The Impact of Post-Quantum Cryptography on DNSSEC

DNSSEC and Security Virtual ICANN70 Workshop – 2021-03-24

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?

Post Quantum Cryptography

Quantum computing

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

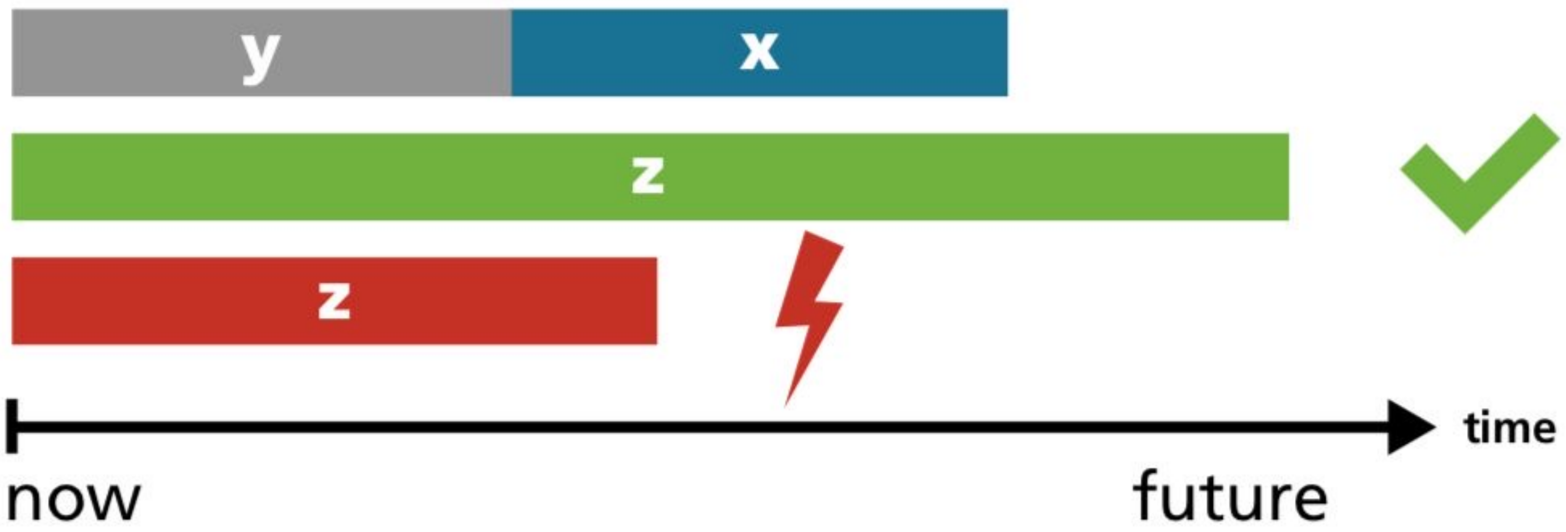
\- **All current public key cryptography must be replaced by a quantum-safe alternative!**

- DNSSEC's signature schemes must be replaced.

- When may this quantum computer be there:

\- Perhaps in the 2030's [Migration to quantum-safe cryptography, TNO, 2020]

