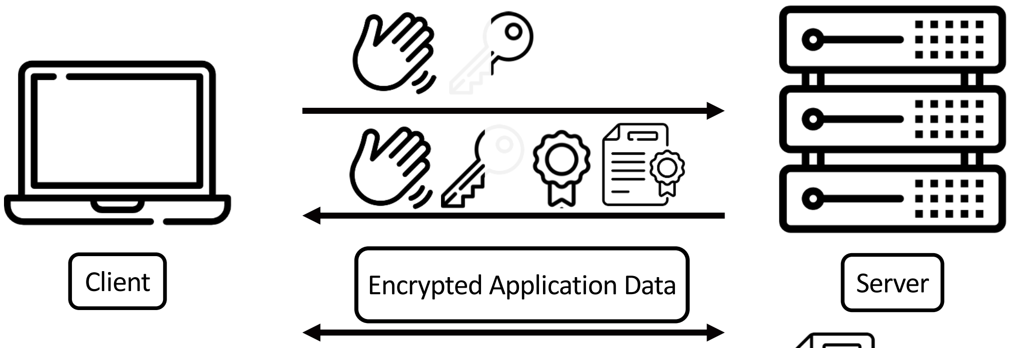
Batch me if you PQ-Sign
-- 7th NIST PQ Seminar

Carlos Aguilar-Melchor SANDBOXAQ™
Martin R. Albrecht SANDBOXAQ™
Thomas Bailleux SANDBOXAQ™
James Howe SANDBOXAQ™
Andreas Hülsing TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY
David Joseph SANDBOXAQ™
Marc Manzano SANDBOXAQ™

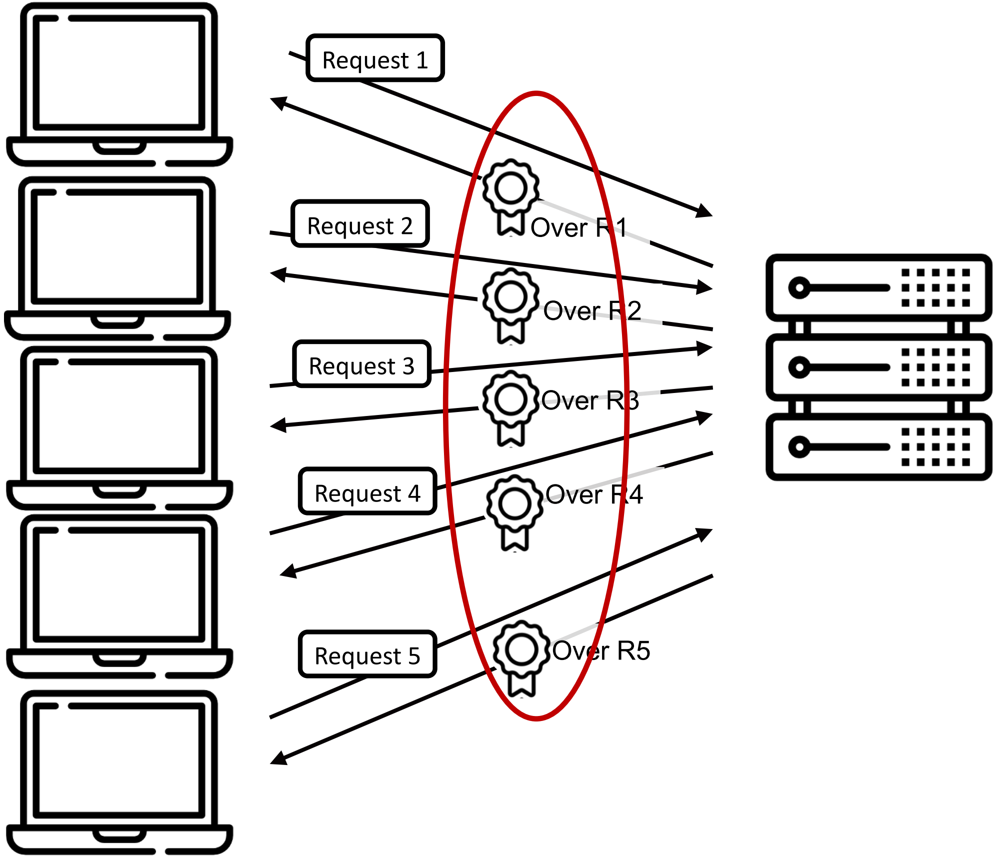
Nina Bindel
SANDBOXAQ™
@NinaBindel
nbindel

