NDSS 2020,
February 26, 2020

Post-Quantum Authentication in TLS 1.3: A Performance Study

Dimitrios Sikeridis[1,2], Panos Kampanakis[2], Michael Devetsikiotis[1]

[1]Dept. of Electrical and Computer Engineering, The University of New Mexico, USA

[2]Security & Trust Organization, Cisco Systems, USA

Quantum Computing

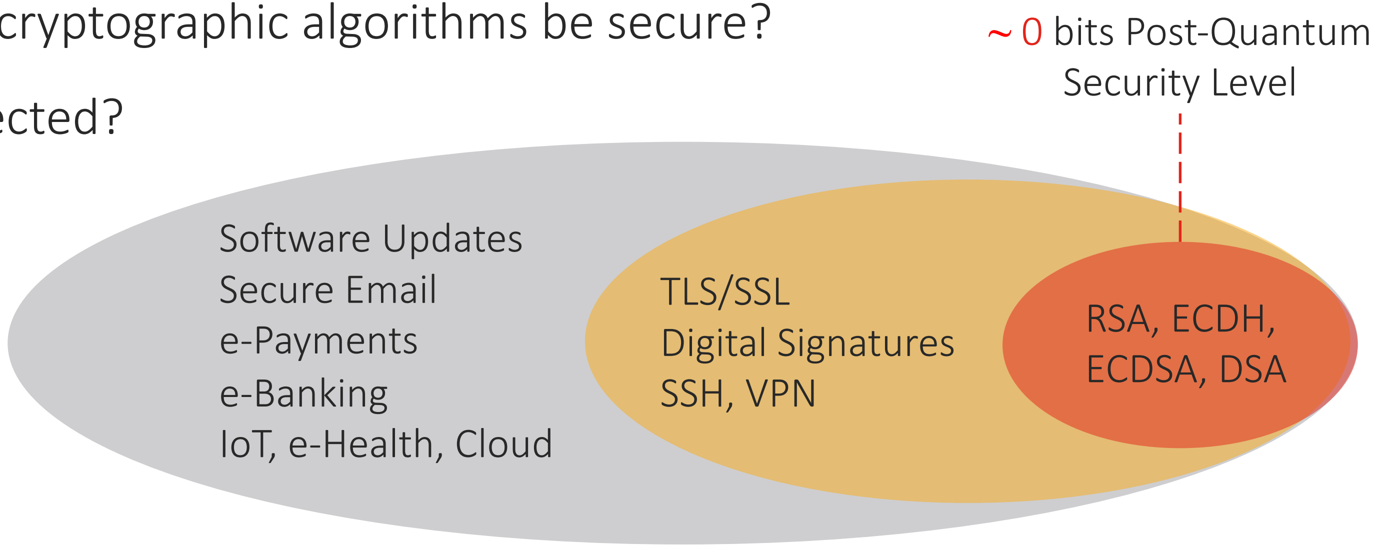
- Practical Quantum Computing existence/timeline is still debatable[1]
- QC research funding is increasing
- IBM has multiple small-scale prototypes
- Google's quantum supremacy claim

IBM's Quantum Computer

[1]Dyakonov, Mikhail. "When will useful quantum computers be constructed? Not in the foreseeable future, this physicist argues. Here's why: The case against: Quantum computing." *IEEE Spectrum* 56.3 (2019): 24-29

Quantum Computing – Practical impact?

- A large scale QC will be able to solve Integer Factorization and Discrete Logarithm Problems[1]
- Will our current cryptographic algorithms be secure?
- What will be affected?



