Preparing DNSSEC for quantum computing

Moritz Müller | Nordic Domain Days | 2022-05-10

Just to be sure everyone is on the same page …

- DNSSEC adds **authenticity** and **integrity** to the DNS
- Domain operators **sign** their domain name using **cryptographic algorithms**
- Recursive resolvers can be sure that they've received the correct information if they **validate** the **signatures**

Photo by Towfiqu barbhuiya on Unsplash

Attacking DNS(SEC), hypothetically

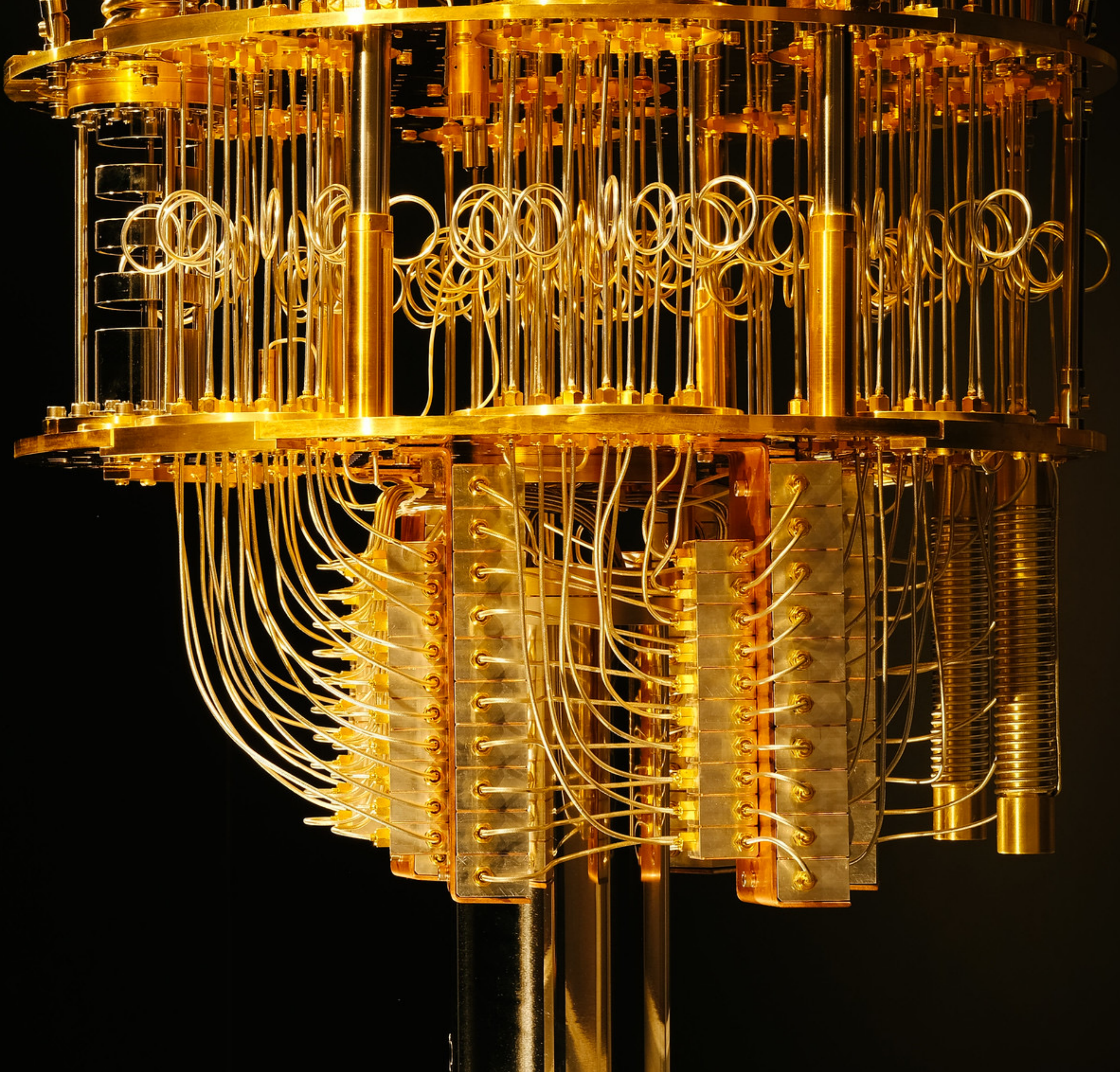
1) Steal secret key used for signing a domain name
2) Create fake resource records e.g., with a malicious IP address
3) Sign fake resource record with the stolen key
4) Perform "regular" cache poisoning attack against a recursive resolver