TLS – Quick Explainer



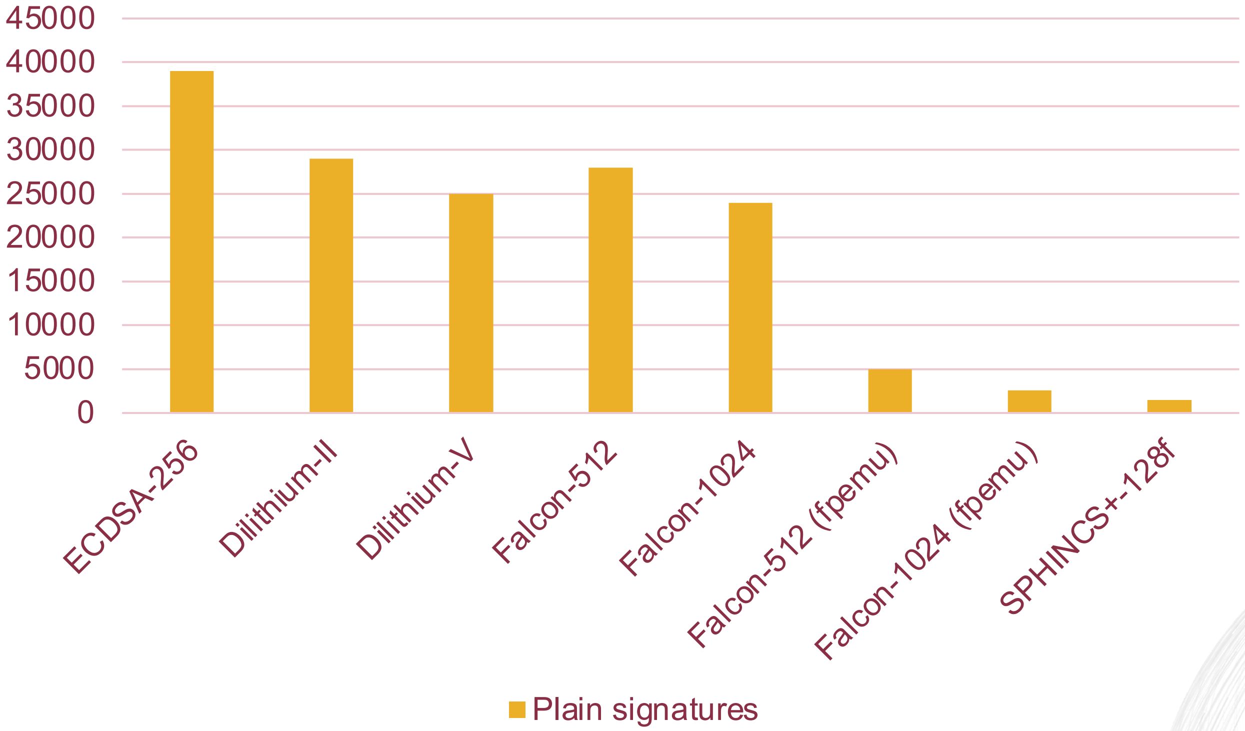
Server’s certificate including the server’s verification key

Generate signature over the transcript using the server’s signing key



Signature generation might become the computational bottleneck of TLS connections.

Number of TLS Handshakes per Second



Higher computational cost of PQ signatures severely impacts the ability of systems to scale and might inhibit their migration to PQC, especially in higher-throughput settings.

SANDBOXAQ™

Batch Signing to the Rescue

IETF Internet Draft. *Batch Signing for TLS*. David Benjamin. 2020.

- Uses Merkle trees to decrease number of signature generations needed to ease scalability for classical signature standards

- Particularly useful when used with **PQ signatures**
- We show how to apply it to applications **beyond the intended TLS** use case and to also **decrease communication cost** in additional to computation costs