Mosca’s inequality



x: time that secrets must remain secret

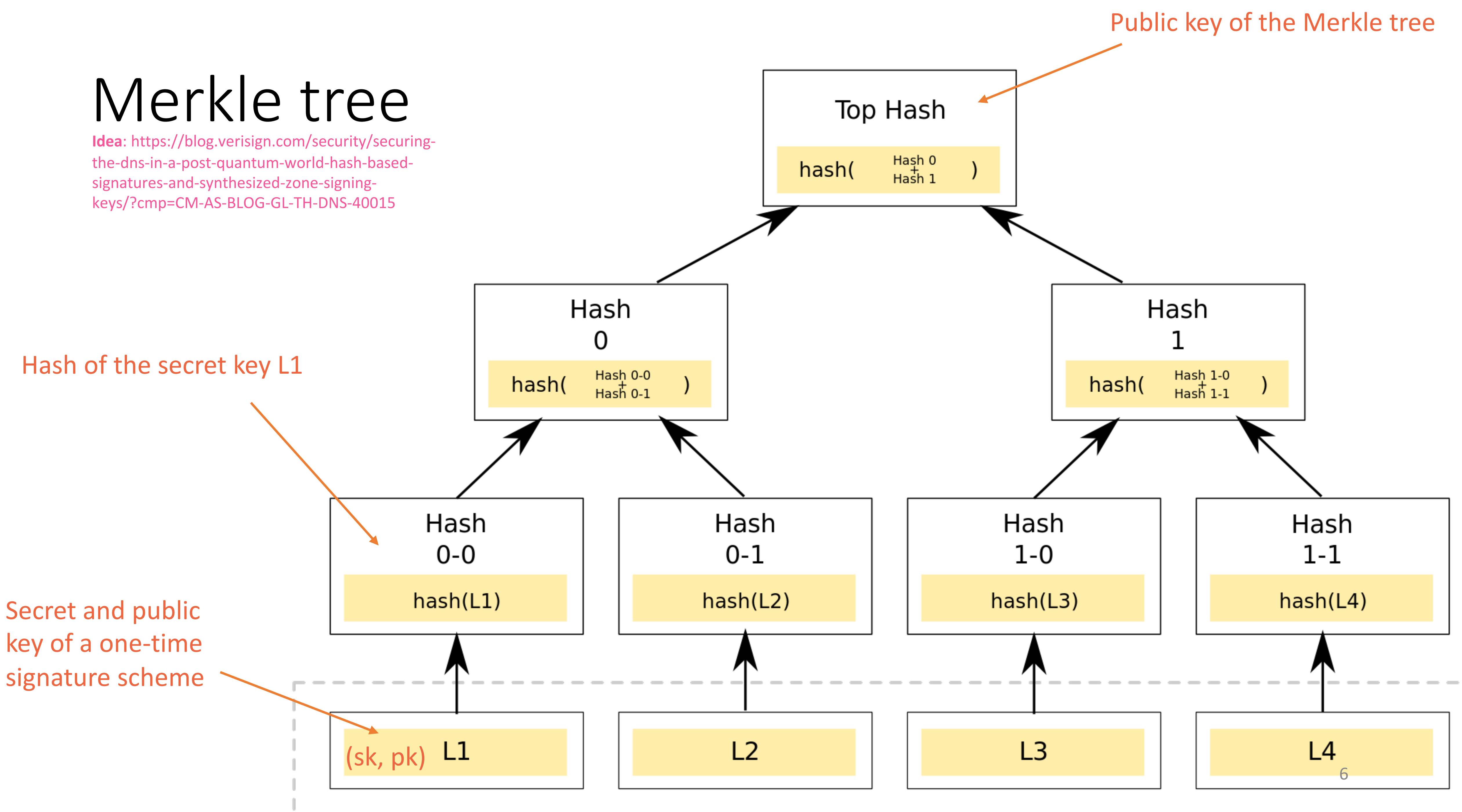
y: time it takes to deploy quantum-computer secure cryptography

z: time it takes until quantum computers break current cryptography

If z is larger than x+y, we are fine. If it is smaller, we are in danger!

