The Impact of Post-Quantum Cryptography on DNSSEC

Encrypted DNS Call – 2021-02-22

Moritz Müller[1,2], Maran van Heesch[3], **Jins de Jong**[3], Benno Overeinder[4], Roland van Rijswijk-Deij[2,4]

[1]SIDN Labs, [2]University of Twente, [3]TNO, [4]NLnet Labs

The Problem

- Quantum Computers *could* break current public-key cryptography
- This is a threat to many Internet protocols, *including DNSSEC*
- New *quantum-safe* algorithms are assessed

Main Research Question:

Are these new quantum-safe algorithms suitable for DNSSEC?

Introduction to Post Quantum Cryptography

Threat to cryptography

- Better search algorithms:

- Grover's algorithm ($t \to \sqrt{t}$)
- Symmetric cryptography is not broken. Only double key sizes needed.

- Finding subgroups:

- Shor's algorithm ($e^{at} \to t^b$)

- Shor's algorithm breaks RSA and discrete logarithm cryptography.

- **All current public key cryptography must be replaced by a quantum-safe alternative!**

- When: perhaps in the 2030's

- Google claimed quantum supremacy in 2019.

Post-quantum cryptography

- No classical or quantum algorithm to break it (quickly) is known.
- The same structure as public key cryptography (public / secret key).
- From them key encapsulation mechanisms (KEM's) and signature algorithms can be generated.
- **For DNSSEC the signature schemes are most interesting.**

Multivariate cryptography

- Bases on systems of polynomial equations in several variables.

- Essential idea:
- - P is a system of m polynomial equations in n variables.
- $(c_1, c_2, \dots, c_m) = P(y_1, y_2, \dots, y_n)$

- KEM: Given a cipher text, there may only be one y: ($m < n$)
- This is hard to construct.

- SIGN: Given a signature, it should be difficult to find any y: ($m > n$)
- This is easy to construct.