SANDBOXAQ™

Outline

Exemplified using TLS

Batch Signing Scheme Using Merkle Trees

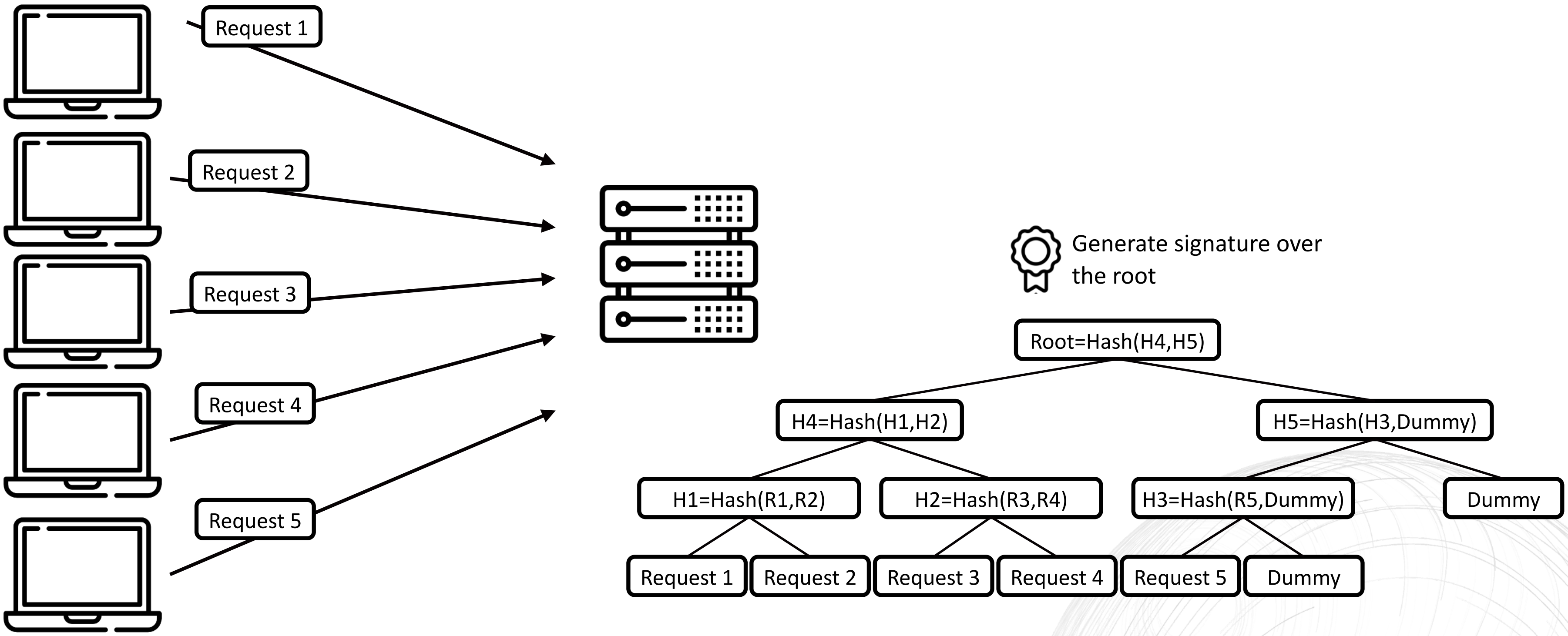
- Construction Idea
- Security and Privacy Guarantees
- Experimental Results

Use-Cases to Decrease Computation and Communication Cost

Using Batch Signing for TLS

Main Idea: Using Batch Signing for TLS

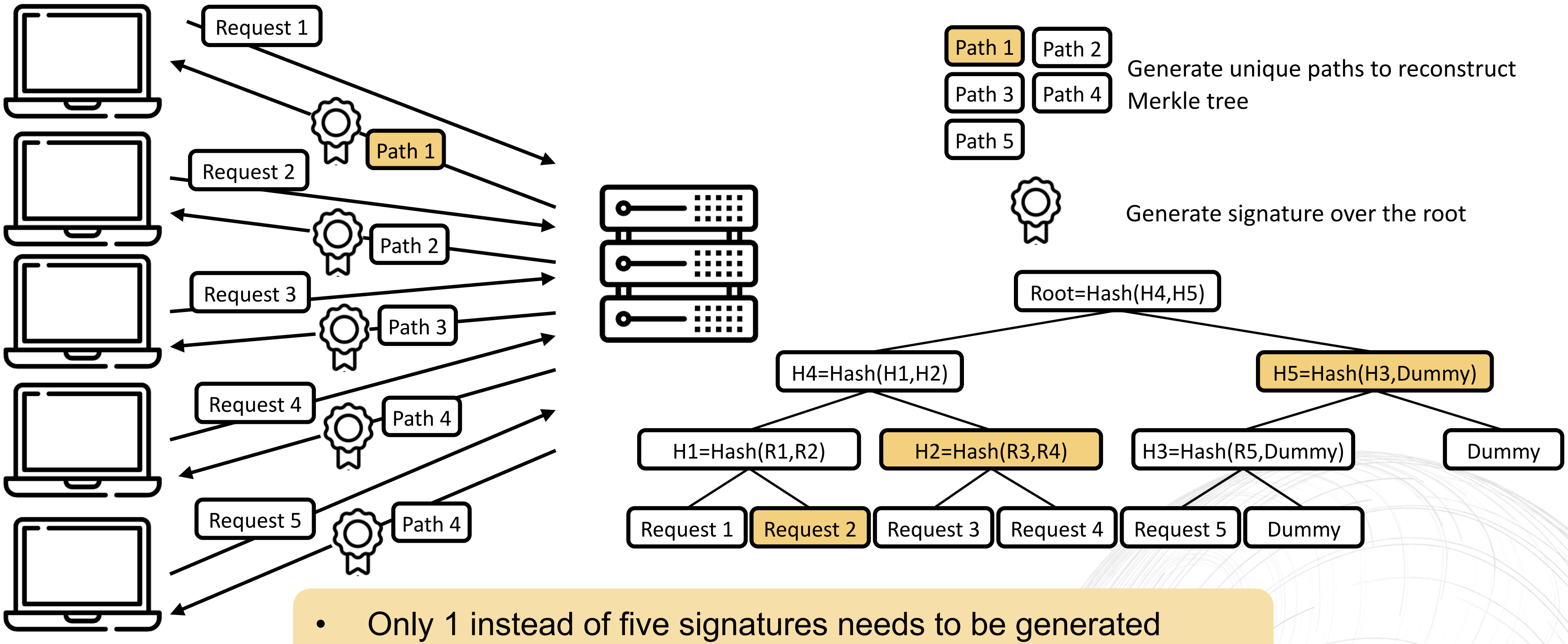
IETF Internet Draft. *Batch Signing for TLS*. David Benjamin. 2020.