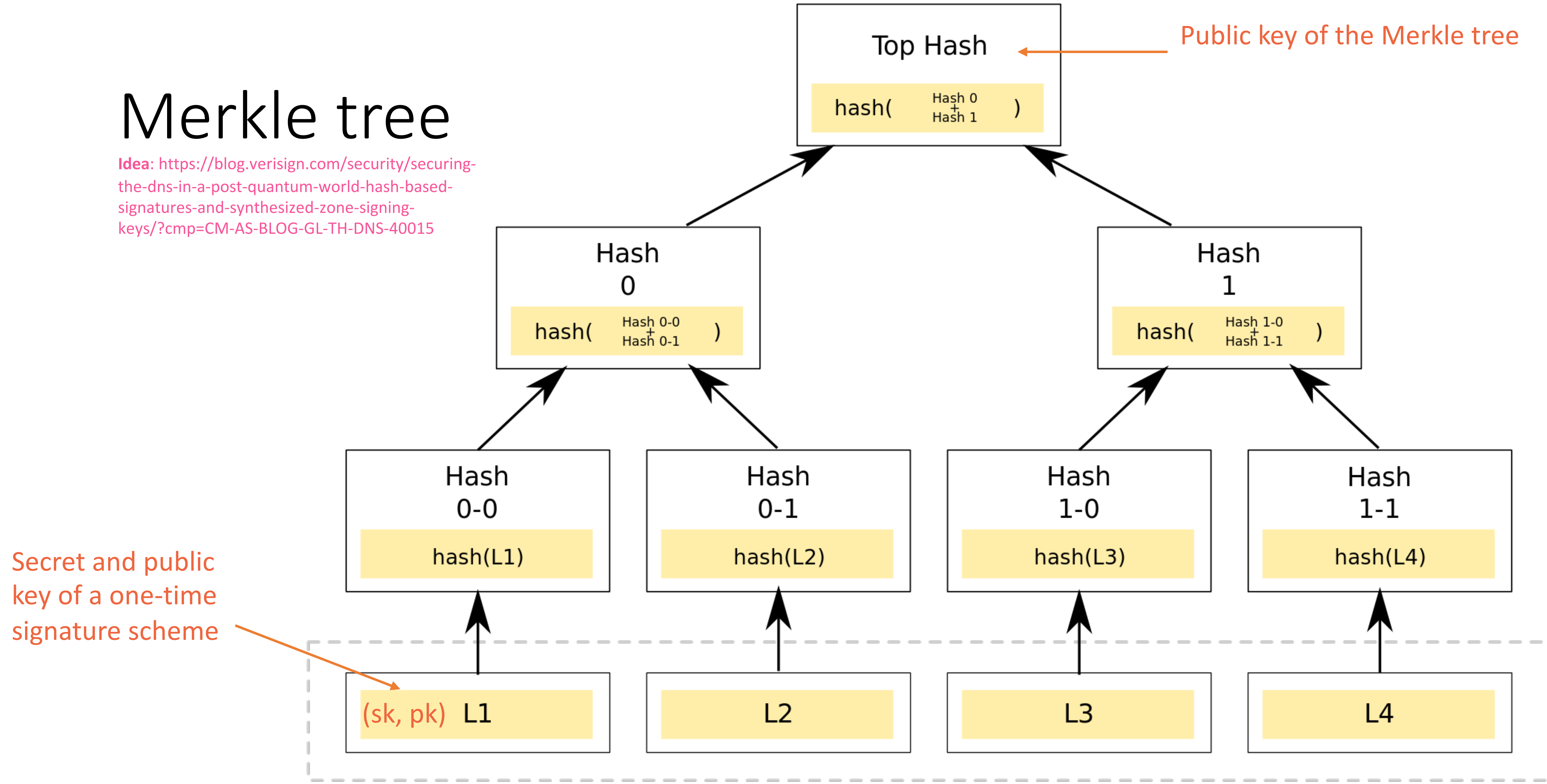
Merkle tree

Idea: https://blog.verisign.com/security/securing-the-dns-in-a-post-quantum-world-hash-based-signatures-and-synthesized-zone-signing-keys/?cmp=CM-AS-BLOG-GL-TH-DNS-40015



Merkle tree

Idea: https://blog.verisign.com/security/securing-the-dns-in-a-post-quantum-world-hash-based-signatures-and-synthesized-zone-signing-keys/?cmp=CM-AS-BLOG-GL-TH-DNS-40015



Signature of message m: (L1(m), pk, Hash0-1, Hash1)

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)