[1]Shor, Peter W. "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer." *SIAM review* 41.2 (1999): 303-332

NIST Post-Quantum Project

- PQ Algorithm Standardization
- Currently in Round 2
- 9 PQ Digital Signature Algorithms
- 17 PQ Key Exchange Algorithms

NIST

Information Technology Laboratory

COMPUTER SECURITY RESOURCE CENTER

PROJECTS | POST-QUANTUM CRYPTOGRAPHY

Post-Quantum Cryptography

Round 2 Submissions

Official comments on the Second Round Candidate Algorithms should be submitted using the "Submit Comment" link for the appropriate algorithm. Comments from the pqc-forum Google group subscribers will also be forwarded to the pqc-forum Google group list. We will periodically post and update the comments received to the appropriate algorithm.

All relevant comments will be posted in their entirety and should not include PII information in the body of the email message.

Please refrain from using OFFICIAL COMMENT to ask administrative questions, which should be sent to pqc-comments@nist.gov

Post-Quantum Transport Layer Security (TLS) Status

- No complete solution yet
  - Google, Cloudflare[1], Microsoft, and Amazon have been looking into PQ Key Exchange

- This work:
  - Focuses on **PQ Authentication**
  - Experiments with **PQ signature algorithm candidates** to study their impact on TLS 1.3

- Open Quantum Safe Project[2]: liboqs, OQS openssl

[1] https://blog.cloudflare.com/the-tls-post-quantum-experiment/
[2] https://openquantumsafe.org