SANDBOXAQ™

Main Idea: Using Batch Signing for TLS

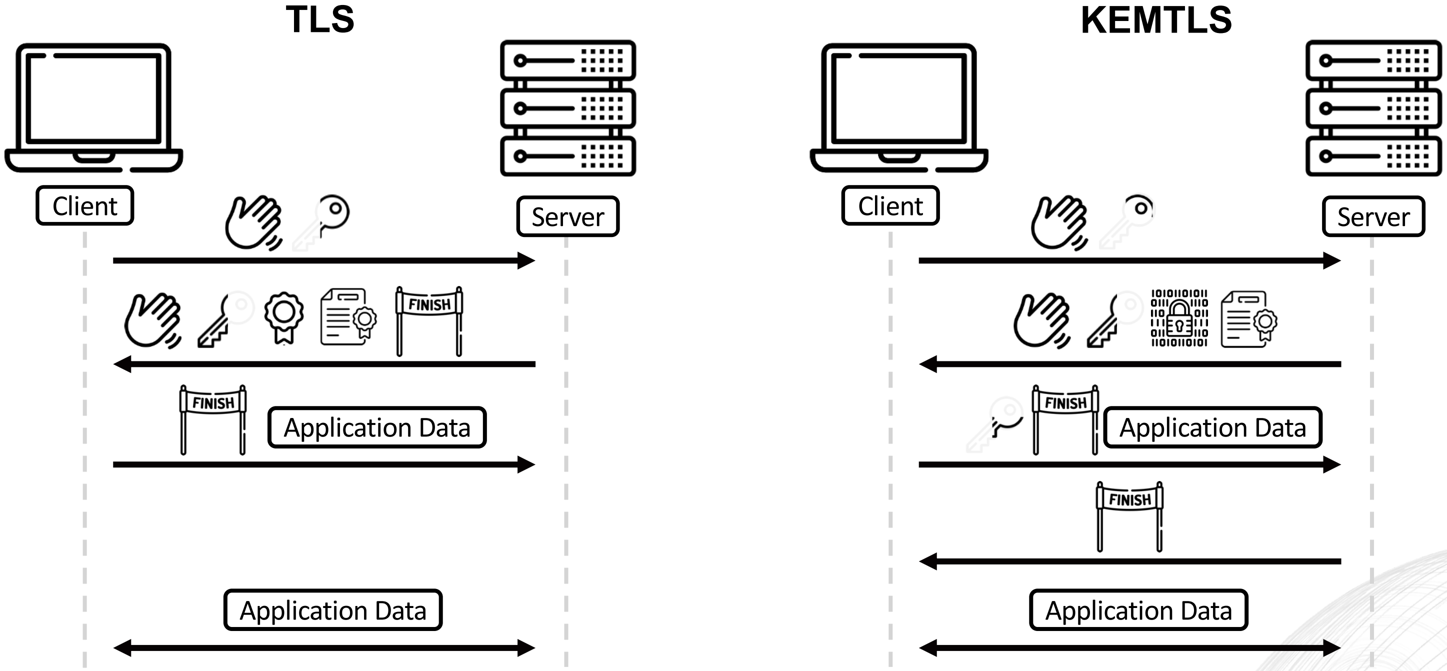
IETF Internet Draft. *Batch Signing for TLS*. David Benjamin. 2020.



- Only 1 instead of five signatures needs to be generated
- At the cost of small increase of the signature size and latency

SANDBOXAQ™

Alternative to KEMTLS



KEMTLS replaces static server authentication with a static KEM, so that only the involved KEM public keys need to be signed rather than the transcript.

- KEMTLS leads to a higher throughput with no latency increase.
- KEMTLS needs a number of significant infrastructure changes.

SANDBOXAQ™

Security and Privacy Guarantees

Security Guarantees

Batch signatures are unforgable if the signatures computed over the Merkle tree root are **unforgable** (EUF-CMA) and the (tweakable) hash function used to build the Merkle tree **is target collision resistant** (SM-TCR).

Essentially the same as plain signature unforgability

- Improvement over IETF Internet Draft that required a collision-resistant hash function.
- Leads to decreased authentication paths (half the size).
- SM-TCR is a fundamentally weaker assumption.

Privacy Guarantees

Batch Privacy

Given two signatures, an adversary cannot decide whether they were signed in the same batch.

