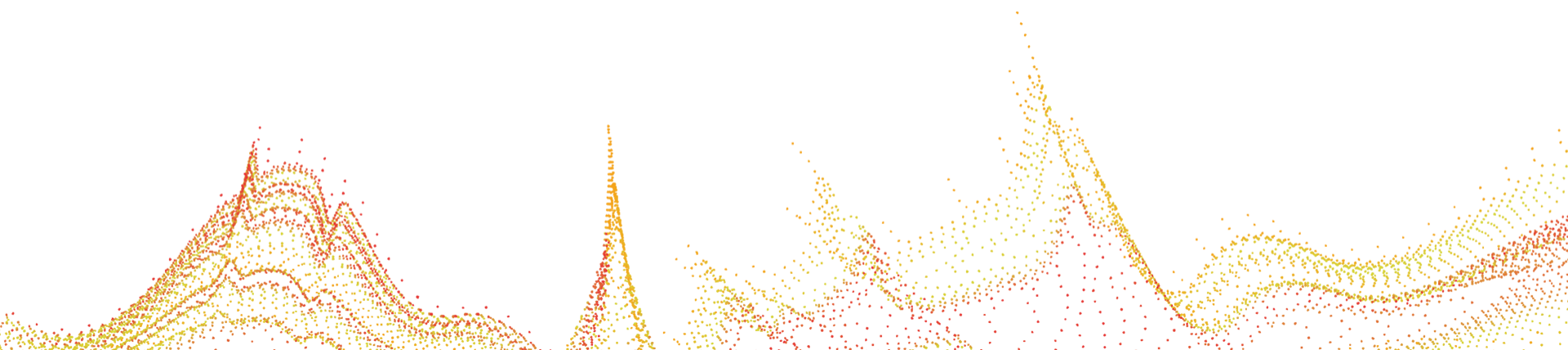


# How are we going to afford post-quantum authentication?

Thom Wiggers





# Who am I?

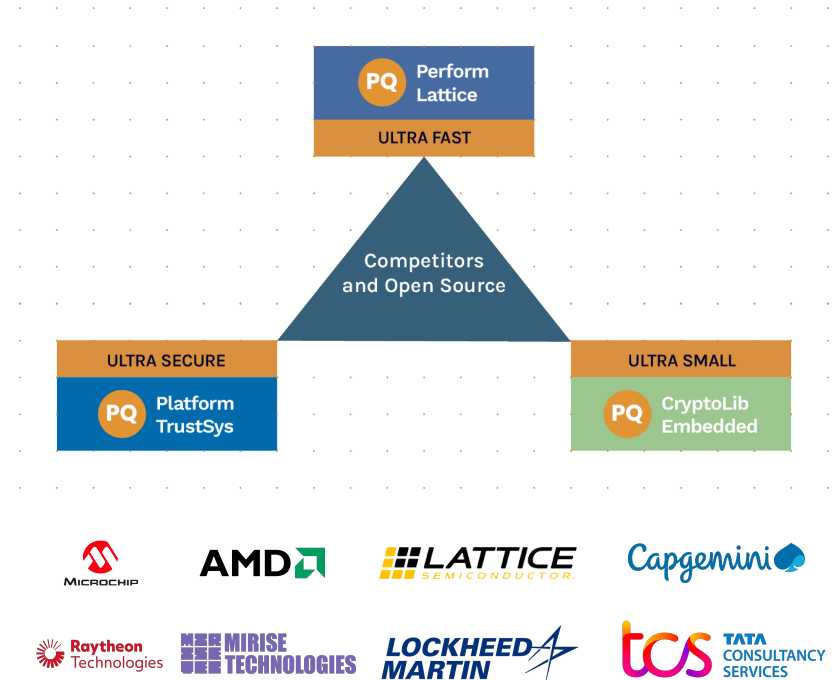
- **Thom Wiggers**
- Cryptography Researcher at **PQShield**
- Research Interest: **Applying PQC to Real-World Systems**
- PhD (with distinction) from Radboud University on **Post-Quantum TLS** (2024)
- Based in The Netherlands





# PQShield: global post-quantum cryptography experts

- World leaders in PQC with an 80+ strong team in UK, US, EU and Japan, **incl. 50+ Specialist PQC Cryptographers and Hardware & Software Engineers**
  - Team PQShield co-authored all of the new PQC NIST standards, and have created 85 peer reviewed papers and 30+ Patents to date
  - Contributors to bodies like RISC-V, IETF, ETSI, WEF, GSMA+
  - Leading authority on real-world PQC implementation in software, FPGA and ASIC - including our own silicon test chip driving SCA and FIA resistance techniques, FIPS 140-3 certification and much more
  - Deep specialism in optimisations for speed, size or security
  - PQShield licenses high quality PQC software IP and hardware IP to the global secure products supply chain. For example:





# PQ is going great

Cloudflare, Google Chrome, Microsoft Edge, and Firefox deployed PQ key exchange in web browsing.