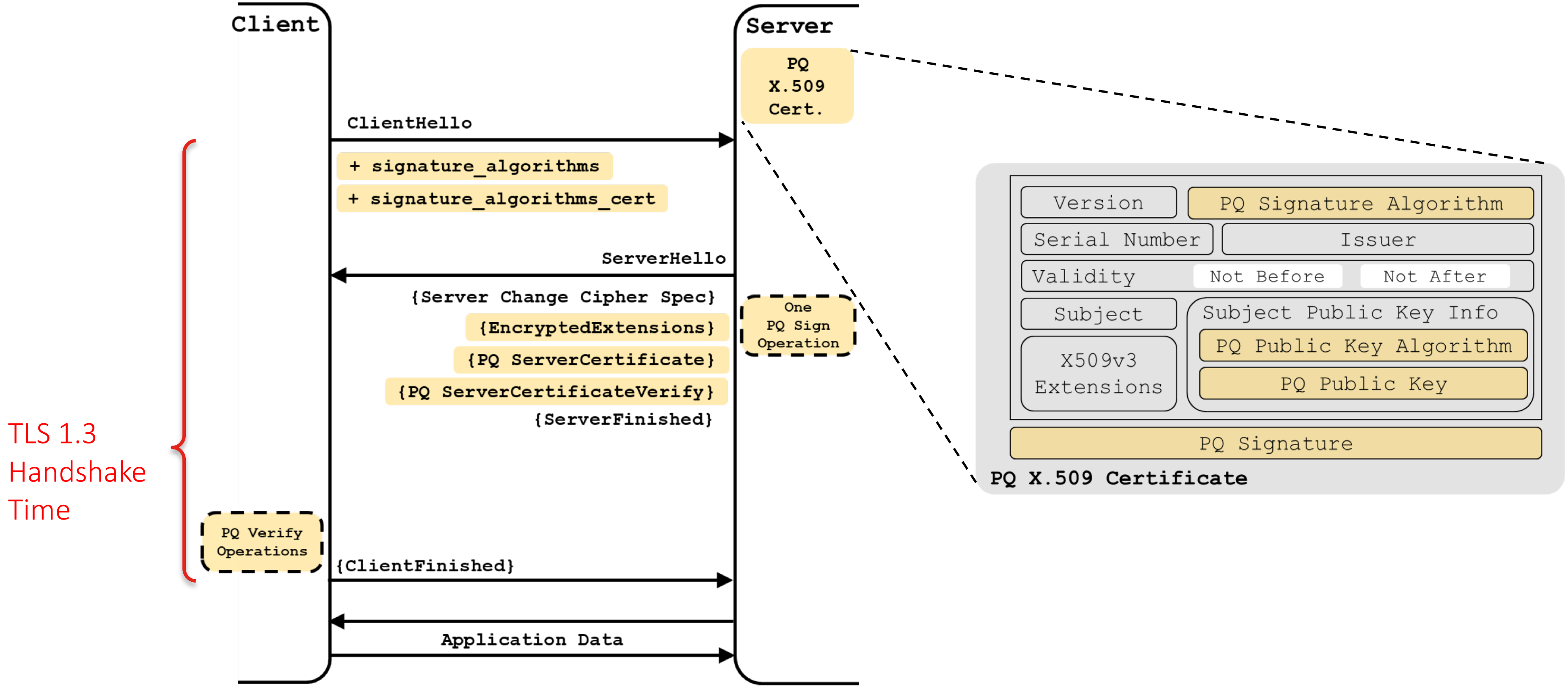
Post-Quantum Authentication in TLS 1.3

- 9 PQ Signature Algorithms for possible integration
  - SPHINCS+, Dilithium, Falcon, MQDSS, Picnic, Rainbow, qTesla, LUOV, GeMSS
- Performance Differences for Sign/Verify Operations
- Various Key/Signature Sizes
- Various Certificate Sizes

Current	PQ
$\sim$ 1 KB to $\sim$ 1.5 KB	$\sim$ 4.3 KB to $>$ 54 KB

- What will be the impact on TLS 1.3?

TLS 1.3 Handshake and PQ X.509 Certificate



Performance of Sign/Verify Operations

- Average Sign and Verify Times

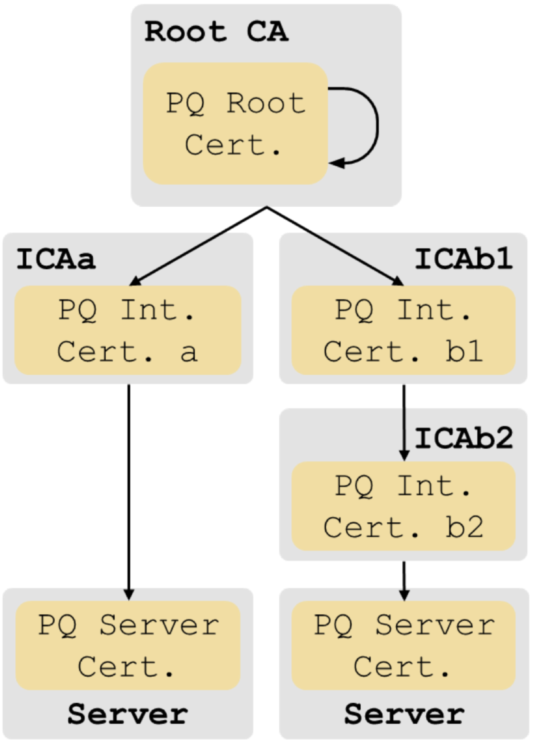
Signature Algorithm	Local Machine (ms) Sign	Verify
RSA 3072	3.19	0.06
ECDSA 384	1.32	1.05
Dilithium II	0.82	0.16
Falcon 512	5.22	0.05
MQDSS 48	10.30	7.25
Picnic $L1FS$	4.09	3.25
SPHINCS$^+$ SHA256-128f-simple	93.37	3.92
Rainbow Ia	0.34	0.83
Dilithium IV	1.25	0.30
Falcon 1024	11.37	0.11

NIST Category 1 (~ 128-bit security)

NIST Category 3 (192-bit security)

NIST Category 5 (256-bit security)

Certificate Chains and Sizes



Signature Algorithm	Cert Chain Size (KB)		CertificateVerify Size (KB)
	One ICA	Two ICAs	
RSA 3072	1.63	2.44	0.38
ECDSA 384	1.34	2.15	0.05
Dilithium *II*	6.90	10.42	2.04
Falcon 512	3.54	5.37	0.69
MQDSS 48	42.24	63.42	20.85
Picnic *L1FS*	66.20	99.57	30.03
SPHINCS$^+$	34.46	51.74	16.98
Rainbow *Ia*	116.86	175.35	0.06
Dilithium *IV*	10.70	16.11	3.37
Falcon 1024	6.56	9.89	1.33

