

Field Experiments on Post-Quantum DNSSEC

Jason Goertzen <jason.goertzen@sandboxaq.com>
Peter Thomassen <peter@desec.io>
Nils Wisiol <nils@desec.io>

MAPRG, Vancouver – July 24, 2024

Context & Motivation

- In 2022, performed (local-only) DNSSEC study with **Falcon** in PowerDNS
 - Results: <https://blog.powerdns.com/2022/04/07/falcon-512-in-powerdns>
- Now: Broader experiments with **multiple PQC algorithms**
 - fast validation, short signatures, short-ish keys
- Goal: **Public deployment** on the Internet, to investigate ...
 - behavior of non-PQC-aware resolvers typically used by clients
 - behavior of PQC-aware resolvers
- Parameters:
 - KSK/ZSK (BIND) vs. CSK (PowerDNS)
 - Name existence and NSEC vs. [NSEC3 conventional (BIND) vs. minimal (PowerDNS)]
 - UDP vs. TCP
 - DO bit

Algorithm Considerations

Algorithm	NIST Verdict	Approach	Private key	Public key	Signature	Sign/s	Verify/s
Crystals-Dilithium-II [29]	Finalist	Lattice	2.8kB	1.2kB	2.0kB		
Falcon-512 [31]	Finalist	Lattice	57kB	0.9kB	0.7kB	3,307	20,228
Rainbow- I_a [56]	Finalist	Multivariate	101kB	158kB	66B	8,332	11,065
RedGeMSS128 [16]	Candidate	Multivariate	16B	375kB	35B	545	10,365
Sphincs ⁺ -Haraka-128s [11]	Candidate	Hash	64B	32B	8kB		
Picnic-L1-FS [17]	Candidate	Hash	16B	32B	34kB		
Picnic2-L1-FS [17]	Candidate	Hash	16B	32B	14kB		
EdDSA-Ed25519 [12]		Elliptic curve	64B	32B	64B	25,935	7,954
ECDSA-P256 [12]		Elliptic curve	96B	64B	64B	40,509	13,078
RSA-2048 [12]		Prime	2kB	0.3kB	0.3kB	1,485	49,367

Müller, M. et al.: Retrofitting post-quantum cryptography in internet protocols: a case study of DNSSEC. SIGCOMM Comput. Commun. Rev. 50, 49–57 (2020)

- Selected algorithms with public keys and signatures < 10 KB
- Plus: a stateful hash-based algorithm (XMSS)

Steps Taken

- Implemented via **liboqs** (with regular unassigned algorithm numbers)
 - Falcon512
 - Dilithium2
 - SPHINCS+-SHA256-128s
 - XMSSmt-SHA256-h40-4 / XMSSmt-SHA256-h40-8 (and other parameter sets)
- Measurements using **RIPE ATLAS** (~10,000 probes, ~2M queries in May 2024)
- Output variables: **Rcode, Correctness, AD bit, response time**
- Pre-selection: Exclude ...
 - probe-resolver combinations with incorrect response for RSA-SHA256 (due to noise)
 - resolvers in private IP ranges (due to RIPE ATLAS limitation for TCP)
 - timeouts and network errors