## Connection - **secure connection settings**

The connection to this site is encrypted and authenticated using QUIC, X25519MLKEM768, and AES\_128\_GCM.



# PQ is going great

Signal and Apple deployed PQ key exchange in messaging

## Quantum Resistance and the Signal Protocol

[ehrenkret](#) on 19 Sep 2023

February 21, 2024

## iMessage with PQ3: The new state of the art in quantum-secure messaging at scale

Posted by Apple Security Engineering and Architecture (SEAR)





# PQ is going great

Zoom offers PQ key exchange in video calls

**Zoom bolsters security offering with the inclusion of post-quantum end-to-end encryption in Zoom Workplace**

Post-quantum E2EE now available for Zoom Meetings, making Zoom the first UCaaS provider to offer the new security feature

Published May 21, 2024





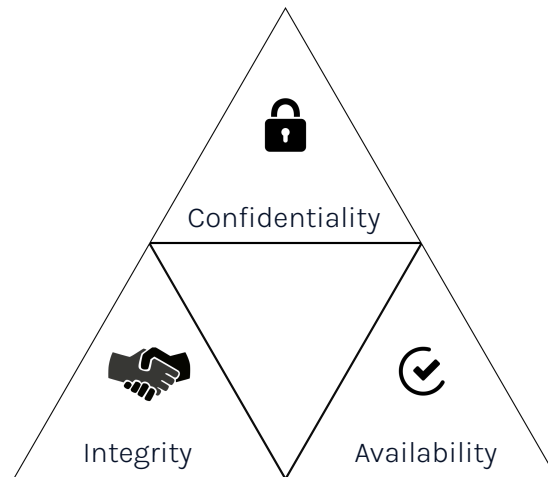
# What the headlines are not telling you

All of these examples: only PQ key exchange

Apple: “We will **continue to assess** the need for post-quantum authentication to thwart such attacks.”

Signal: “Further research in the area of post-quantum cryptography will be needed to **fill in the remaining gaps.**”

Cloudflare: “Over the coming years, we’ll be working with browsers to **test the viability** and performance impact of post-quantum authentication in TLS.”





# Why the focus on PQ key exchange?

- Confidentiality is **strongest link**
  - Only needs to be updated in 1 spot: **much easier**
- ML-KEM is *small-ish* and very fast
- “Harvest-Now-Decrypt-Later” makes it urgent

We are right to focus on PQ key exchange first!

But PQ Authentication will be more complicated, take more effort, and more time.

We should start now.



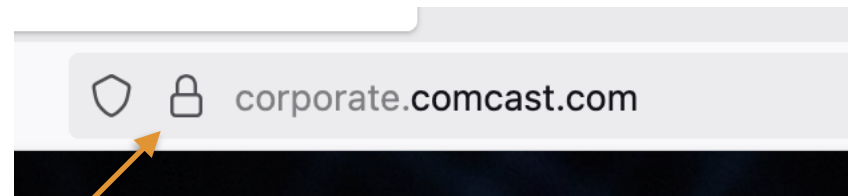
# More data = more slow

- Google reported that adding ML-KEM **slows down TLS by 4%**
- This only added 2kB to the client and server messages
- Google requires to **stay under 10%** slowdown
  - They estimate a <7 kB budget