- We can't achieve that because of the shared root signature throughout the batch.
- Adversary learns two connections were
  - ➡ **in a certain time window**
  - ➡ to the same client

- Known because of same client certificate in many applications
- True also for BPrivate schemes

Weak Batch Privacy

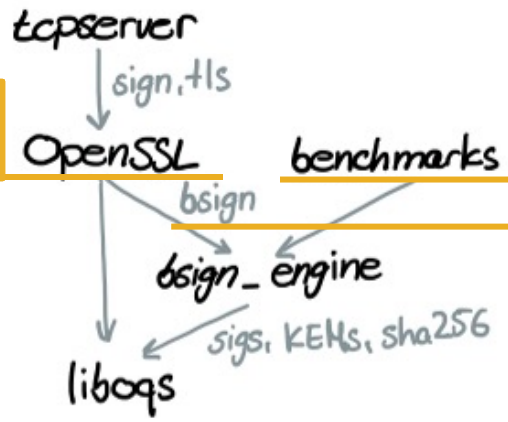
Signatures from the same batch do not leak anything about a message for which no signature is made available

Batch signatures offer 'weak batch privacy' if the hash function used to build the Merkle tree is a one-time pseudo-random function.

Experimental Results for TLS

Experimental Setup

OpenSSL fork making use of liboqs (PQ algorithms) and ring (ECDSA)



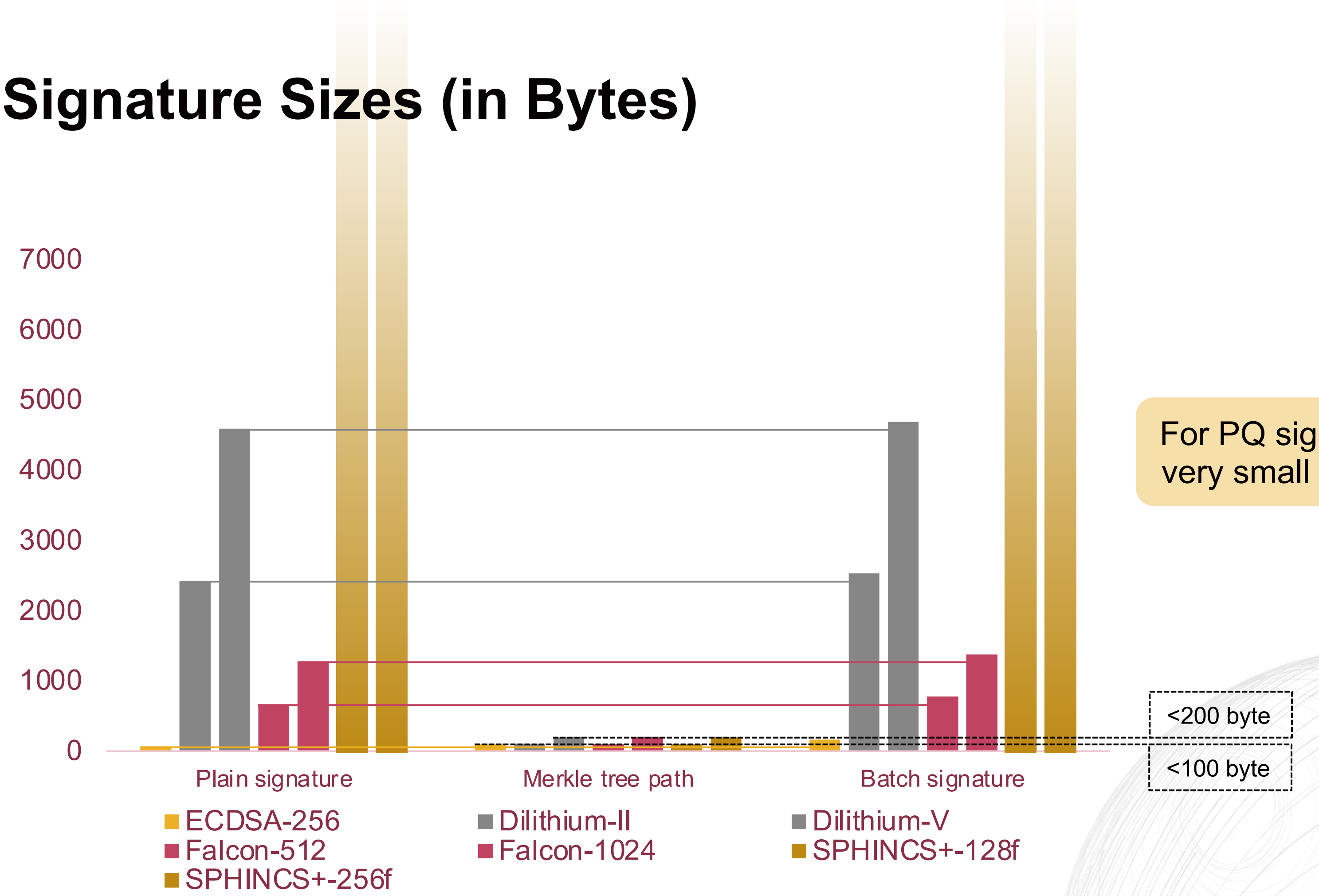
The high-level architecture and dependencies of our batch signing TLS experiments