→ The resolver believes that the fake record is valid

Attacking DNS(SEC), hypothetically

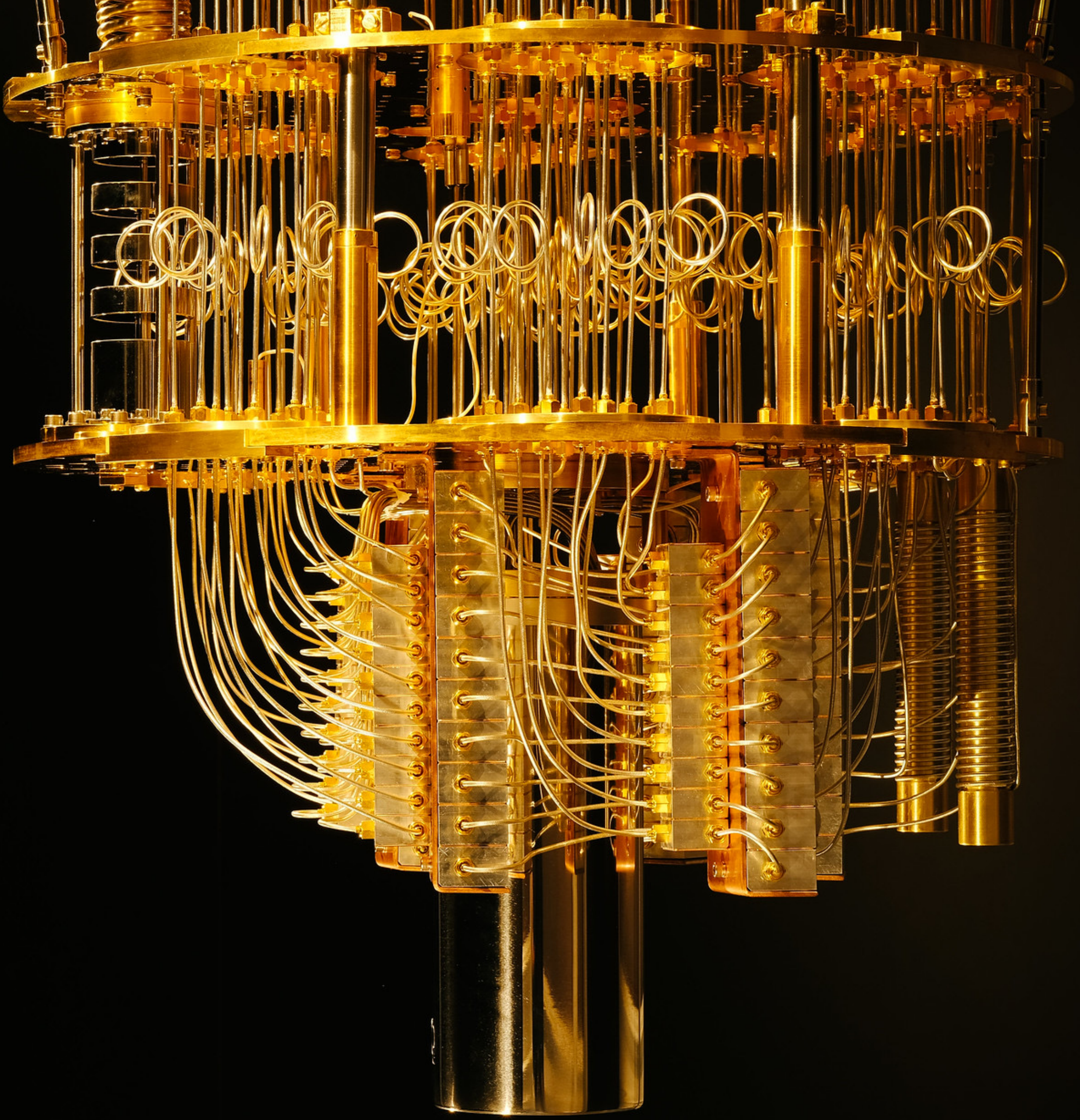
1) ~~Steal~~ **Recreate** secret key used for signing a domain name
2) Create fake resource records e.g., with a malicious IP address
3) Sign fake resource record with the stolen key
4) Perform “regular” cache poisoning attack against a recursive resolver

→ The resolver believes that the fake record is valid



A Quantum Computer

- Can run Shor's algorithm
- *Could* break the keys of all cryptographic algorithms currently used in DNSSEC
- Unclear if and when a powerful enough computer exists



Why bother now?