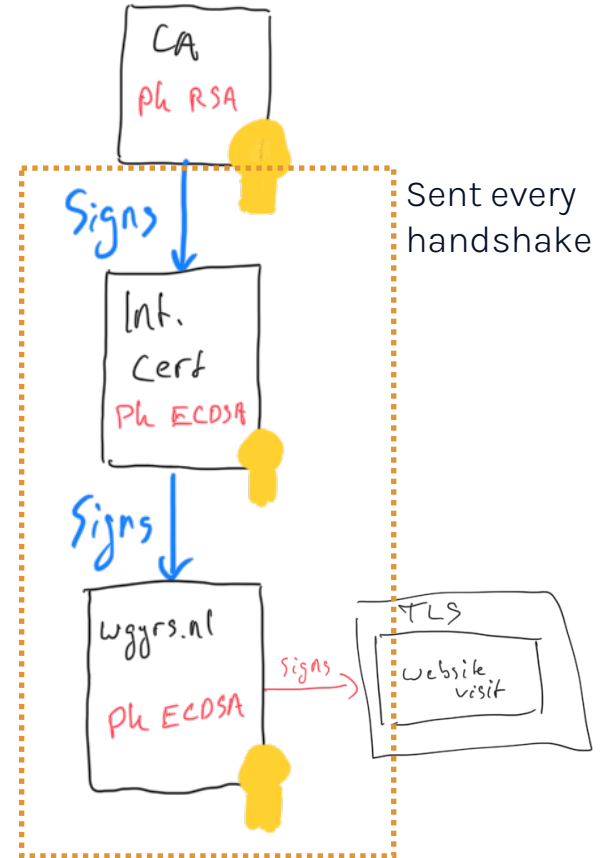
## Case study: TLS

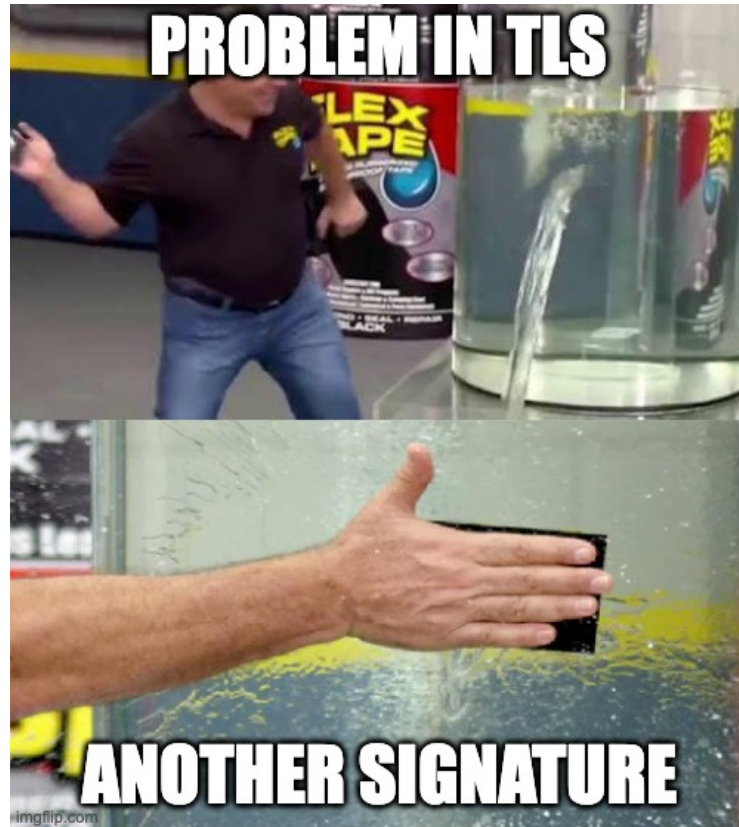
- “The lock in the browser”
- Web browsers run on powerful, well-connected devices
- Probably most-used cryptographic protocol
- If we can't transition TLS, what can we?



# Signatures don't come alone

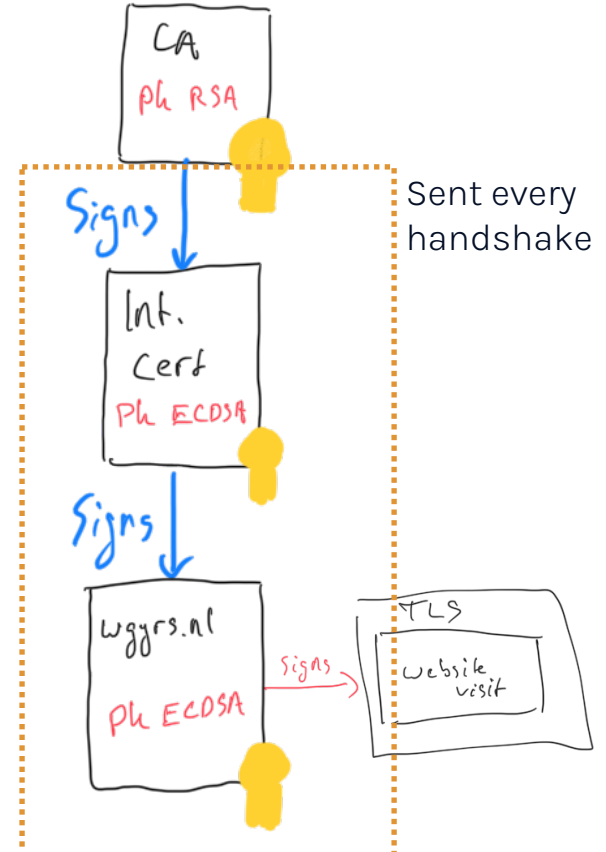
- Authentication “Chain” to link the website to the issuers that you trust
- Turns 1 signature into 3 signatures and 2 public keys