- Between 1-4 clients and 1 server
- Batch size of 16 or 32 (to optimize latency)
- Averaged over 20 sec
- Google Cloud C2 instance with Intel 3.9 GHz Cascade Lake processor

RUST implementation of batch signing, including building the Merkle tree and batch signing functionalities

SANDBOXAQ™

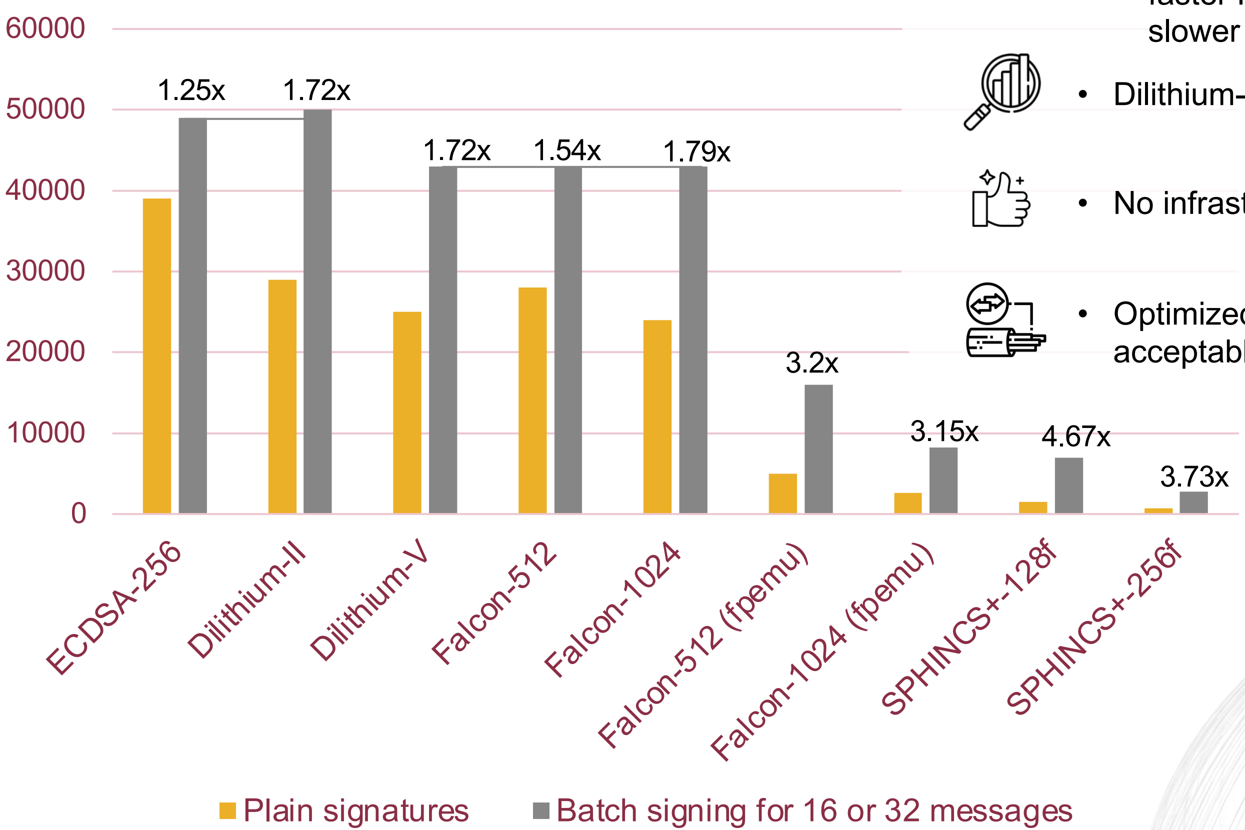
Signature Sizes (in Bytes)



