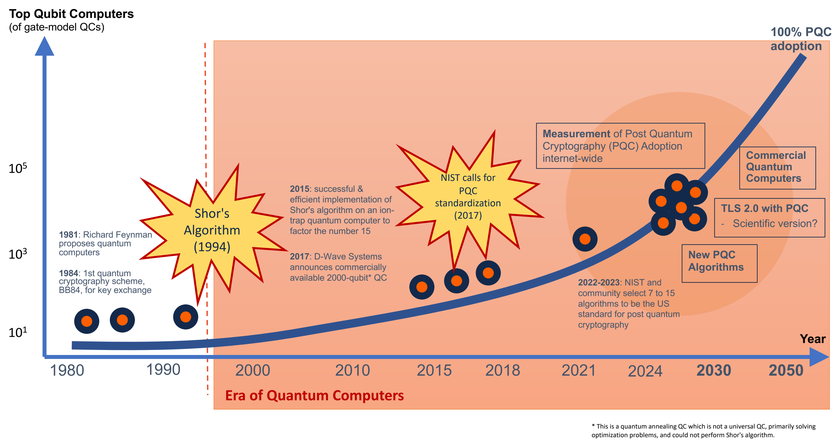
Post-Quantum Cryptography (PQC) Adoption Measured at the National Center for Supercomputing Applications (NCSA)

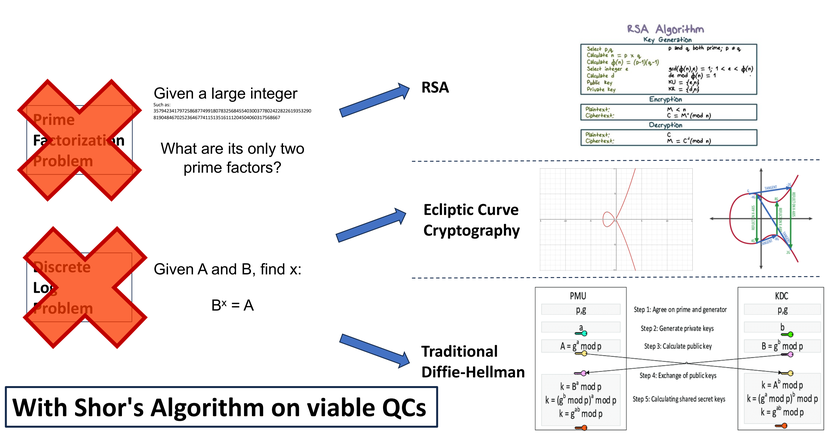
Jakub Sowa[1], Bach Hoang[2], Phuong Cao[1]

[1]National Center for Supercomputing Applications, UIUC, [2]Department of Mathematics, UIUC

Quantum computing opens new challenges for cryptography



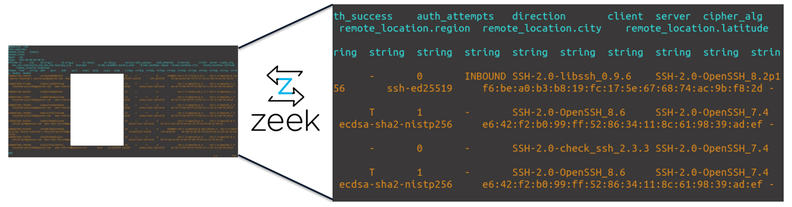
Quantum algorithms can break traditional encryptions



With Shor's Algorithm on viable QCs

Data is gathered from NCSA network metadata

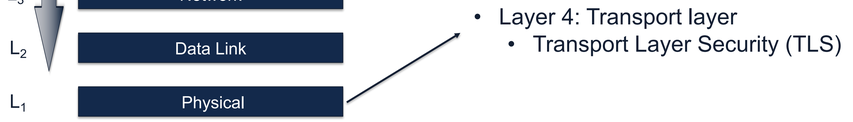
- Sampled a few hours of network metadata generated by Zeek at NCSA
- No information beyond metadata was used
- Zeek logs were parsed in Python for analysis of network traffic of certain protocols



PQC in RDP and DNS is nearly unused

- Remote Desktop Protocol (RDP):
  - Can be configured to use TLS encryption and Network-layer authentication but only on Windows 11
  - Out of 26 connections in sample data, only 2 used both encryption and authentication
- Domain Name System (DNS):
  - Not encrypted at all by default -- anyone can see what websites you try to visit, even on the NCSA network
    - Can enable HTTPS encryption on some browsers (Firefox, Chrome etc.)
    - Can also configure DNS to encrypt DNS-over-TLS (DoT)

PQC observatory analytics workflow



- Layer 7: Application layer
  - Remote Desktop Protocol (RDP)
  - Domain Name System (DNS)
  - Secure Shell (SSH)
- Layer 4: Transport layer
  - Transport Layer Security (TLS)