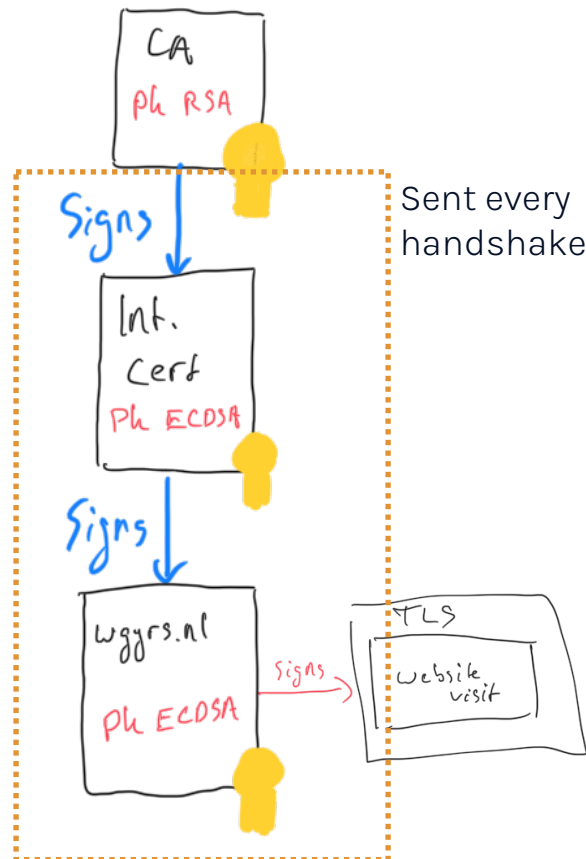
# Case study: TLS

- TLS sends a lot of certificates every time
- These certificates contain further signatures for Certificate Transparency and certificate revocation
- Typical **web** TLS handshake:
  - handshake signature
  - leaf certificate:
    - pk
    - + signature by intermediate CA crt
    - + OCSP staple
    - + 3x SCT
  - intermediate CA certificate:
    - pk + signature by root CA
  - root certificate (preinstalled)



# The cost of PQ authentication

- An RSA public key + signature require ~0.5kB
- ML-DSA-44 requires 3732 bytes for the same
- This means that e.g. TLS overhead increases by up to ~18kB





## Point of View: website operator

### PQ key exchange:

- Solves Harvest-Now-Decrypt-Later
- Only need to update in 1 spot
- Well-tested now
- Adds ~2kB of data to handshake
- ~4% slowdown for most clients is costly but acceptable (src: Google)

### PQ authentication:

- Will only protect after quantum apocalypse
- Requires updating certificate infra
- Dependencies on suppliers and ecosystem
- ~4% slowdown for ~2kB means **big slowdown** for ~18kB

“I’ll wait”



imgflip.com

“But what about the...