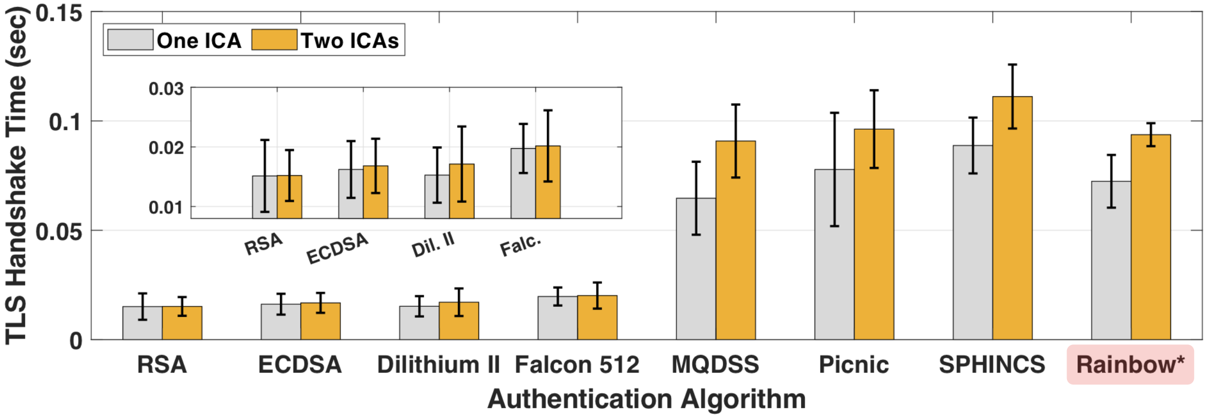
Experimental Procedures

- Goal: Evaluate PQ Authentication Impact on TLS 1.3 under realistic network conditions

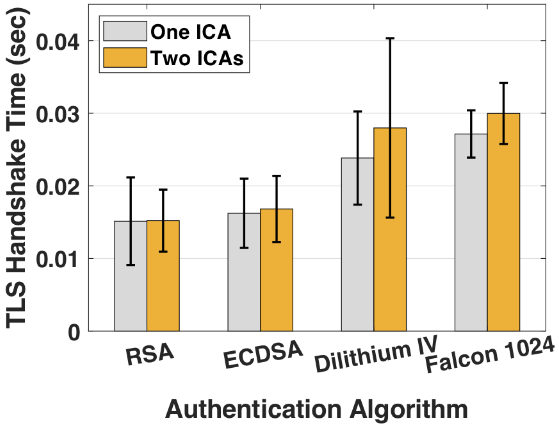
- Local client in RTP, NC – Remote Google Cloud Platform server
- X25519 key exchange
- RSA 3072, ECDSA 384 used as baselines
- No AVX2 optimizations
- TCP initial congestion window parameter at 10 MSS

PQ Handshake Time

NIST Category 1 (~128-bit security)

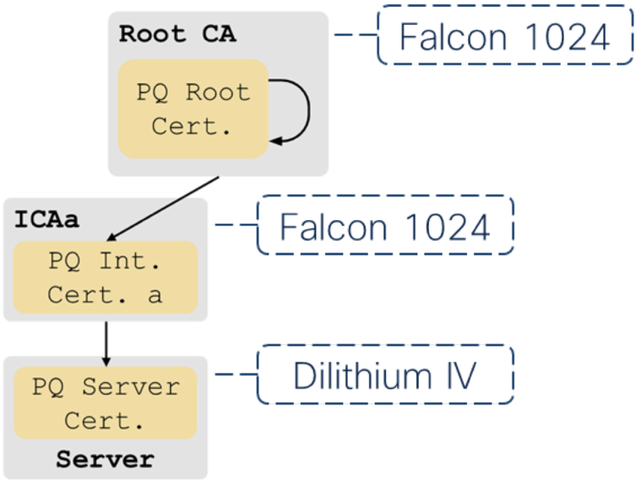


NIST Category 3,5
(~192, 256-bit security)