Queries Using an Unaware Resolver

```
dig +dnssec A dilithium2.pdns.pq-dnssec.dedyn.io @8.8.8.8
```

```
;; Truncated, retrying in TCP mode.

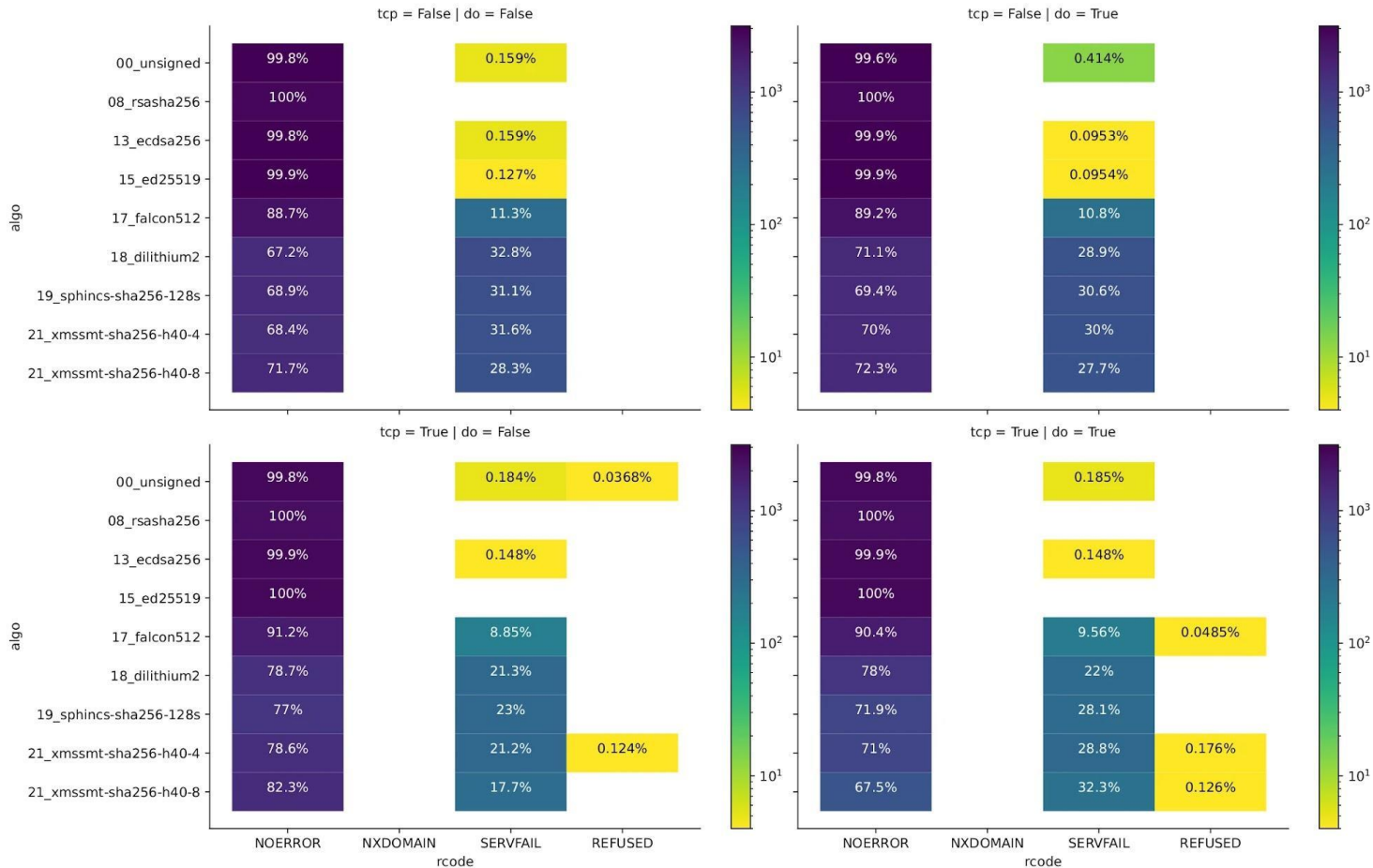
; <<>> DiG 9.18.24-0ubuntu0.22.04.1-Ubuntu <<>> +dnssec A dilithium2.pdns.pq-dnssec.dedyn.io @8.8.8.8
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29078
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 512
;; QUESTION SECTION:
;dilithium2.pdns.pq-dnssec.dedyn.io. IN A

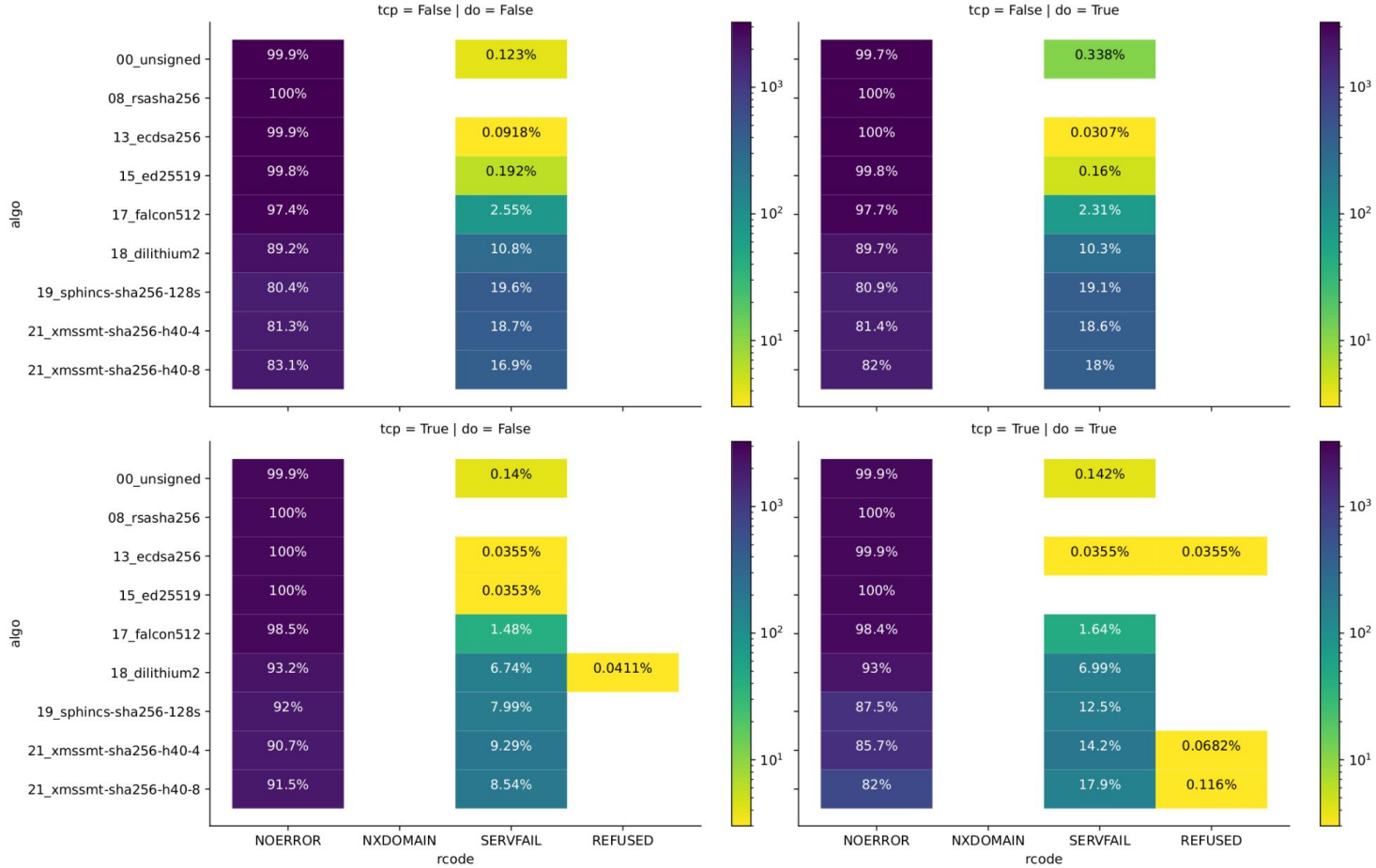
;; ANSWER SECTION:
dilithium2.pdns.pq-dnssec.dedyn.io. 3600 IN A 95.217.209.184
dilithium2.pdns.pq-dnssec.dedyn.io. 3599 IN RRSIG A 18 5 3600 20240801000000 20240711000000 3978 dilithium2.pdns.pq-dnssec.dedyn.io. 19/28JXGCgGbNtEAtU0zv1/SzP+kr6vBlglWrJ/ZfYgdC1DXZHdh+xo1 rnZ9uhvmADCqZzJX0y0U1Tyw2sHN32Vmcv4KLR8lI7TBwftJq6T3nGfV oQnv9 DNvPJTyb4VonYH3fLTMYeQ3/0WY9gbv0ngy55QqRjw+ikhS0yIp ezpZYH3ArY/xxmTgM70BW0yBg3gXgo1G2mrX97ufqrwk0/n0Vu/xXfSI npGKq+dVu7LQQR7nM lmM3FkbaRAFyo0FjmbzXDPTyrwJekJP8dfQ5zvc p0CRfrpjrJg+ZBUofhdk1PUR0539JwD[...]AA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABAcJzc=

;; Query time: 208 msec
;; SERVER: 8.8.8.8#53(8.8.8.8) (TCP)
;; WHEN: Fri Jul 19 23:57:40 PDT 2024
;; MSG SIZE rcvd: 2565
```