# **Request for Additional Digital Signature Schemes for the Post-Quantum Cryptography Standardization Process**

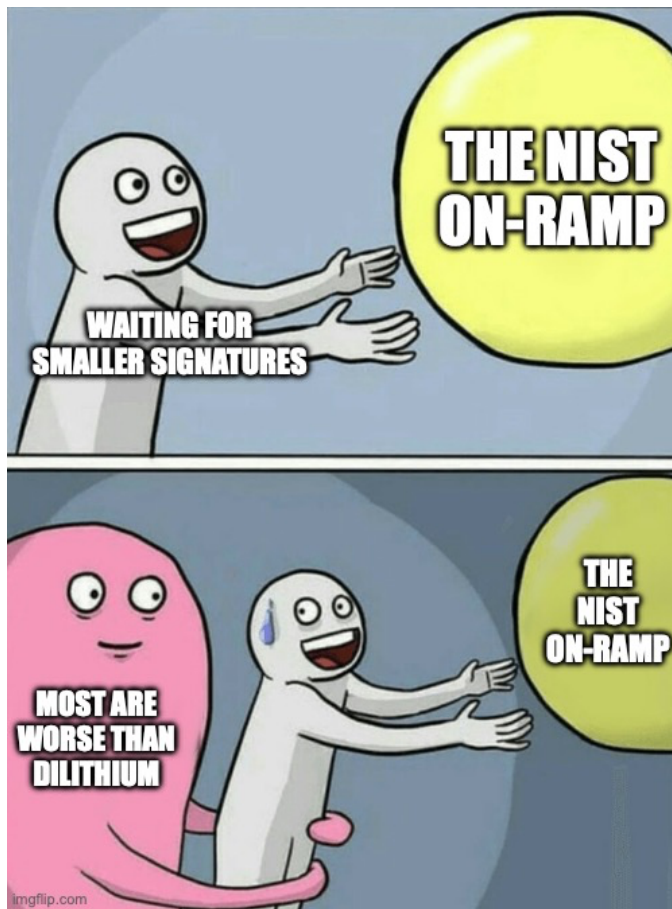
September 06, 2022



# 2024-10-24 NIST round 2 selections

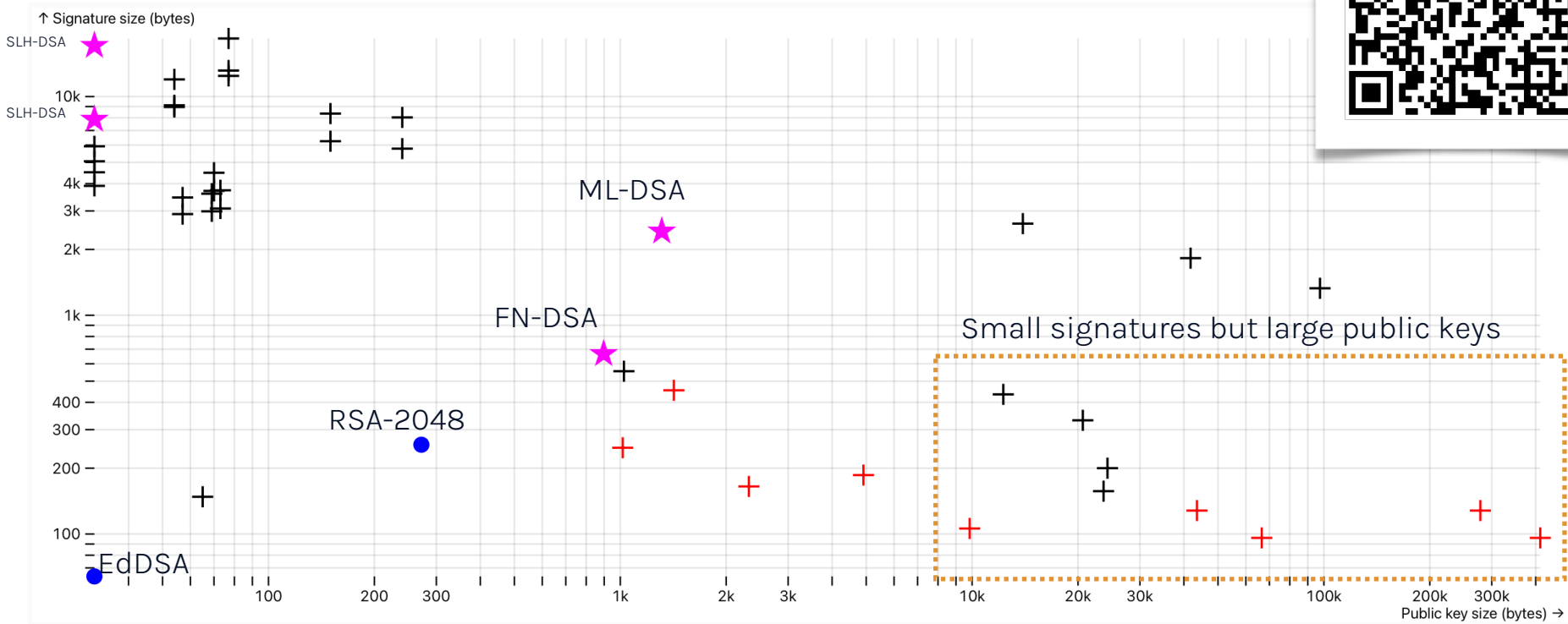
- Code-Based
  - LESS
  - CROSS
- Lattice-based
  - HAWK
- Symmetric-based
  - FAEST
- MPC-in-the-Head
  - Mirath
  - MQOM
  - PERK
  - RYDE
  - SDitH
- Isogeny-based
  - SQIsign
- Multivariate
  - UOV
  - MAYO
  - QR-UOV
  - SNOVA

Signatures  
will not get  
(much)  
better soon



Caveat: I'm ignoring  
different security  
assumptions and just  
focusing on practicalities

# Key and signature sizes



NIST Level-I parametersets

<https://pqshield.github.io/nist-sigs-zoo/>



## Too Big: pk+sig < 4000 bytes

- Code-Based
  - ~~LESS~~
  - ~~GROSS~~
- Lattice-based
  - **HAWK**
- Symmetric-based
  - **FAEST**
- MPC-in-the-Head
  - **Mirath**
  - ~~MQOM~~
  - ~~PERK~~
  - **RYDE**
  - **SDitH**
- Isogeny-based
  - **SQIsign**
- Multivariate
  - ~~UOV~~
  - **MAYO**
  - ~~QR-UOV~~
  - **SNOVA**



## Too Slow: verification < 10ms

- Code-Based
  - ~~LESS~~
  - ~~GROSS~~
- Lattice-based
  - HAWK
- Symmetric-based
  - **FAEST**
- MPC-in-the-Head
  - Mirath
  - ~~MQOM~~
  - ~~PERK~~
  - RYDE
  - SDitH
- Isogeny-based
  - SQIsign
- Multivariate
  - ~~UOV~~
  - MAYO
  - ~~QR-UOV~~
  - SNOVA

Big improvements since Round 1!