- `excessive message size` error
- `SSL Alert` for certificate public key size
- *: partial handshake

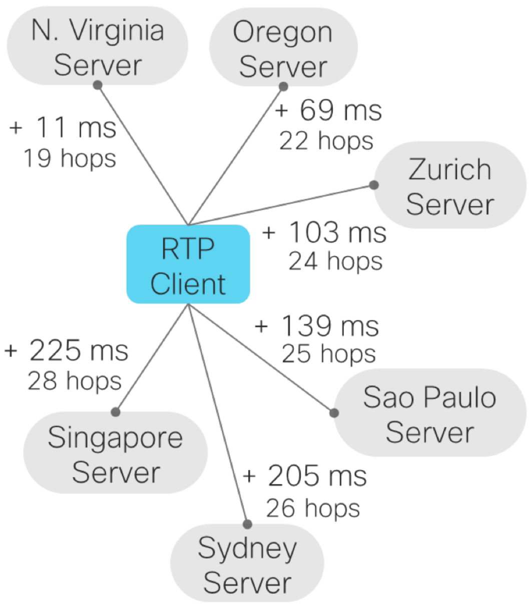
Combining PQ Signature Schemes

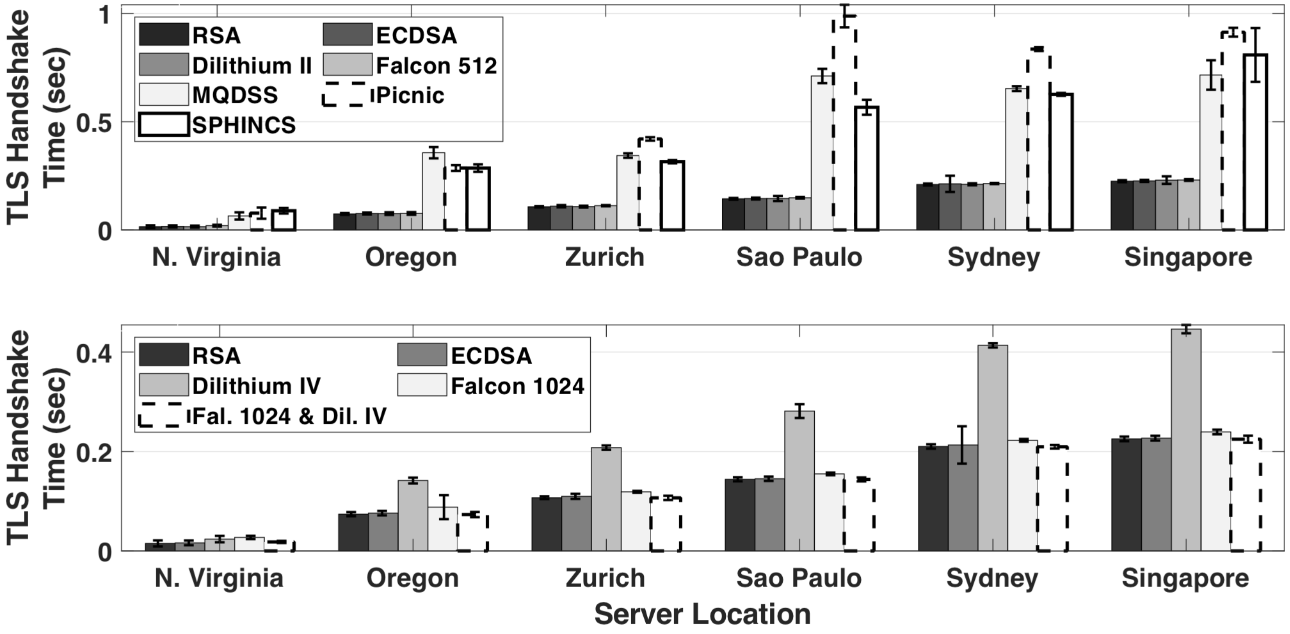


Signature Scheme	TLS Handshake (ms)	
	Mean	St. Dev.
RSA 3072	15.13	6.03
Dilithium *IV*	24.20	2.62
Falcon 1024	27.14	3.30
Fal. 1024 & Dil. *IV*	18.11	1.58

- Single ICA, Client – Server roundtrip ~11ms
- TLS Handshake Time of the Dilithium-Falcon Combination:
  - ↓ 25% vs Dilithium IV
  - ↓ 33% vs Falcon 1024

PQ TLS 1.3 - Global Scale Performance





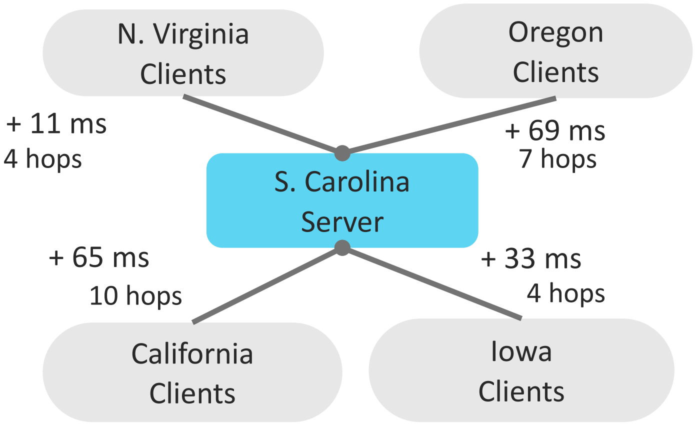
Additional Latency by PQ - Percentiles

- Additional Latency over RSA at the 50th and 95th Percentile
- 5-10% slowdown