vendor='bind9', is_nx=False, good-rsa

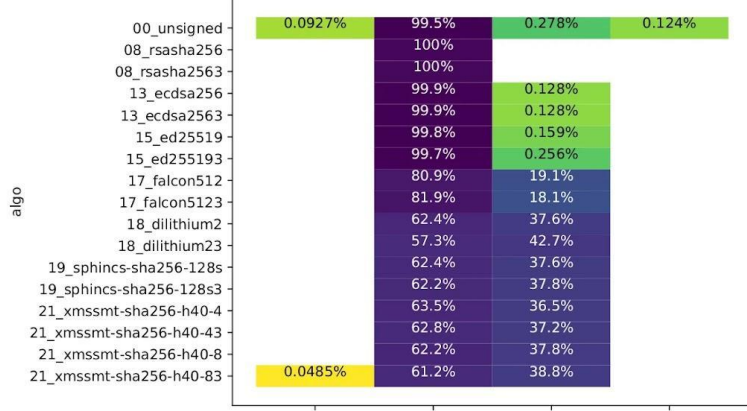


vendor='pdns', is_nx=False, good-rsa



vendor='bind9', is_nx=True, good-rsa

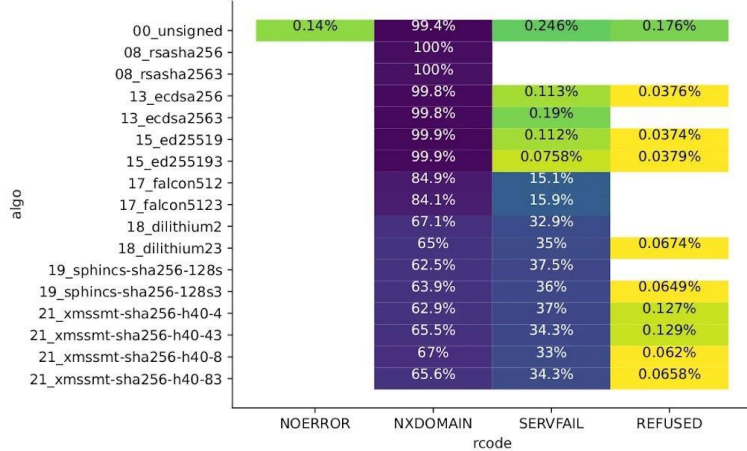
tcp = False | do = False



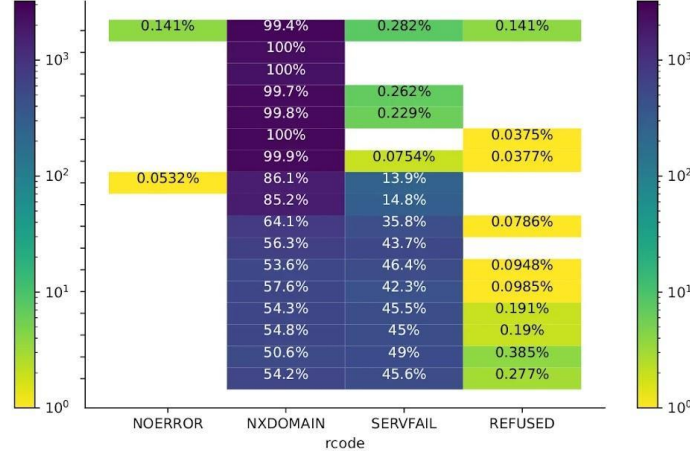
tcp = False | do = True



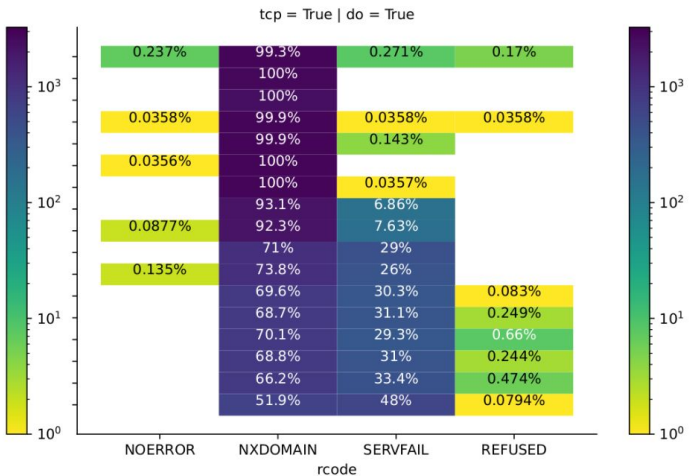
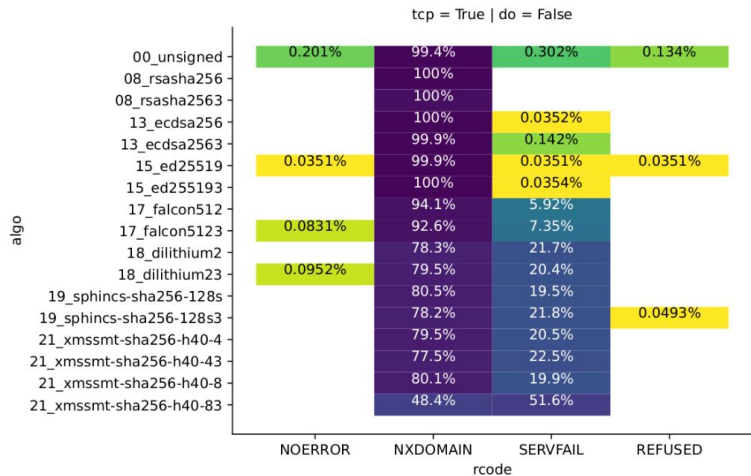
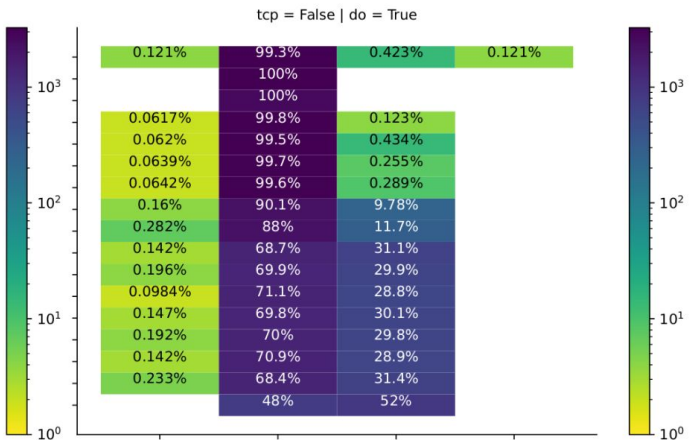
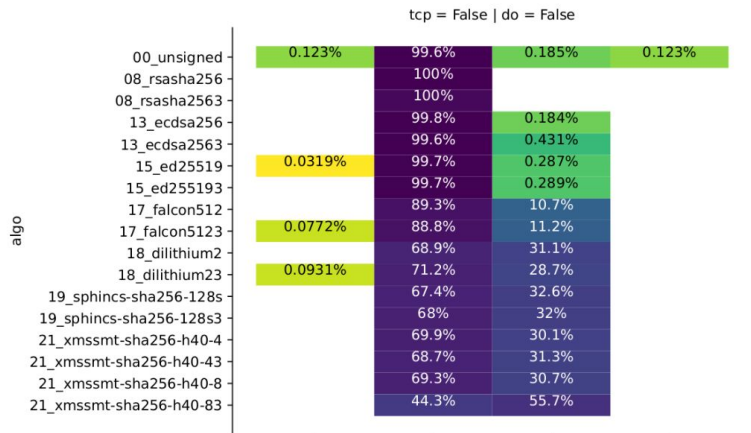
tcp = True | do = False