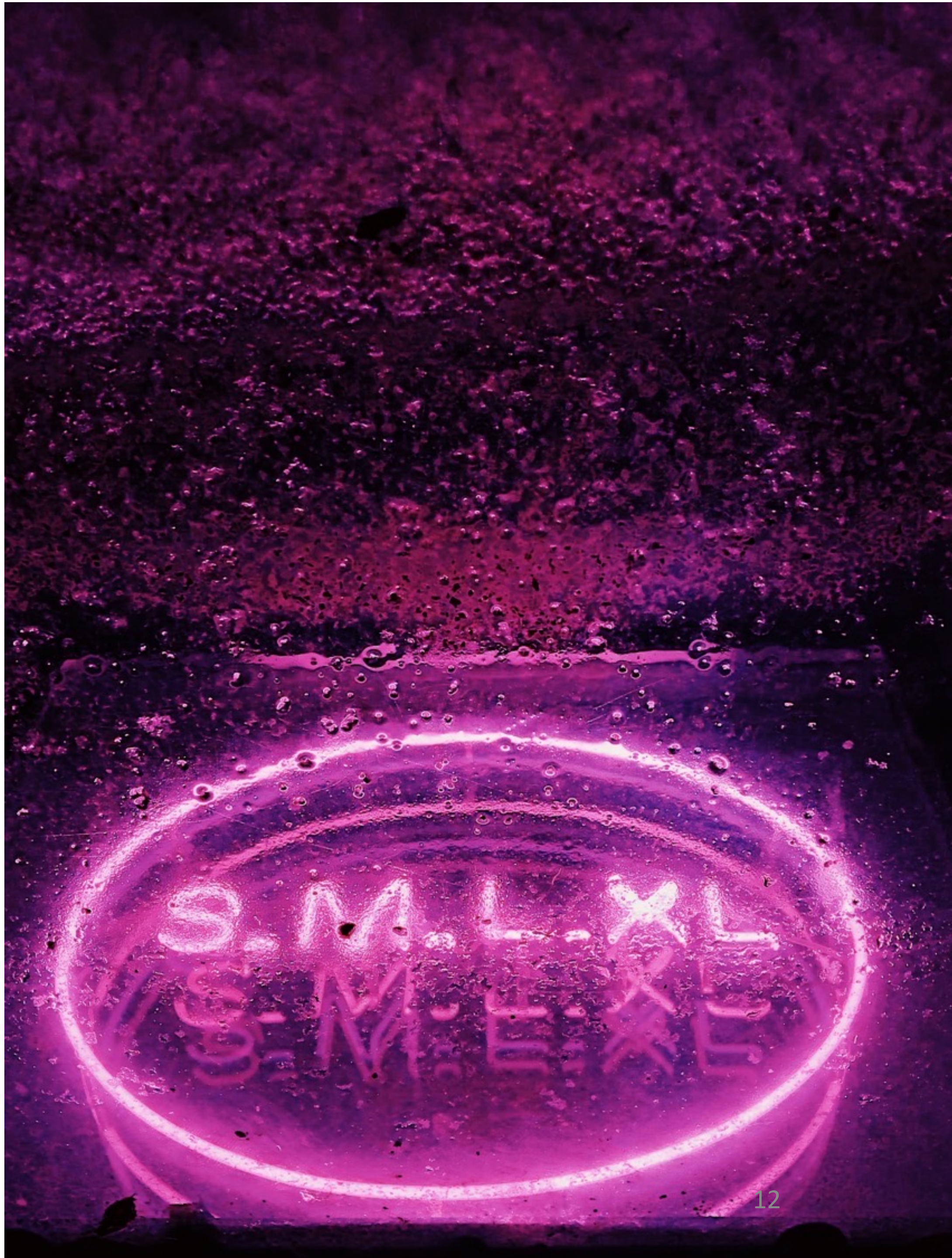
Developments

- Rainbow is not (yet) royalty-free.
- New (non-fatal) publications and attacks on the security of GeMSS and Rainbow.
- Lattice attacks may improve.
- NIST: Concern about the lack of diversity of the candidates.