- < 20% slowdown for Falcon 1024

Signature Algorithm	Handshake (ms)		Latency (%)	
	50^{th}	95^{th}	50^{th}	95^{th}
RSA3072	131.54	227.26	0	0
Dilithium *II*	140.20	232.51	6.58	2.31
Falcon 512	142.22	235.46	8.12	3.49
MQDSS 48	598.61	726.20	355.05	219.53
Picnic $L1FS$	634.90	985.88	382.63	333.79
SPHINCS^+ 128f	553.15	904.98	320.49	298.19
Dilithium *IV*	276.55	449.88	110.22	97.95
Falcon 1024	152.96	240.74	16.28	5.93
Fal. 1024 & Dil. *IV*	140.74	228.42	6.98	0.50

PQ Authenticated Server – Stress Testing

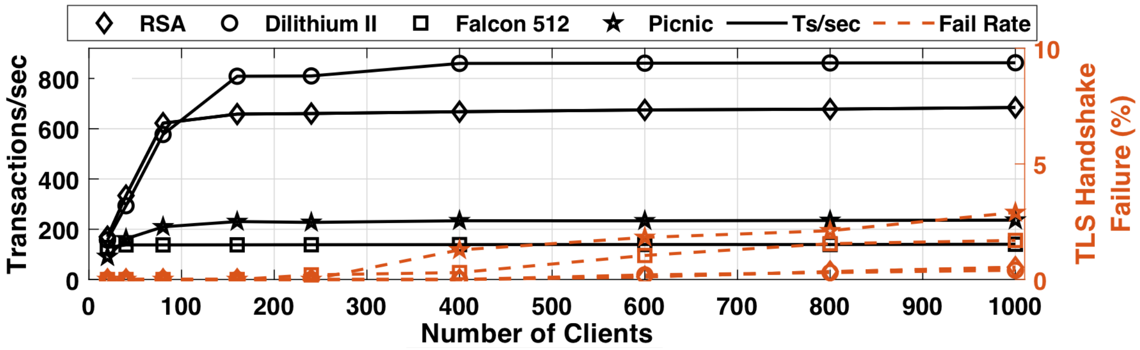
- PQ TLS 1.3 on NGINX Server
- Siege 4.0.4 with PQ TLS 1.3
- Google Cloud Platform servers
- Clients uniformly allocated across four US locations
- Requested webpage size → 0.6 KB



