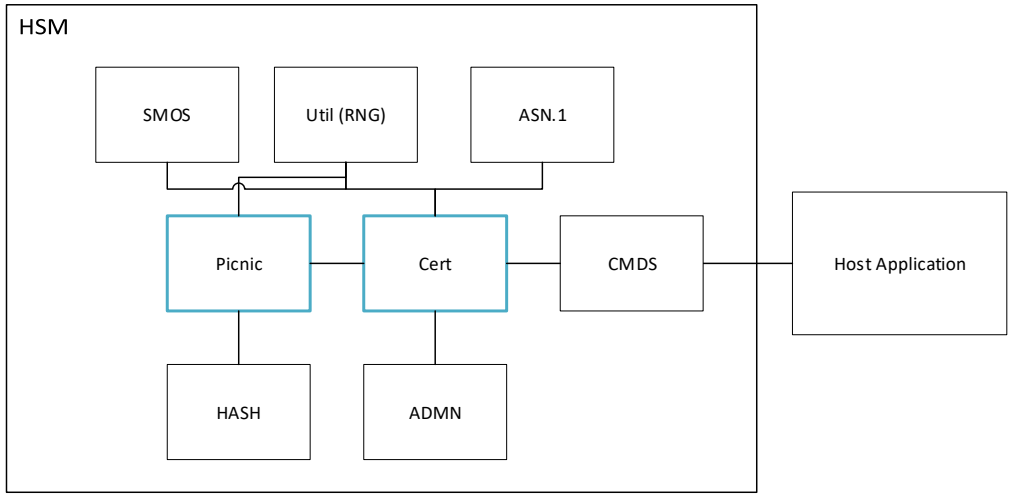
- Integration in OpenVPN 2.4.4
  - Uses OQS-OpenSSL to protect TLS key establishment
  - Uses RSA or Picnic auth
- Easy way to achieve PQC tunnel to the cloud even if applications haven't been updated
  - Good for backward compatibility



- Tested with Raspberry Pi and Windows clients, and Azure Linux VM service
- https://github.com/Microsoft/PQCrypto-VPN

HSM integration

- Integrated Picnic into an Utimaco HSM (Security Server Se50 LAN v4)
- Experiment consisted of
  1. Picnic key generation and signing in HSM (using reference implementation)
  2. Generated self-signed root Picnic cert
  3. Issued end-user RSA certs using the Picnic cert
- https://microsoft.github.io/Picnic/



Demo

TLS 1.3 – OpenSSL 1.1.1

- KEX: ECDHE P256 + Frodo 640
- Auth: ECDSA P256 + qTESLA I

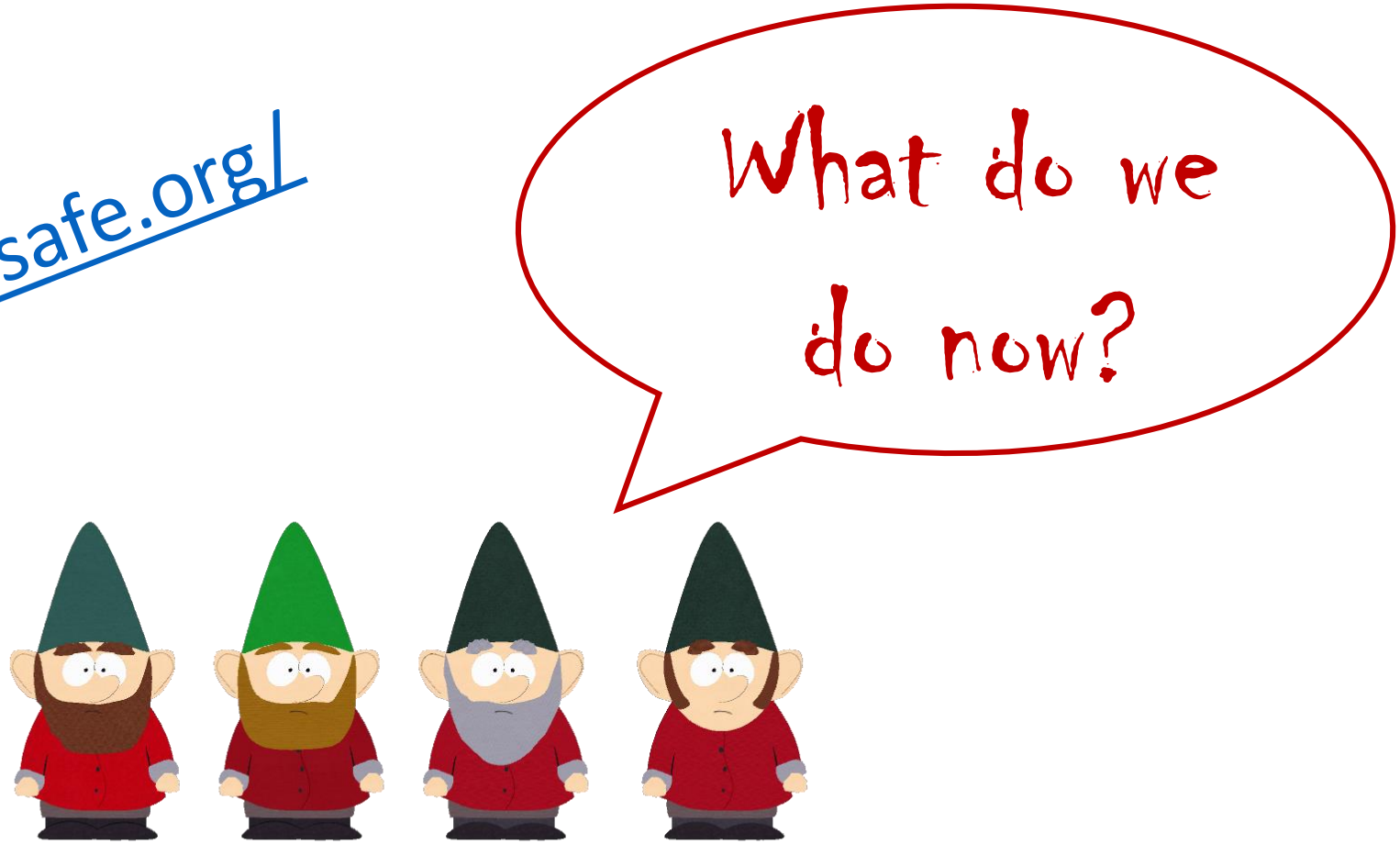
SSH2 – OpenSSH 7.9

- KEX: ECDHE P384 + SIKE 503
- Auth: ECDSA P256 + Picnic L1FS

The road ahead

- Start planning transition to PQC
- Make sure your apps/services are crypto agile
- Consider deploying hybrid solutions for long-lived, high-value data
- Consider wrapping long-tail apps/services in a PQC-VPN tunnel

https://openquantumsafe.org/



cpaquin@microsoft.com

@chpaquin