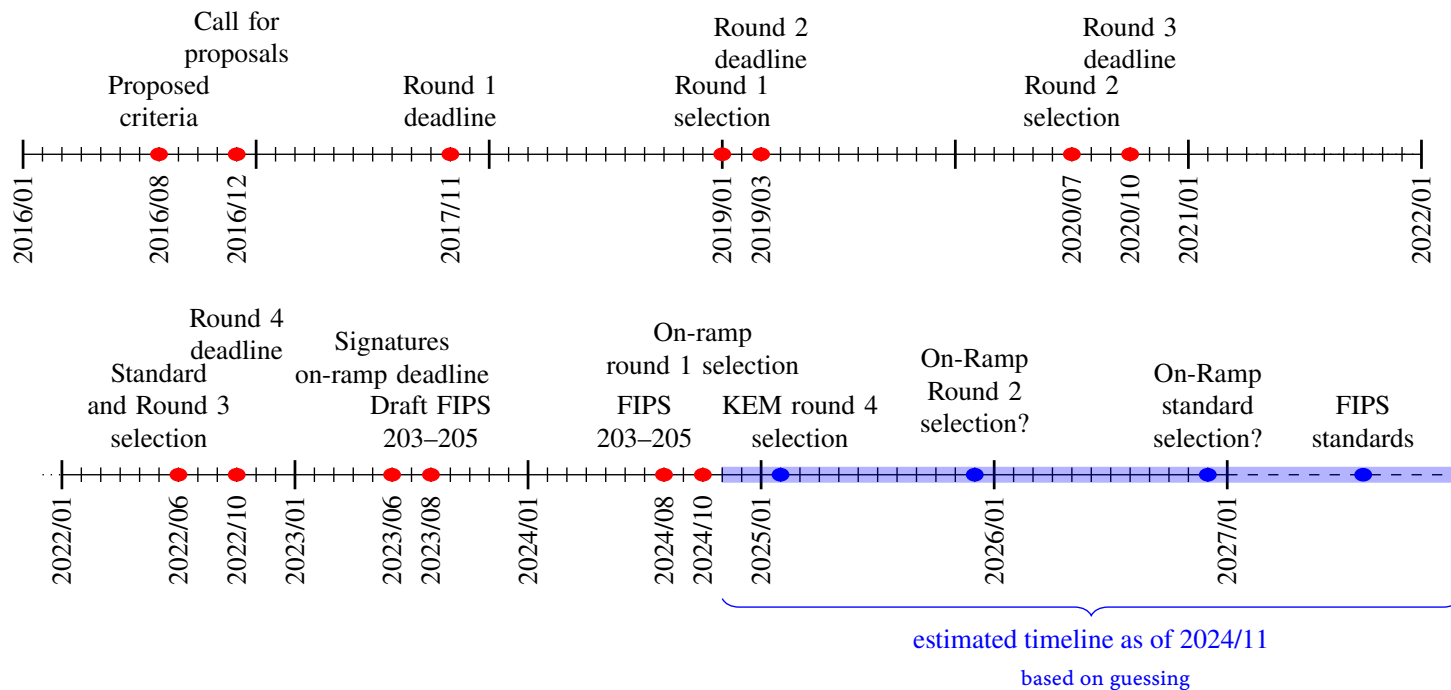
# Promising candidates

- MAYO / SNOVA
  - UOV + additional structure 🌶️
  - MAYO: very new assumptions especially 🌶️🌶️
- MPC-in-the-Head (RYDE, SDitH, Mirath):
  - Perf improvements make it now somewhat attractive
  - Fairly new paradigm, more improvements? Or attacks?
- HAWK:
  - Evolution of Falcon
  - Less-studied security assumptions (🌶️) than Falcon
  - Easier to securely implement than Falcon
  - Will NIST keep an even more spicy modular-lattice based scheme?
- SQL-sign
  - Isogeny-based 🌶️🌶️
  - Huge improvements since round 1





# When will NIST be done?





# The On-Ramp for Signatures

- Only a few “better” algorithms with potential (for general applicability)
- Academics are still working out their security
- Performance/sizes/security can both go up and down
- In any case, NIST won't likely be done soon



# What about other KEMs?

- ML-KEM (Kyber) got picked by **NIST** first, based on lattices
  - ML-KEM-512 pk: 800 bytes, ciphertext: 768 bytes
- HQC: **NIST**'s Round-4 pick
  - Based on error-correcting codes
  - Otherwise fairly similar to ML-KEM
  - HQC-1 pk: 2249 bytes, ct: 4497 bytes
- Classic McEliece: **ISO**
  - Based on McEliece (1978) (error-correcting codes)
  - McEliece-34864: pk 261120 bytes, ct: 96 bytes

## Takeaway:

Useful for diversification,  
Or if you can make use of  
McEliece's trade-off in sizes



# How to afford PQ auth



# Radical proposals: Merkle Tree Certificates

What if we just fully reconsider how authentication works in TLS?