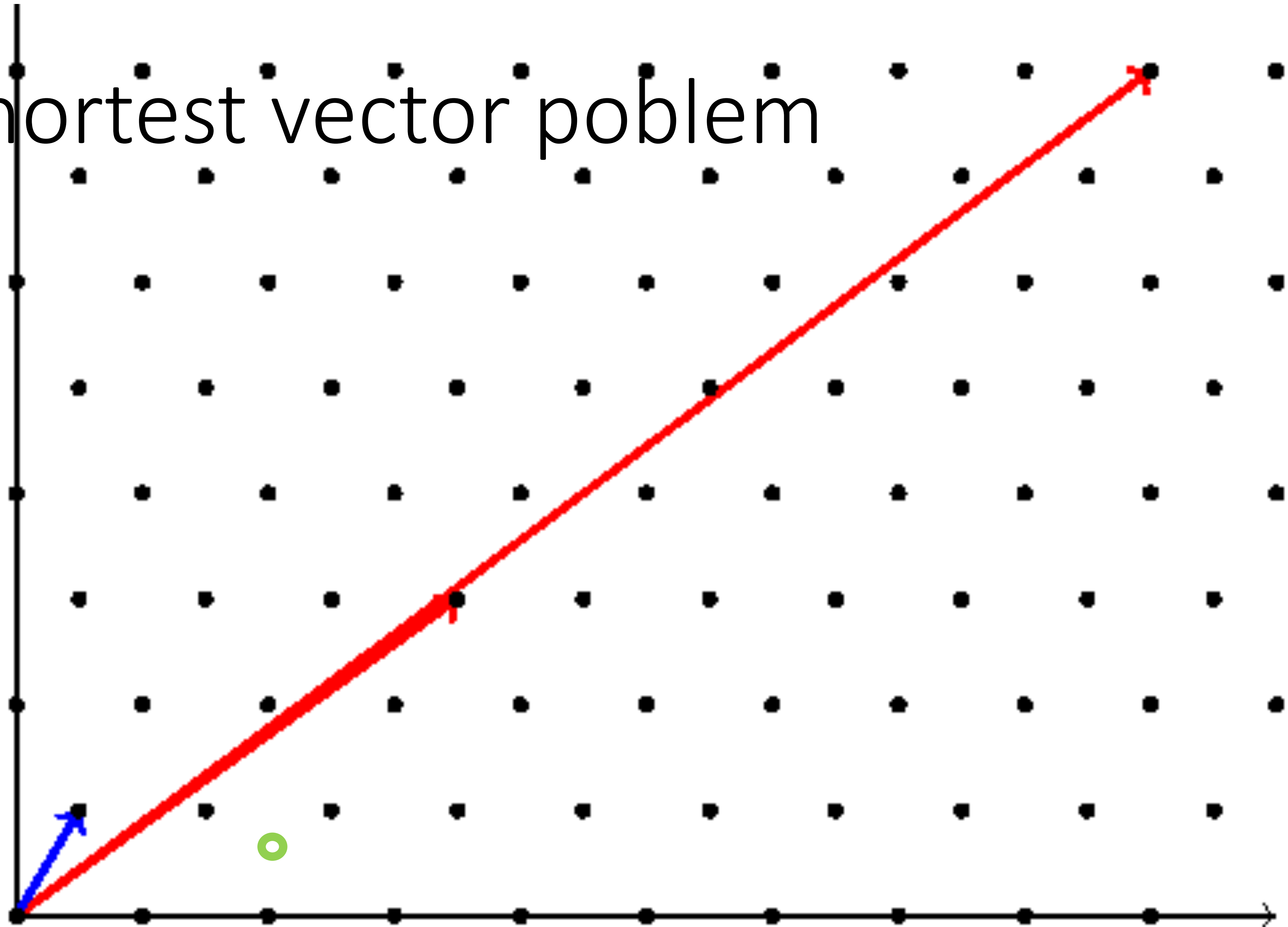
Lattice-based cryptography

- Flexible basis: many constructions possible
- Well-studied (by far the most published articles)
- Both Signatures, KEM's and much more…

- Idea: Given an arbitrary lattice, find the lattice point closest to a given point (CVP) or the shortest vector in the lattice (SVP).

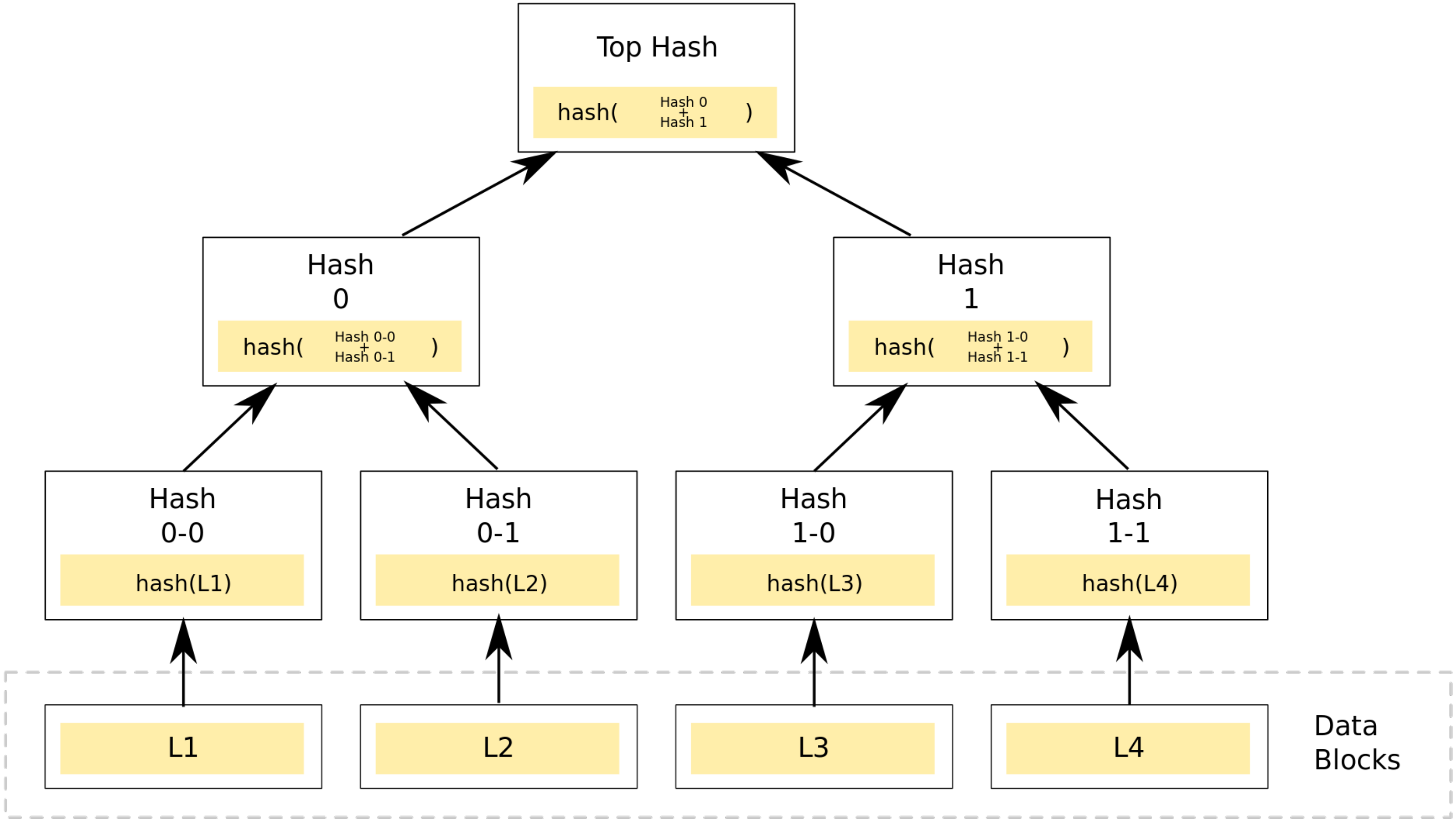
- The lattice is presented in an ugly basis. Reducing the basis to a practical form (LLL-algorithm) takes a lot of time.