Potential Solutions

- More generally, make sure to keep software like SSH protocols and browsers updated to use the safest cryptography
- Potentially configure most network protocols to run over TLS, once TLS has PQC
  - A **TLS Termination Proxy** can be used as a wrapper around current infrastructure to make it easier to secure traffic
  - Streamlining and simplifying cryptography and security
- A TLS v2.0, introducing PQC by default
  - Securing all network traffic even against quantum adversaries

Minimal of PQC present in Secure Shell

Encryption Algorithm	Occurrences
aes256-gcm@openssh.com	1686 (66.93%)
aes128-ctr	454 (18.02%)
chacha20-poly1305@openssh.com	188 (7.46%)
aes128-gcm@openssh.com	156 (6.19%)
aes256-ctr	31 (1.23%)
aes128-cbc	2 (0.08%)
3des-cbc* ✖	1 (0.04%)

MAC Algorithm	Occurrences
hmac-sha2-256-etm@openssh.com	1844 (73.20%)
hmac-sha2-256	457 (18.14%)
umac-128-etm@openssh.com	154 (6.11%)
umac-64-etm@openssh.com	33 (1.31%)
hmac-sha1	17 (0.67%)
hmac-sha2-512	13 (0.52%)

Host Key Algorithm	Occurrences
ecdsa-sha2-nistp256	1275 (50.62%)
ssh-ed25519	1233 (48.95%)
ssh-rsa* ✖	5 (0.20%)
rsa-sha2-512	4 (0.16%)

Key Exchange Algorithm	Occurrences
curve25519-sha256	2030 (80.59%)
curve25519-sha256@libssh.org	473 (18.78%)
diffie-hellman-group-exchange-sha256	6 (0.24%)
diffie-hellman-group1-sha1** ✖	5 (0.20%)
sntrup761x25519-sha512@openssh.com* ★	2 (0.08%)
diffie-hellman-group14-sha1	2 (0.08%)

★ = post-quantum ✖ = not secure even now

- 99.92% of all SSH traffic was **not secure** against quantum adversaries
- **sntrup761x25519**: Streamlined NTRU Prime
  - A hybrid classical-PQ key exchange algorithm available by default in OpenSSH v9.0 and above as of **2022**
- Over 83% of server-side SSH protocol versions were from **2019 and earlier**

Key Exchange Algorithm	Occurrences
curve25519-sha256	2030 (80.59%)
curve25519-sha256@libssh.org	473 (18.78%)
diffie-hellman-group-exchange-sha256	6 (0.24%)
diffie-hellman-group1-sha1** ✖	5 (0.20%)
sntrup761x25519-sha512@openssh.com* ★	2 (0.08%)
diffie-hellman-group14-sha1	2 (0.08%)

PQC used in Transport Layer Security (TLS) is limited

- About 65% of connections were using TLSv1.3; about 35% were TLSv1.2
- Many unsecure cipher suites were in use – 4 had over 1000 connections!
- No standard version of TLS has any PQC -- none at NCSA either
- Many designs in the works by the IETF & NIST; some companies even trying to integrate PQC into their TLS
- The difficulty to even adopt TLS v1.3 internet-wide foreshadows PQC adoption as well

TLS Ciphersuites	Occurrences
TLS-AES-128-GCM-SHA256*	416447 (53.02%)
TLS-ECDHE-RSA-WITH-AES-256-GCM-SHA384	117788 (15.00%)
TLS-AES-256-GCM-SHA384*	100708 (12.82%)
TLS-ECDHE-RSA-WITH-AES-128-GCM-SHA256	79171 (10.08%)
TLS-DH-ANON-WITH-AES-256-GCM-SHA384**	42261 (5.38%)
TLS-ECDH-ANON-WITH-AES-256-CBC-SHA**	14787 (1.88%)
TLS-ECDHE-RSA-WITH-NULL-SHA**	5612 (0.71%)
TLS-ECDHE-ECDSA-WITH-AES-128-GCM-SHA256	3382 (0.43%)
TLS-ECDHE-RSA-WITH-CHACHA20-POLY1305-SHA256	2787 (0.35%)
TLS-ECDHE-ECDSA-WITH-AES-256-GCM-SHA384	2497 (0.32%)

Table 2: A list of the top 10 cipher suites found in sample TLS connection data (*in TLSv1.3, **considered non-secure).

Future Work

- Creating a web tool that measures Post-Quantum Cryptography at NCSA and other organizations: "Network of PQC telescopes"
- Creating a tool to quickly scan a network and analyze its usage of PQC
- Analyzing the risk of and figuring how to mitigate Post-Quantum "cipher suite downgrade attacks"

ACKNOWLEDGEMENTS NSF, CSL, NCSA Students Pushing Innovation (SPIN) Program, NCSA ICI/CSD