Transport Layer Security  
Internet-Draft  
Intended status: Experimental  
Expires: 9 March 2025

D. Benjamin  
D. O'Brien  
Google LLC  
B. E. Westerbaan  
Cloudflare  
5 September 2024

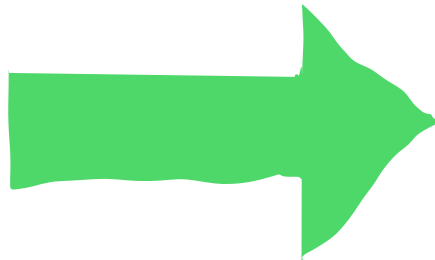
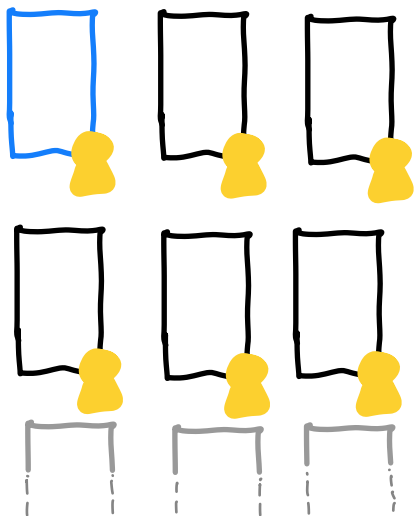
Merkle Tree Certificates for TLS  
draft-davidben-tls-merkle-tree-certs-03