PQ Authenticated Server – Stress Testing

- Dilithium II vs RSA3072:
  - ~25% more connections/sec
- Falcon underperforms due to slow signing

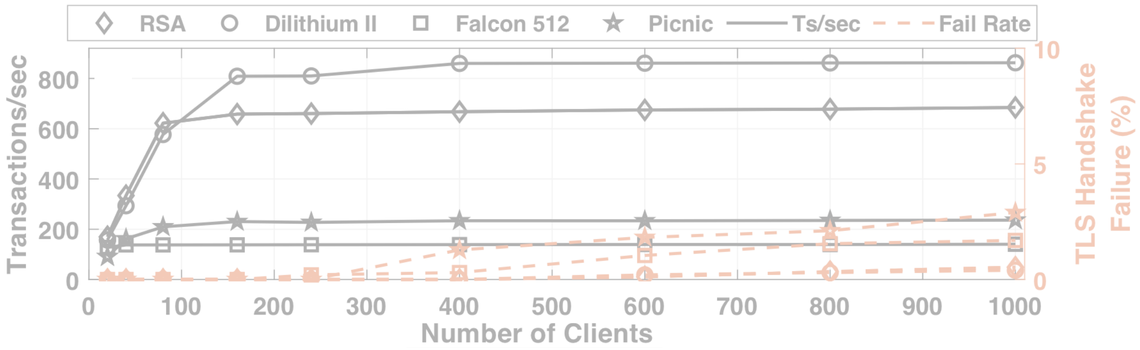
NIST Category 1 (~ 128-bit security)



PQ Authenticated Server – Stress Testing

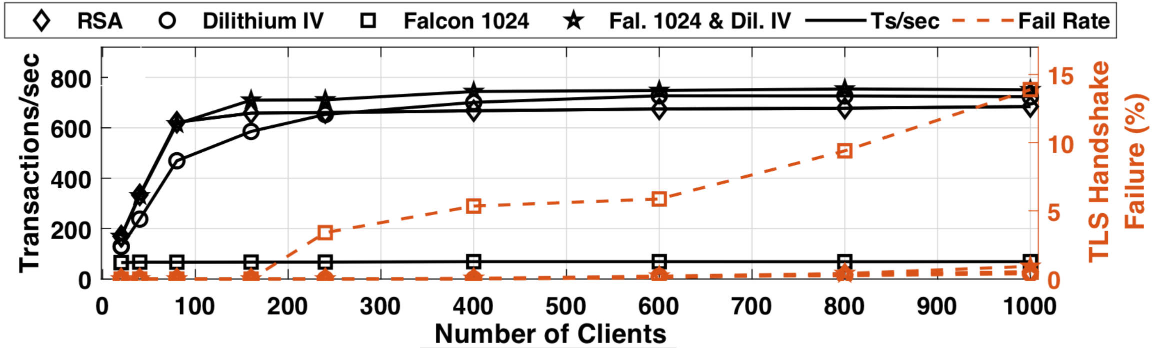
- Dilithium II vs RSA3072:
  - ~25% more connections/sec
- Falcon underperforms due to slow signing

NIST Category 1 (~ 128-bit security)



- Transaction rate of the multi-algorithm combination:
  - ↑ 10% vs RSA 3072
  - ↑ 4% vs Dilithium IV

NIST Category 3,5 (~ 192, 256-bit security)



Changes to Enable PQ Authenticated Tunnels

- ICA Suppression
  - TLS extension to convey ICA certificate unnecessity[1]
  - Omit certificates from handshake using pre-established dictionary[2]
- PQ Scheme Combinations: Root CA
  - Multivariate candidates or Stateful HBS with small tree heights
- Increase TCP initial congestion window parameter (`initcwnd`)
  - >34 MSS to accommodate all PQ algorithms without round-trips
  - Effect on TCP congestion control ?