Things take time:

1) Finding a suitable quantum-safe algorithm

2) Adapting it for DNSSEC

3) Rolling it out on a larger scale

Finding a suitable quantum-safe algorithm

Algorithm	Approach	Private key	Public key	Signature	Status
Crystals-Dilithium-II	Lattice	2.8kB	1.3kB	2.4kB	Finalist
Falcon-512	Lattice	1.3kB	0.9kB	0.7kB	Finalist
Rainbow-I	Multivariate	101kB	158kB	64B	Finalist
RedGeMSS-128	Multivariate	16B	375kB	36B	Alternate
Sphincs+-128s	Hash	64B	32B	8kB	Alternate
Picnic-L1-FS	Hash/ZKP	16B	32B	33kB	Alternate
EdDSA-Ed22519	Elliptic curve	64B	32B	64B	Currently used

Security level ~ 1, Source https://csrc.nist.gov/News/2020/pqc-third-round-candidate-announcement

Finding a suitable quantum-safe algorithm

Algorithm	Approach	Private key	Public key	Signature	Status
Crystals-Dilithium-II	Lattice	2.8kB	1.3kB	2.4kB	Finalist
Falcon-512	Lattice	1.3kB	0.9kB	0.7kB	Finalist
~~Rainbow-I~~	~~Multivariate~~	~~101kB~~	~~158kB~~	~~64B~~	~~Finalist~~
~~RedGeMSS-128~~	~~Multivariate~~	~~16B~~	~~375kB~~	~~36B~~	~~Alternate~~
Sphincs+-128s	Hash	64B	32B	8kB	Alternate
Picnic-L1-FS	Hash/ZKP	16B	32B	33kB	Alternate
EdDSA-Ed22519	Elliptic curve	64B	32B	64B	Currently used

Security level ~ 1, Source https://csrc.nist.gov/News/2020/pqc-third-round-candidate-announcement

KISS: Keep it Small, Stupid

→Large DNS messages

→Fragmentation

→ Increased RTTs, packet loss, and security vulnerability

Finding a suitable quantum-safe algorithm

Algorithm	Approach	Private key	Public key	Signature	Status
Crystals-Dilithium-II	Lattice	2.8kB	1.3kB	2.4kB	Finalist
Falcon-512	Lattice	1.3kB	0.9kB	0.7kB	Finalist
~~Rainbow-I~~	~~Multivariate~~	~~101kB~~	~~158kB~~	~~64B~~	~~Finalist~~
~~RedGeMSS-128~~	~~Multivariate~~	~~16B~~	~~375kB~~	~~36B~~	~~Alternate~~
Sphincs+-128s	Hash	64B	32B	8kB	Alternate
Picnic-L1-FS	Hash/ZKP	16B	32B	33kB	Alternate
EdDSA-Ed22519	Elliptic curve	64B	32B	64B	Currently used

Security level ~ 1, Source https://csrc.nist.gov/News/2020/pqc-third-round-candidate-announcement

Finding a suitable quantum-safe algorithm

Algorithm	Approach	Private key	Public key	Signature	Status
Crystals-Dilithium-II	Lattice	2.8kB	1.3kB	2.4kB	Finalist
Falcon-512	Lattice	1.3kB	0.9kB	0.7kB	Finalist
~~Rainbow-I~~	~~Multivariate~~	~~101kB~~	~~158kB~~	~~64B~~	~~Finalist~~
~~RedGeMSS-128~~	~~Multivariate~~	~~16B~~	~~375kB~~	~~36B~~	~~Alternate~~
Sphincs+-128s	Hash	64B	32B	8kB	Alternate
Picnic-L1-FS	Hash/ZKP	16B	32B	33kB	Alternate
EdDSA-Ed22519	Elliptic curve	64B	32B	64B	Currently used

Security level ~ 1, Source https://csrc.nist.gov/News/2020/pqc-third-round-candidate-announcement

How far are we?

1) Finding a suitable quantum-safe algorithm

2) Adapting it for DNSSEC

3) Rolling it out on a larger scale

How far are we?

1) Finding a suitable quantum-safe algorithm ***WIP***

2) Adapting it for DNSSEC

3) Rolling it out on a larger scale

How far are we?

1) Finding a suitable quantum-safe algorithm ***WIP***

2) Adapting it for DNSSEC ***WIP***

3) Rolling it out on a larger scale

How far are we?

1) Finding a suitable quantum-safe algorithm	***WIP***
2) Adapting it for DNSSEC	***WIP***
3) Rolling it out on a larger scale	***X***

Is there something operators can do?

- Make sure that you follow current DNS best practices
- Make sure that you follow current DNSSEC best practices

Photo by Cesar Carlevarino Aragon on Unsplash

Open Questions

- Are messages above 1.2kB but smaller than 64kB really that bad?
- What about performance?
- If and how could hash based algorithms deployed?
- Do we have to move away entirely from the current DNSSEC model, and should we rely on KEMs?
- When do we really need to get moving?

Photo by Alessandro Bianchi on Unsplash

Are there any questions?

Follow us

.nl sidnlabs.nl

@moritzcm_

Thank you for your attention!