Shortest vector poblem



Hash-based cryptography

- Only requires secure hash-functions
- Considered safe
- Only signature schemes
- Fast, but large signatures
- Stateful signature schemes (Merkle trees)



NIST standardization

- There is no perfect Post-Quantum candidate yet,
  but the threat of a Quantum computer is imminent.

- NIST standardization process (2016)

- Round 1: 59 KEM + 23 SIGN. [15 published attacks]
- Round 2: 17 KEM + 9 SIGN.
- Round 3 (July 2020 – Dec 2021):
  - Finalists: 4 KEM + 3 SIGN
  - Alternative candidates: 5 KEM + 3 SIGN

The remaining algorithms

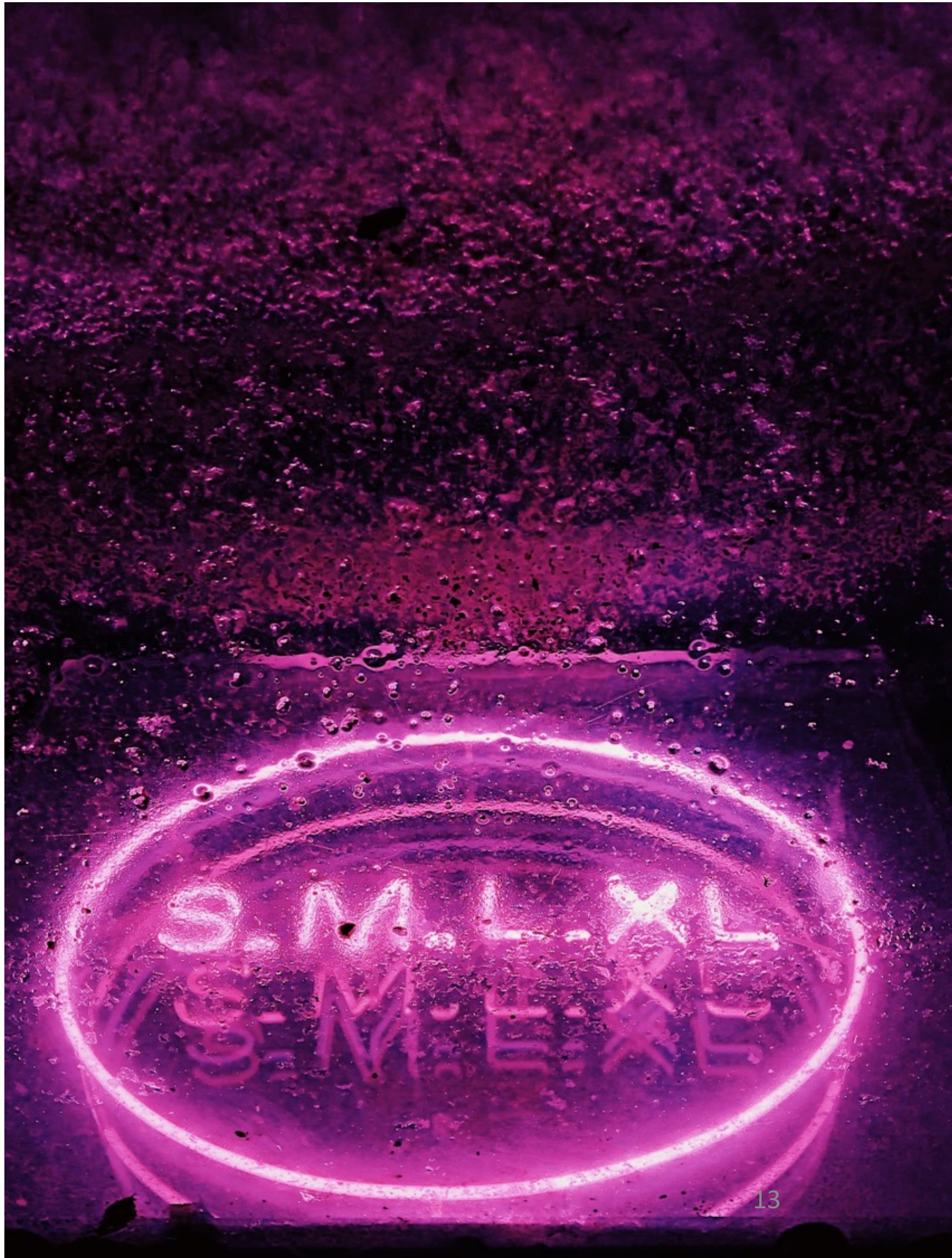
Algorithm	Approach	Private key	Public key	Signature	Status
Crystals-Dilithium-II	Lattice	2.8kB	1.3kB	2.4kB	Finalist
Falcon-512	Lattice	1.3kB	0.9kB	0.7kB	Finalist
Rainbow-I	Multivariate	101kB	158kB	64B	Finalist
Cyclic Rainbow-I	Multivariate	101kB	59kB	64B	Finalist
RedGeMSS-128	Multivariate	16B	375kB	36B	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: ~128 bits)