[1] https://datatracker.ietf.org/doc/html/draft-thomson-tls-sic-00
[2] https://datatracker.ietf.org/doc/html/draft-rescorla-tls-ctls-03

PQ Authenticated Tunnels: Key Takeaways (1/2)

- Dilithium and Falcon
  - Dilithium/Falcon NIST Level 1 performed **sufficiently**, but at <128 bits of classic security
  - Scheme combinations made schemes of NIST Level >3 **competitive**
  - Falcon uses significantly more power than Dilithium[1]

- **Web** connections will be more impacted
  - Short-lived, Small amounts of data per connection
  - Is there an acceptable slowdown value ?

[1] Saarinen, Markku-Juhani O. "Mobile Energy Requirements of the Upcoming NIST Post-Quantum Cryptography Standards." arXiv preprint arXiv:1912.00916 (2019)

PQ Authenticated Tunnels: Key Takeaways (2/2)

- VPNs would not suffer by slower PQ Authentication
  - Long-lived Tunnels, Establishment takes ~5 seconds

- Complications will arise for TLS in case Dilithium/Falcon are not standardized
  - Industry constantly striving for faster handshakes
  - Drastic protocol changes

- Further experimentation
  - PQ **Key Exchange** (Cloudflare, Google) **+ Authentication** impact on tunnels
  - Impact of PQ signatures on authenticated tunnels in **lossy environments** (e.g. wireless)

Thank you!

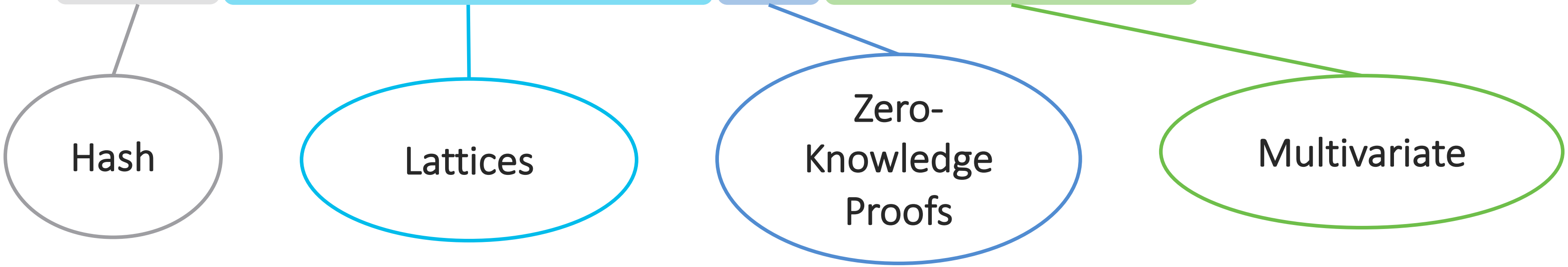
Questions?

dsike@unm.edu

Appendix

Post-Quantum Authentication – NIST Candidates

- 9 PQ Signature Algorithms for possible integration
  - SPHINCS+, Dilithium, qTesla, Falcon, Picnic, Picnic, LUOV, GeMSS, Rainbow



Dilithium: MLWE - Module Learning with Errors
Falcon: NTRU with Fast Fourier trapdoor Gaussian sampling
qTesla: R-LWE
Picnic: Multiparty computation as (Zero Knowledge Proofs) using Hash commitment