<https://datatracker.ietf.org/doc/draft-davidben-tls-merkle-tree-certs/>

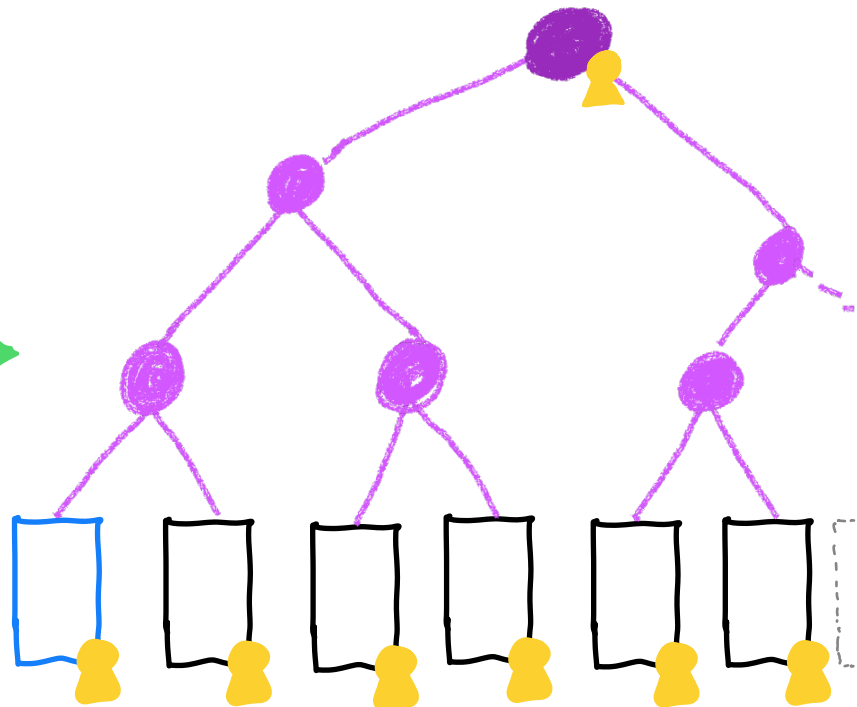


# MTC: Step 1

Thom trust Inc.\*



Merkle tree  
of valid certs





## MTC: Step 2

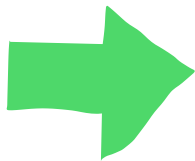
Thom frust



Bas cert



Lets Ekrypt



**moz://a**

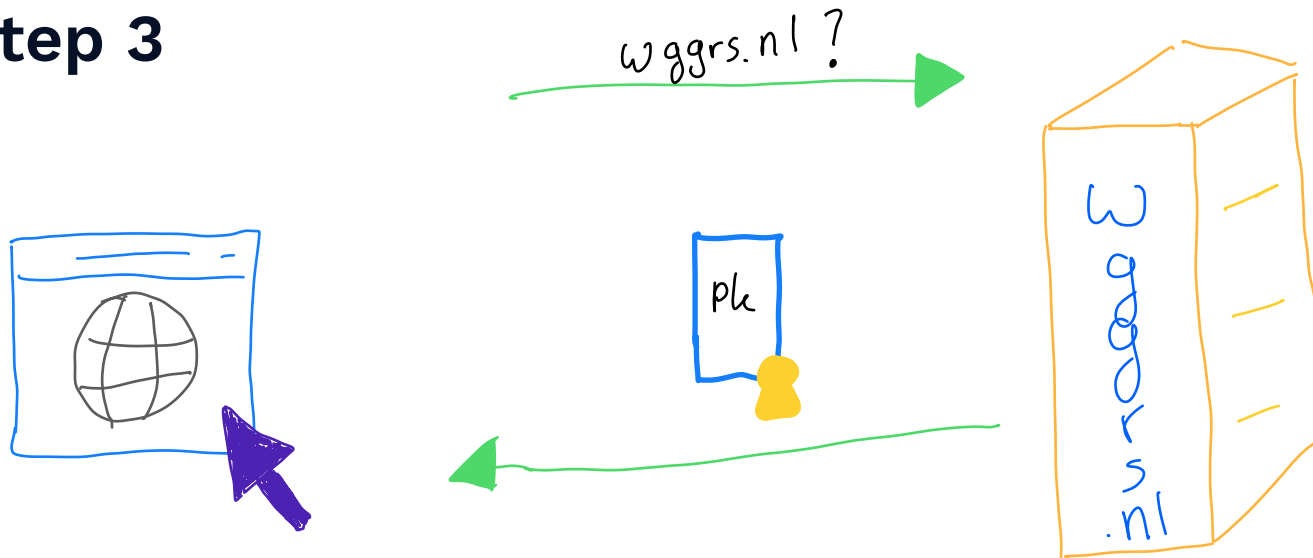


- Audits
- Transparency





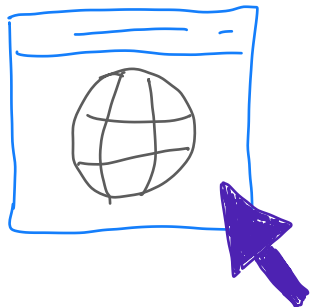
# MTC: Step 3



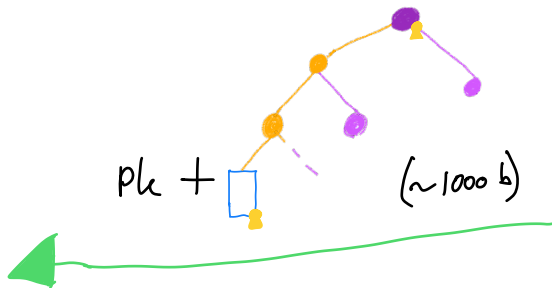
Without MTC



# MTC: Step 3



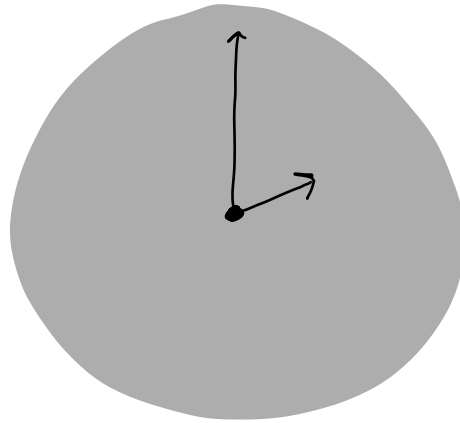
wggrs.nl ?



With MTC



# Merkle Tree Certificates



Repeat every  
hour



# Merkle Tree Certificates

- **Takeaway:** Reconsideration of the status quo can result in significant savings
  - MTC still saves data for classical cryptography!
- **But:** Big changes necessary to every part of the TLS ecosystem
  - Short-lived certificates
  - Webserver must continuously fetch the latest authentication paths
  - Clients must keep downloading currently valid tree heads
  - Automated certificate provisioning such as ACME [RFC8555] should help with this
- New trust model makes security analysis more complicated
  
- MTC is designed for **big deployments and publicly trusted CAs**
  - What about IoT? What about a bank's internal stuff?



## Finally: we're **not done yet**

- **Boring cryptography:** KEM and Signatures (and hashes, symmetric encryption, ...)
  - Solves obvious problems in obvious ways
- **Fancy cryptography** solves subtle privacy and security problems
  - Result of decades of fundamental cryptography research

**Anonymous credentials** and **zero-knowledge proofs** in [Signal's private group system](#). **Oblivious PAKEs** in WhatsApp's [encrypted backups](#), and regular ones in [Magic Wormhole](#). **Unlinkable tokens** in [Apple Private Relay](#) (blind signatures), [Privacy Pass](#) (OPRF), and Dutch [CoronaCheck app](#) (Idemix). **Attribute-Based Encryption** in Cloudflare's [GeoKDL](#). **Private set intersection** with blinding for [password protection in Chrome](#).

We're working on a list: <https://github.com/fancy-cryptography/fancy-cryptography>



# Fancy cryptography requires fancy research

- **More research** is needed to develop (practical, efficient) building blocks to migrate fancy cryptography to a post-quantum world
- Fancy cryptography problems often compete with **not doing anything**
- If we don't solve these problems, we could seriously regress on privacy

Upside: **you like know it** if you are using fancy cryptography.



# We need to think about authentication today

- Where does post-quantum authentication hurt?
- How do we make PQ authentication attractive enough to get people to adopt?
- Can we change protocols to solve our authentication needs with fewer (big, PQ) signatures?
  - Maybe we can **use KEMs to do authentication** [SSW20]
  - Work that you put in today, will still pay off if NIST standardises a smaller scheme
- Ask your protocol designers and developers what happens when you switch to ML-DSA
- Even though things might be fine in theory, practice might require big investments
- While solving Key Exchange first, don't forget to consider authentication

Thanks for your attention