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

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

Main Challenges

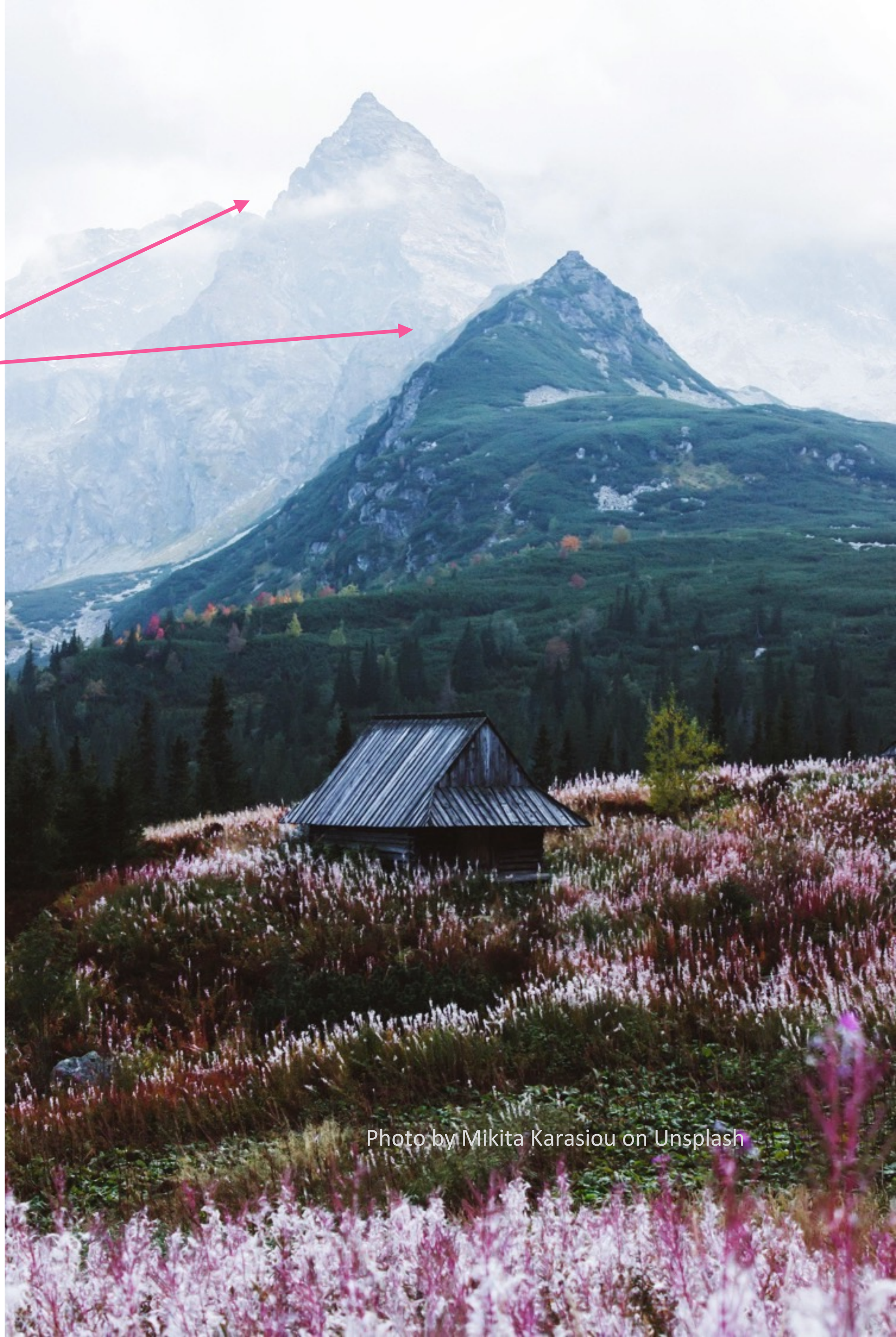
- Keys & Signatures > 1.232B
- Keys > 64kB

Main Challenges

- Keys & Signatures > 1.232B
- Keys > 64kB

Main Challenges

- Keys & Signatures > 1.232B
- Keys > 64kB



Possible Solutions

Photo by Rona Lao on Unsplash

- Keys & Signatures > 1.232B

- TCP fallback

\+ regular DNS

\- not everywhere supported

\- increased server requirements

Possible Solutions

Photo by Rona Lao on Unsplash

- Keys & Signatures > 1.232B
- TCP fallback

\+ regular DNS

\- *? not everywhere supported ? [1]*

\- *? increased server requirements ? [2]*

[1] https://blog.apnic.net/2020/12/14/measuring-the-impact-of-dns-flag-day-2020/

[2] L. Zhu, Z. Hu, J. Heidemann, D. Wessels, A. Mankin and N. Somaiya, "Connection-Oriented DNS to Improve Privacy and Security," *2015 IEEE Symposium on Security and Privacy*, San Jose, CA, USA, 2015, pp. 171-186, doi: 10.1109/SP.2015.18.

Possible Solutions


Photo by Rona Lao on Unsplash

- Keys > 64kB

- Splitting key in RRs

\+ modest DNS extension

\- additional round trips

\- higher risk of packet loss

Possible Solutions

- Keys > 64kB

- Splitting key in RRs

+ modest DNS extension
- additional round trips
- higher risk of packet loss

- Distributing key out of band

+ less prone to packet loss
- requires support of different protocol

Possible Solutions

- Keys > 64kB

- Splitting key in RRs
- Extending max DNS message size
- Distributing key out of band

+ Keys are not exchanged often

- Add to the "DNS Camel"

Next Steps and Conclusions

- Future developments may force us to reconsider our options/preferences
- Keep in mind: *rolling* to a new algorithm *will take time [1]*
- *Paper:* [*https://ccronline.sigcomm.org/2020/ccr-october-2020/retrofitting-post-quantum-cryptography-in-internet-protocols-a-case-study-of-dnssec/*](https://ccronline.sigcomm.org/2020/ccr-october-2020/retrofitting-post-quantum-cryptography-in-internet-protocols-a-case-study-of-dnssec/)

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