For PQ signature schemes: very small relative increase

SANDBOXAQ™

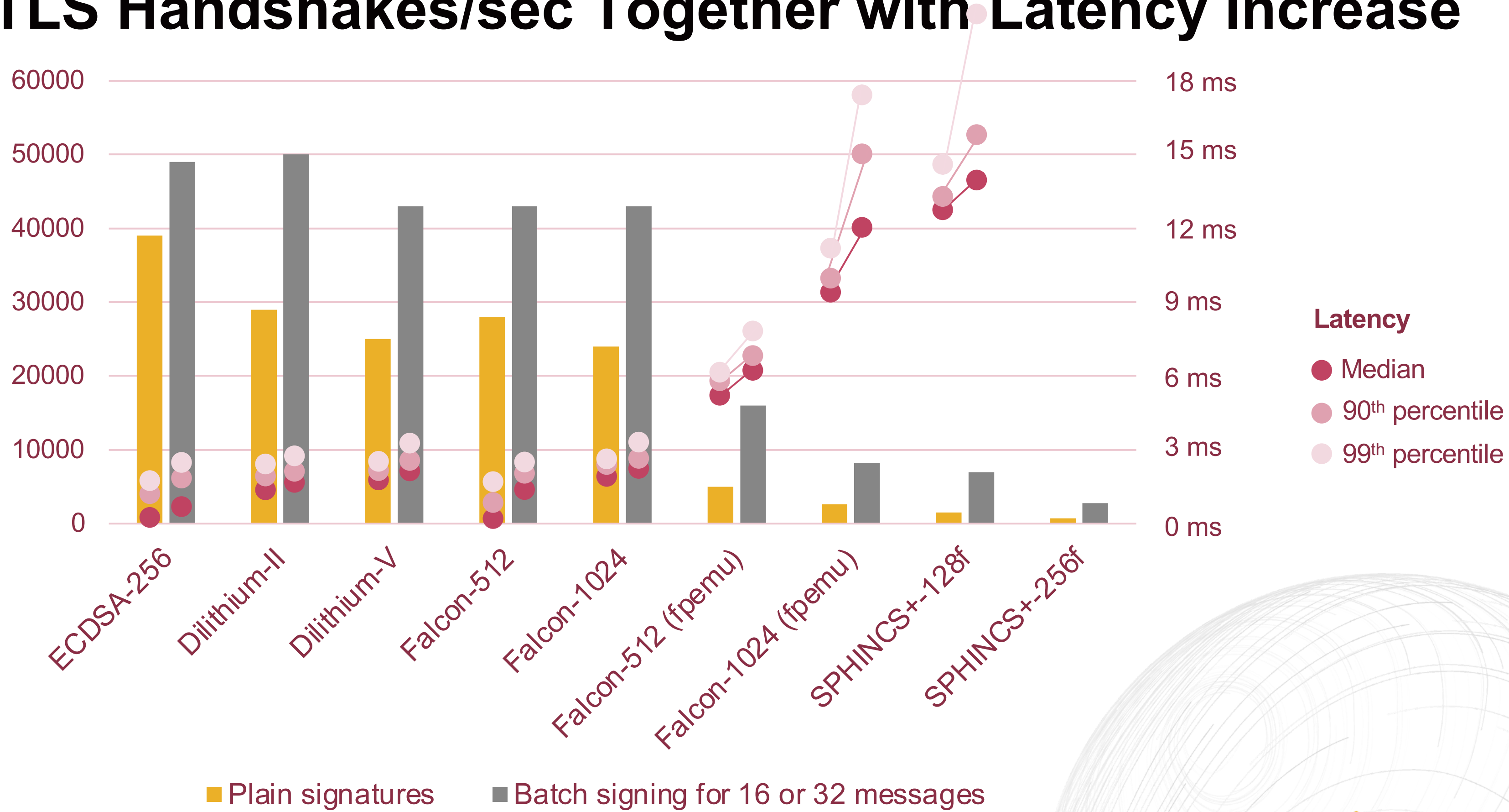
#TLS Handshakes/sec



- Speed-up for
  faster PQ algs ~1.5x-2.0x
  slower PQ algs ~4x
- Dilithium-II more handshakes than ECDSA
- No infrastructure changes, just client update
- Optimized for max #handshakes and acceptable latency delay

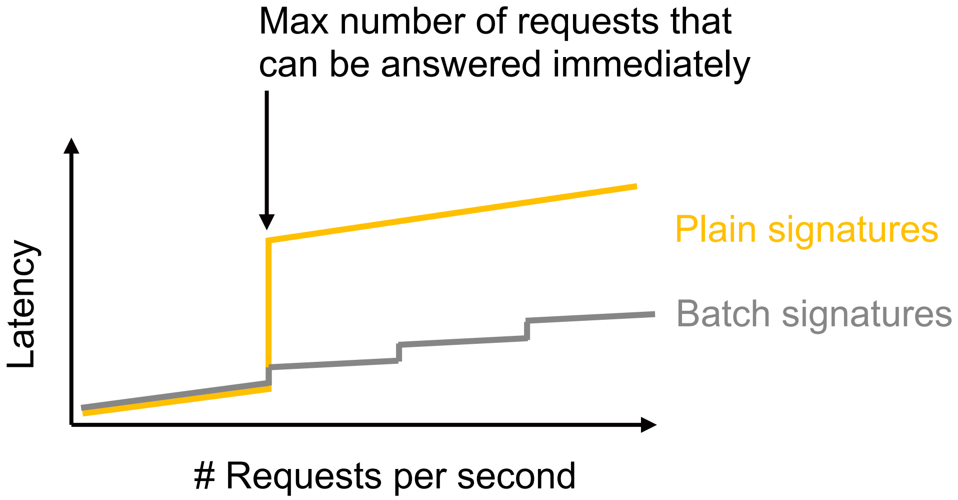
SANDBOXAQ™

#TLS Handshakes/sec Together with Latency Increase



SANDBOXAQ™

Latency comparison



- Flexibility to react to number of incoming request by adjusting tree size
- Slower increase of latency with increased number of requests after capacity of server's signature generations is reached.