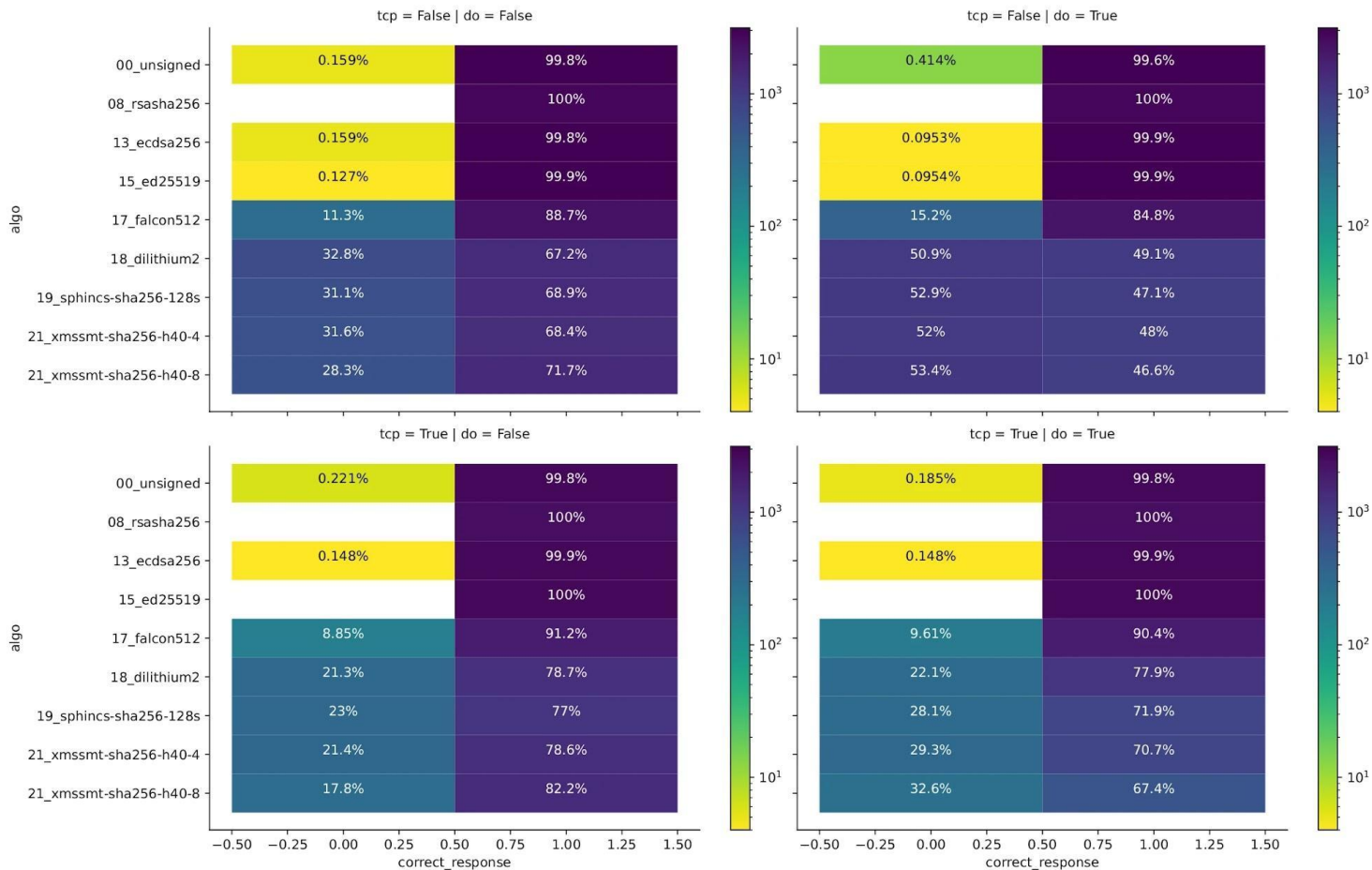
tcp = True | do = True



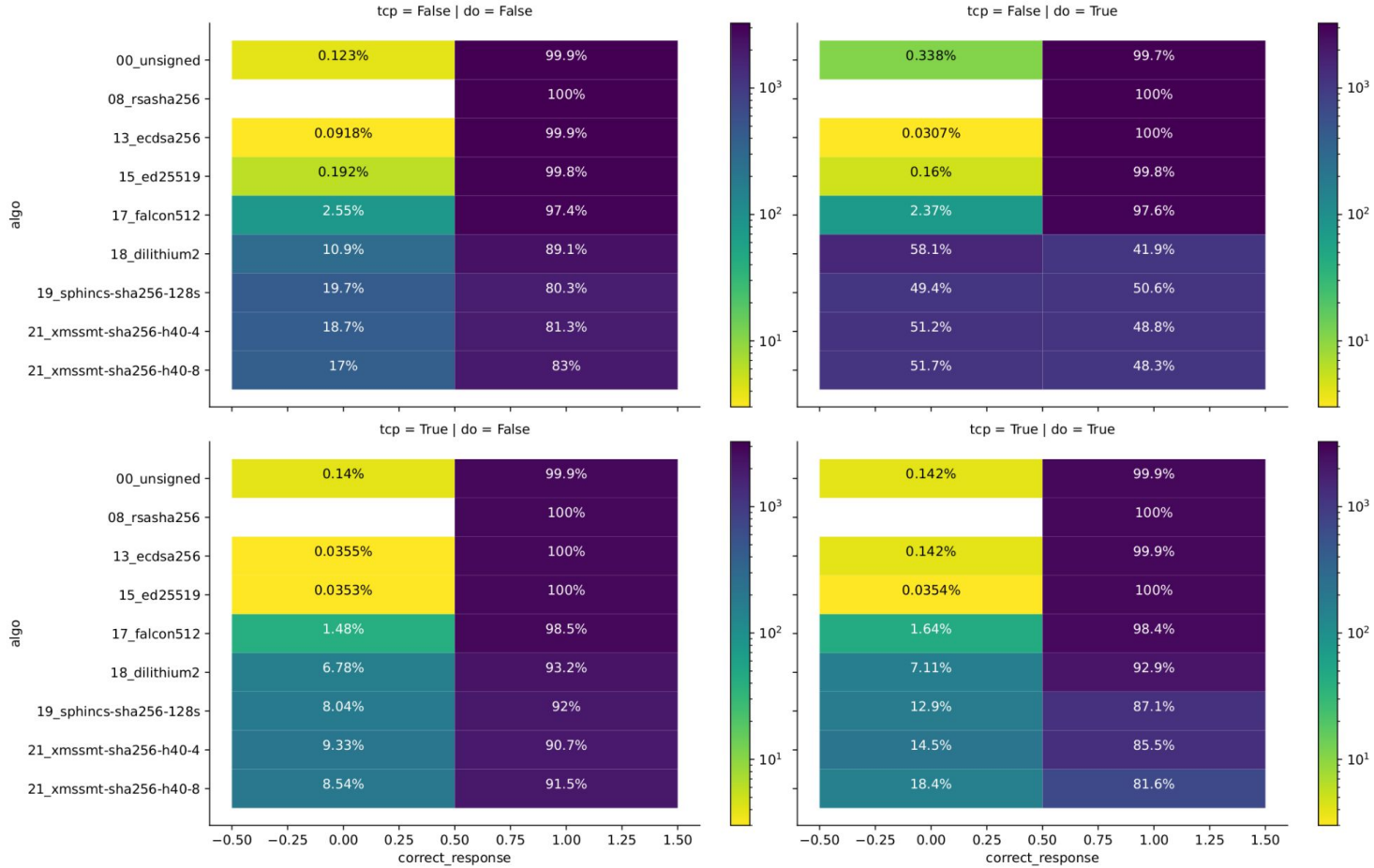
vendor='pdns', is_nx=True, good-rsa



vendor='bind9', is_nx=False, good-rsa

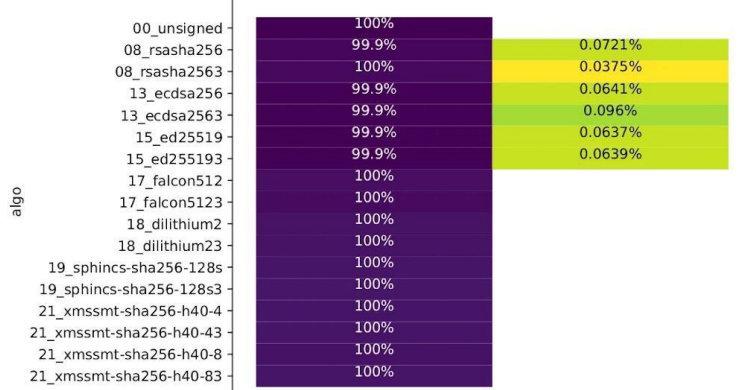


vendor='pdns', is_nx=False, good-rsa



vendor='bind9', is_nx=True, good-rsa

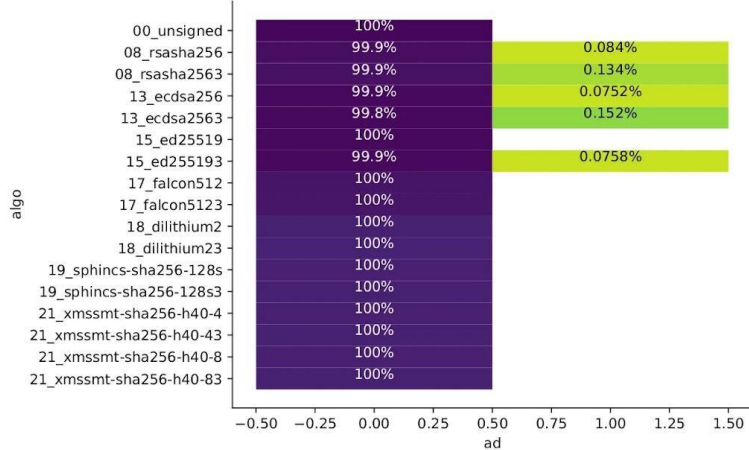
tcp = False | do = False



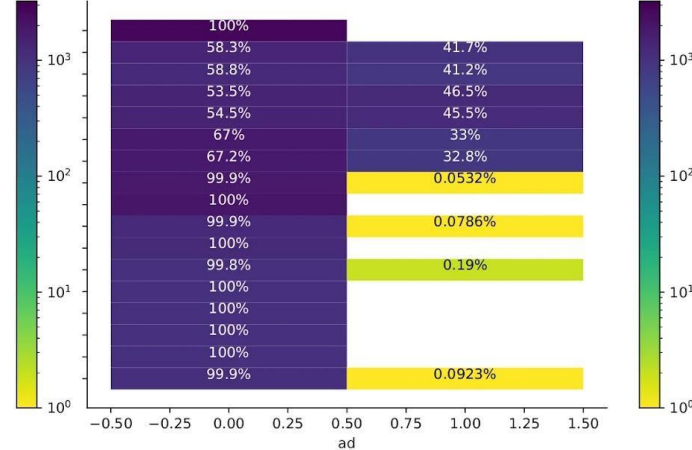
tcp = False | do = True