Applying PQC to DNSSEC

Restrictions of DNSSEC

- Key and Signature Size
- Validation Performance
- Signing Performance



Restrictions of DNSSEC

- **Key and Signature Size**
- Validation Performance
- Signing Performance

- > 1,232 bytes often cause fragmentation
- Larger records attractive for DDoS attacks

Restrictions of DNSSEC

- Key and Signature Size
- **Validation Performance**
- Signing Performance

- Resolvers can validate thousands of signatures per second

Restrictions of DNSSEC

- Key and Signature Size
- Validation Performance
- **Signing Performance**
  - On-the-fly signing most time critical

Requirements of DNSSEC

• **Signature Size:**	**≤ 1,232 bytes**
• **Validation Performance:**	**≥ 1000 sig/s**
• **Signing Performance:**	**≥ 100 sig/s**

Finding the Right Algorithm

Algorithm	**Public Key**	**Signature**	**Sign/s**	**Verify/s**
ED25519	32B	64B	~ 26,000	~8,000
RSA-2048	0.3kB	0.3kN	~1,500	~50,000

Finding the Right Algorithm

Algorithm	Public Key	Signature	Sign/s	Verify/s
Falcon-512	0.9kB	0.7kB	~ 3,300	~20,000
ED25519	32B	64B	~ 26,000	~8,000
RSA-2048	0.3kB	0.3kN	~1,500	~50,000

Finding the Right Algorithm

Algorithm	Public Key	Signature	Sign/s	Verify/s
Falcon-512	0.9kB	0.7kB	~ 3,300	~20,000
Rainbow-Ia	158kB	64B	~ 8,300	~ 11,000
ED25519	32B	64B	~ 26,000	~8,000
RSA-2048	0.3kB	0.3kN	~1,500	~50,000

Finding the Right Algorithm

Algorithm	Public Key	Signature	Sign/s	Verify/s
Falcon-512	0.9kB	0.7kB	~ 3,300	~20,000
Rainbow-Ia	158kB	64B	~ 8,300	~ 11,000
RedGeMSS128	375kB	36B	~ 540	~ 10,000
ED25519	32B	64B	~ 26,000	~8,000
RSA-2048	0.3kB	0.3kN	~1,500	~50,000

Preparing DNSSEC for PQC

- **Key and Signature Size**
- Validation Performance
- Signing Performance

- Increased TCP support
- Out of band key distribution

Preparing DNSSEC for PQC

- Key and Signature Size
- **Validation Performance**
  - Less frequent validation
- Signing Performance

Preparing DNSSEC for PQC

- **Key and Signature Size**
- **Validation Performance**
- **Signing Performance**

- Zone dependent algorithms

Next Steps and Conclusions

- Future developments may force us to reconsider our options/preferences
- New signing and key distribution approaches need to be better understood
- Keep in mind: *rolling* to a new algorithm *will take time [1]*

[1] https://dl.acm.org/doi/abs/10.1145/3419394.3423638

Next Steps and Conclusions

- Future developments may force us to reconsider our options/preferences
- New signing and key distribution approaches need to be better understood
- Keep in mind: *rolling* to a new algorithm *will take time*

Paper: https://ccronline.sigcomm.org/2020/ccr-october-2020/retrofitting-post-quantum-cryptography-in-internet-protocols-a-case-study-of-dnssec/