Use-cases Beyond TLS to Decrease Communication and Computational Cost

Reducing Computational Cost

✓

Hardware Security Modules (HSMs)

- Generate large sets of (short-lived) certificates
- HSMs are significantly slower than traditional CPUs:

~10 000 sig/sec modern commodity CPUs	vs	~100 sig/sec enterprise-grade HSM

- Refrain from giving performance comparison, since PQ HSMs are not available yet. Therefore, a performance comparison would not reflect reality.
- Under assumption that certificate requester is able to verify batch signatures

Existing Proposals to Use Batch Signing to Decrease Communication Cost

Using stateful signatures -- Fregly, Harvey, Kaliski Jr., and Sheth. 2022. *Merkle Tree Ladder Mode: Reducing the Size Impact of NIST PQC Signature Algorithms in Practice*. ePrint 2022/1730.

- Using Merkle trees transform any signature scheme into a stateful signature scheme with compressed signature size

Using stateless signatures – IETF Draft. Benjamin, O'Brien, and Westerbaan. 2023. *Merkle Tree Certificates for TLS.*

- Introducing a new certificate format to decrease the signature/certificate/communication size in TLS

In comparison, the presented batch signing approach does not need infrastructure changes and can be offered by the signer as one signature scheme that can be negotiated by the requester.

SANDBOXAQ™

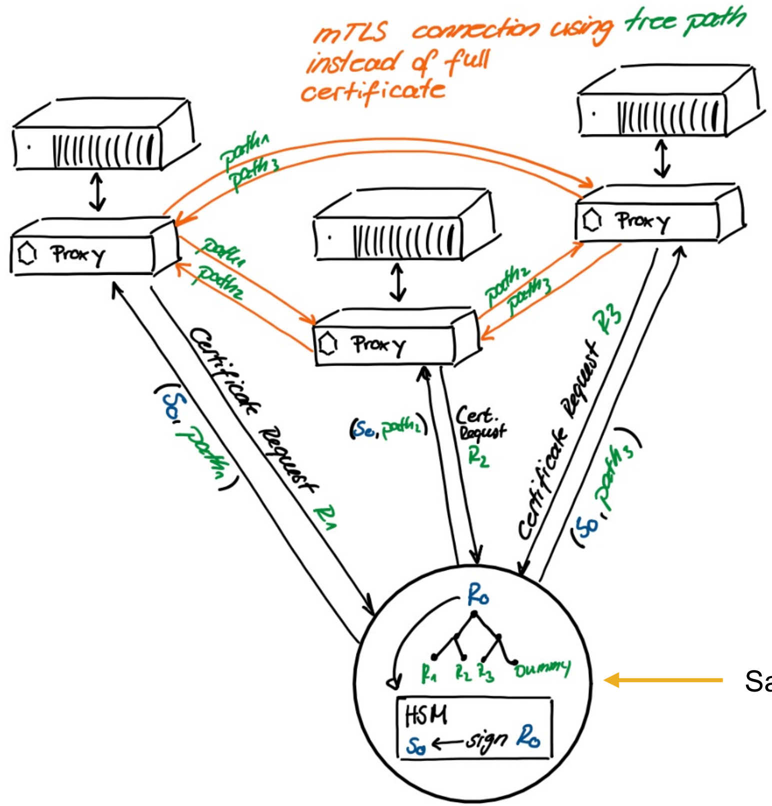
Reducing Communication Cost

Generally, use-cases where verifiers are aware that batch signing is used and in which batch their request is.

During TLS communications in which clients communicate with servers from the same batch, the certificate could consist of the unique path as signature as long as the full certificate with path *and* root signature is communicated **once**.

The HSM can **broadcast** root signature and drop it from the individual batch signatures.

Use-case: Certificate Renewal in Mutual TLS (mTLS) Mesh



- Saving **communication cost** within the mesh network
- Reduced **storage** requirements for certificates

Saving **computational cost**

SANDBOXAQ™

Summary

- Resurrection of IETF Internet Draft. *Batch Signing for TLS*. David Benjamin. 2020.
- Provision of security and privacy foundation using the reductionist approach
- Study performance trade-offs for TLS:
  - Increased throughput for PQ algorithm of 1.5x – 4.6x
  - Under increased signature size of less than 200 byte
  - Under acceptable latency increase of at most 25% or 0.5 ms
- Suggest and explain use-cases beyond TLS

IACR eprint 2023/492

Thank you!

We are looking for interns/ residents. Check out sandboxaq.com/careers

All icons from flaticon with premium.

@NinaBindel

nbindel

SANDBOXAQ™