tcp = True | do = False

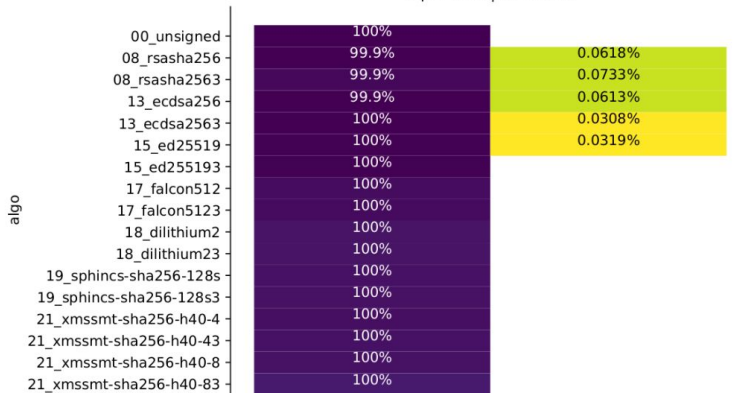


tcp = True | do = True

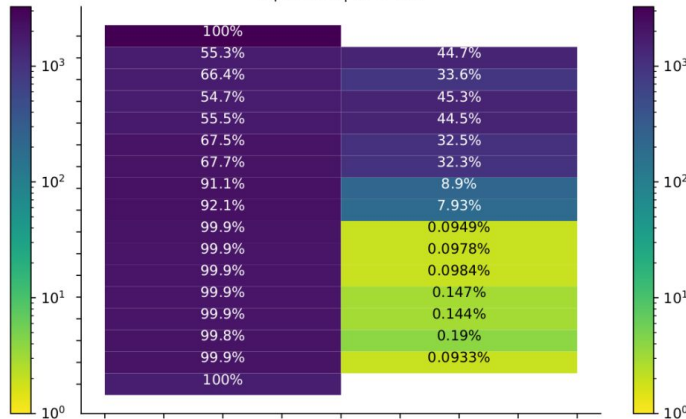


vendor='pdns', is_nx=True, good-rsa

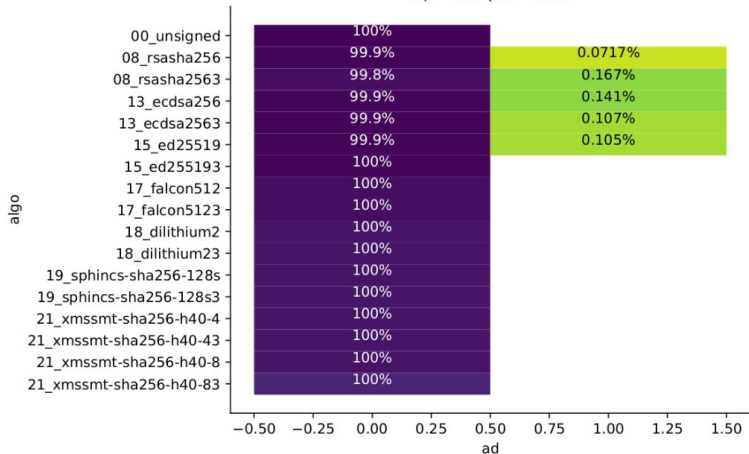
tcp = False | do = False



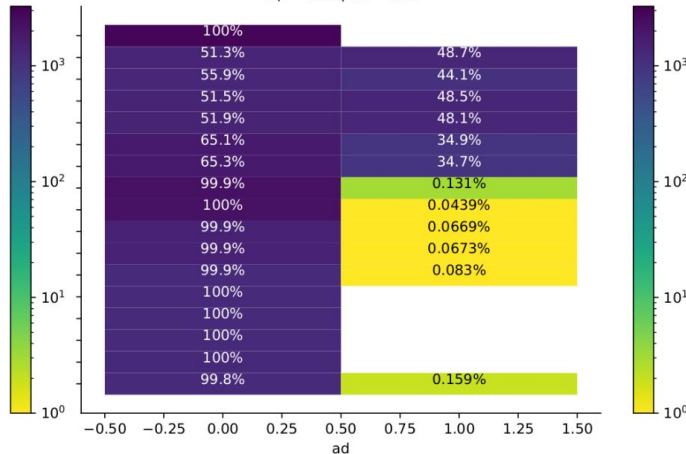
tcp = False | do = True



tcp = True | do = False



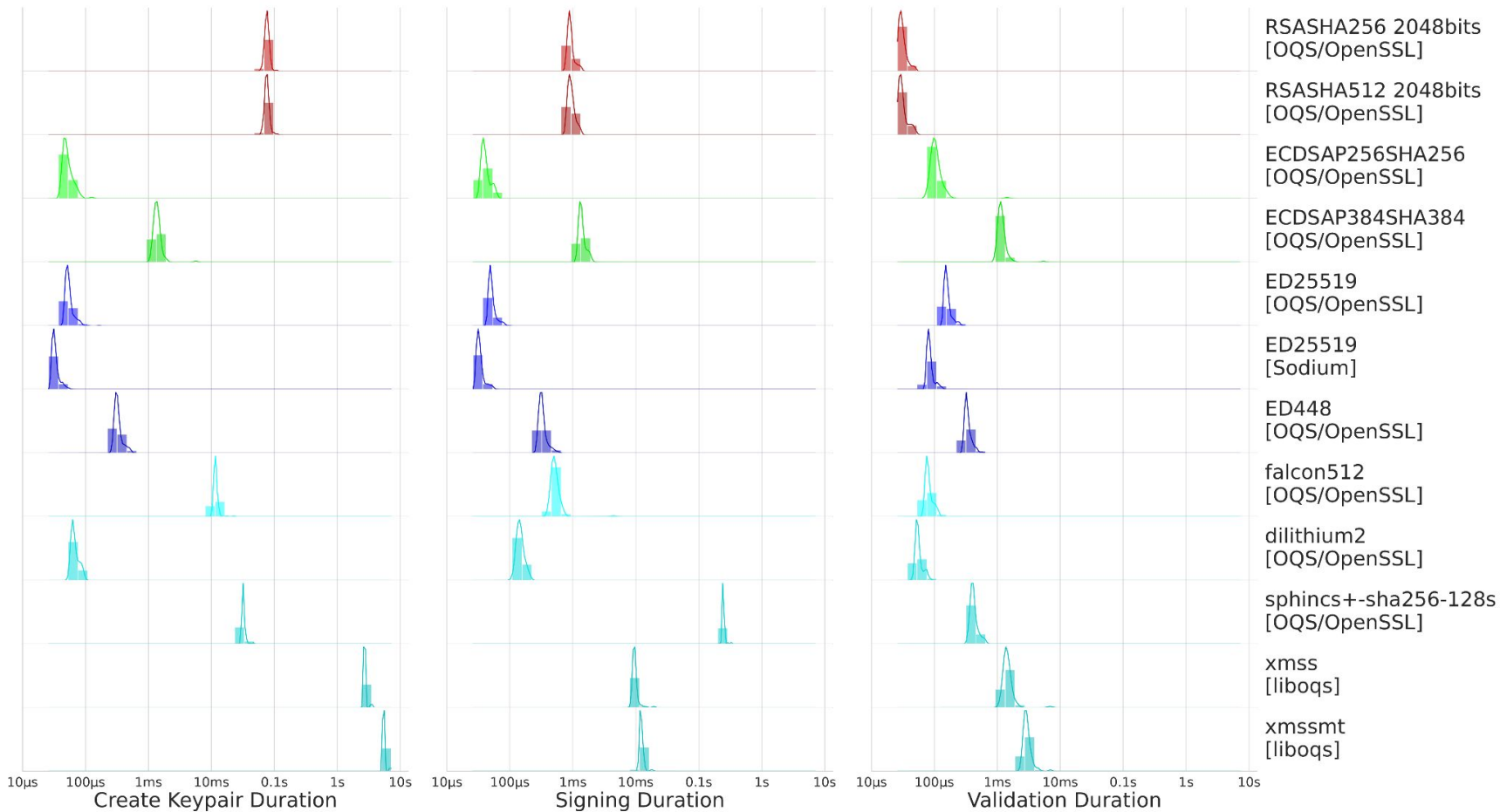
tcp = True | do = True



Takeaways

- **Transmission issues are real**
 - PQC response delivery rates go down significantly as response sizes increase → Falcon leads
 - Gets worse depending on circumstances, like with DO bit or with NSEC3
- **UDP & DO=0:**
 - ~70% KSK/ZSK responses correct
 - ~80% CSK responses correct
 - Goes up by ~10% via TCP
- **UDP & DO=1:**
 - ~50% responses correct
 - Goes up by ~20–40% via TCP
- 8.5% of probe-resolver pairs claim successfully validating Falcon

Crypto Algorithm Run Time (PowerDNS)



Queries Using a PQC-aware Resolver

```
dig +dnssec A dilithium2.pdns.pq-dnssec.dedyn.io @bind9.pq-dnssec.dedyn.io -p 5304
```

```
;; Truncated, retrying in TCP mode.
```

```
; <<>> DiG 9.18.24-0ubuntu0.22.04.1-Ubuntu <<>> +dnssec A dilithium2.pdns.pq-dnssec.dedyn.io @bind9.pq-dnssec.dedyn.io -p 5304
```

```
;; global options: +cmd
```

```
;; Got answer:
```

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22245
```

```
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1
```

```
;; OPT PSEUDOSECTION:
```

```
; EDNS: version: 0, flags: do; udp: 1232
```

```
; COOKIE: 8455829f86d7fb7601000000669b5d9517dfc67dff539cac (good)
```

```
;; QUESTION SECTION:
```

```
;dilithium2.pdns.pq-dnssec.dedyn.io. IN A
```

```
;; ANSWER SECTION:
```

```
dilithium2.pdns.pq-dnssec.dedyn.io. 3599 IN A 95.217.209.184
```

```
dilithium2.pdns.pq-dnssec.dedyn.io. 3599 IN RRSIG A 18 5 3600 20240801000000 20240711000000 3978 dilithium2.pdns.pq-dnssec.dedyn.io. 19/28JXGCGbNtEAtU0zv1/SzP+kr6vBlglWrJ/ZfYgdC1DXZHdh+xol rnZ9uhvmADCqZzJX0y0U1Tyw2sHN32Vmcv4KLR81I7TBwfTJq6T3nGfV oQnv9DNvPJTyb4VonYH3fLTMYeQ3/0Wy9gbv0ngy55QqRjw+ikhS0yIp ezpZYH3ArY/xmTgM70BW0yBg3gXgo1G2mrX97ufqrwk0/n0Vu/xXfSI npGKq+dVu7LQQR7nMlmM3FkbaRAFyo0FjmbzXDPTyrwJekJP8dfQ5zvc pOCRfrpjRg+ZBUofhdk1PUR0539JwD[...shortened...]AA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABAcJzc=
```

```
;; Query time: 56 msec
```



```
;; SERVER: 35.232.14.170#5304(bind9.pq-dnssec.dedyn.io) (TCP)
```

```
;; WHEN: Fri Jul 19 23:47:49 PDT 2024
```

```
;; MSG SIZE rcvd: 2593
```


Post-Quantum DNSSEC Testbed with BIND and PowerDNS

Query our PQC-enabled DNS Resolvers

Send queries to our post-quantum enabled validating resolver! You can choose from a number of post-quantum (and classical) signing schemes, NSEC or NSEC3 mode, and implementations for PowerDNS ([source](#) ) and BIND ([source](#) )

Zones signed accordingly are available at `{algorithm}.{vendor}.pq-dnssec.dedyn.io`, and each has a **A** and a **TXT** record configured. To query a non-existing name, prepend the **nx** label (for example).

Queries will be sent from your browser using DNS-over-HTTPS to a BIND or PowerDNS resolvers with validation support for the selected algorithm. The resolver will talk to the corresponding BIND or PowerDNS authoritative DNS server (again, with support for the selecting signing scheme), to get your response. It will then validate the signature and send the result to your browser.

All queries are send with the **DNSSEC_OK** flag (`+dnssec` in dig), so you will see **RRSIG** and **NSEC/NSEC3** records the the responses.

Query type	Algorithm	Authoritative vendor	Resolver vendor
	Dilithium2	PowerDNS	PowerDNS

NSEC3 non-existent name

Domain name
nx.dilithium2.pdns.pq-dnssec.dedyn.io

[▶ QUERY](#) [DNSviz !\[\]\(d28ca9ce3e0e76b2108fafdd8e1b04fb_img.jpg\)](#)

Try it yourself!

<https://pq-dnssec.dedyn.io/>

Future: Merkle Trees

- Want to use PQC while keeping messages small
- Use Merkle trees to compress zone signing overhead
 - Signatures become authenticating paths
 - ZSK DNSKEY becomes the root hash
 - KSK is some secure algorithm that needs compression

Zone size

A zone file containing 10,000 signatures

Algorithm	Size
ECDSAP256	2.18 MB

Zone size

A zone file containing 10,000 signatures

Algorithm	Size
Dilithium2	38.8 MB
Dilithium2 + Merkle Tree	10.16 MB (3.82x less)

Algorithm	Size
ECDSAP256	2.18 MB

Zone size

A zone file containing 10,000 signatures

Algorithm	Size	Algorithm	Size
Dilithium2	38.8 MB	Falcon512	11.49 MB
Dilithium2 + Merkle Tree	10.16 MB (3.82x less)	Falcon512 + Merkle Tree	10.16 MB (1.13x less)

Algorithm	Size
ECDSAP256	2.18 MB

Plan to measure performance and release results in coming months

Thank you!

Acknowledgments:



Questions?



Backup: Outlook

- Fixing may require **revamping signature representation** in DNS
 - [ARRF?](#)
 - Does not necessarily involve a wire format / spec change
 - Or will more robust DoT/DoH/DoQ gain enough traction?
- What would it take to **make the root quantum-safe?**
 - Further complications from double-signing – is this really needed?
- To transition, **any scalable solution will require DS provisioning automation**
- Future work needed!
→ [Research agenda](#)