Chrome + PQC Update

CABF F2F Toronto

David Adrian
June 10, 2025

I am not one of the usual Chrome representatives

Product Manager, Chrome Security

David Adrian

Network Security
Memory Safety
Web Platform Security

Previously…

PhD @ University of Michigan

Cofounder, Censys

Principal Engineer, Nametag

https://dadrian.io

Post-quantum cryptography

New cryptographic algorithms and primitives that cannot be broken by a future quantum computer

Quantum Threat

Quantum computers will break classical forms of public/private key (asymmetric) cryptography.

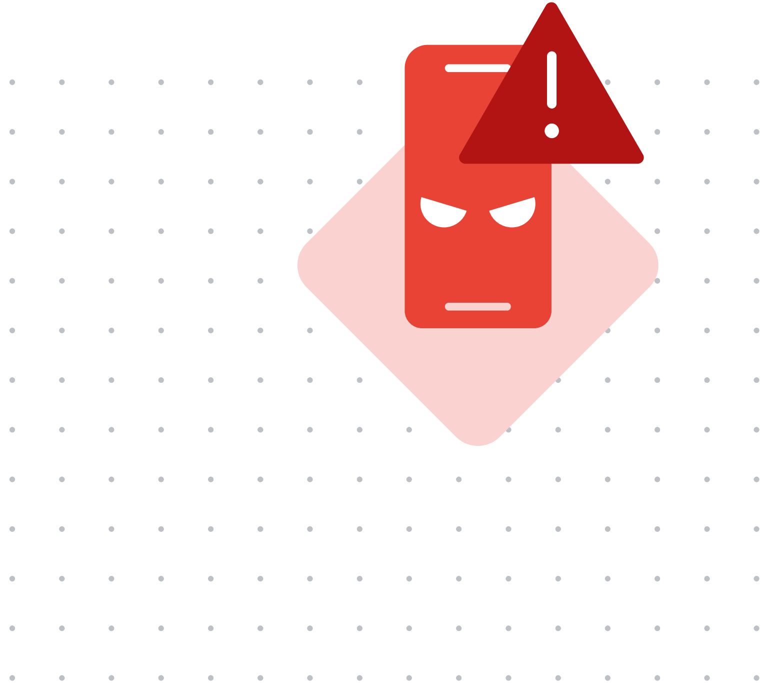
Encryption/Decryption. Encode messages such that a secret key is required to decode the message.
AES, ChaCha-Poly, Simon/Speck ✓

Key Establishment. Securely select a key to use for encryption and decryption Diffie-Hellman, RSA Encrypt, ECDH ✗

Authentication. Ensure the other party is the real thing, not an imposter.
Signatures, RSA Sign, ECDSA ✗

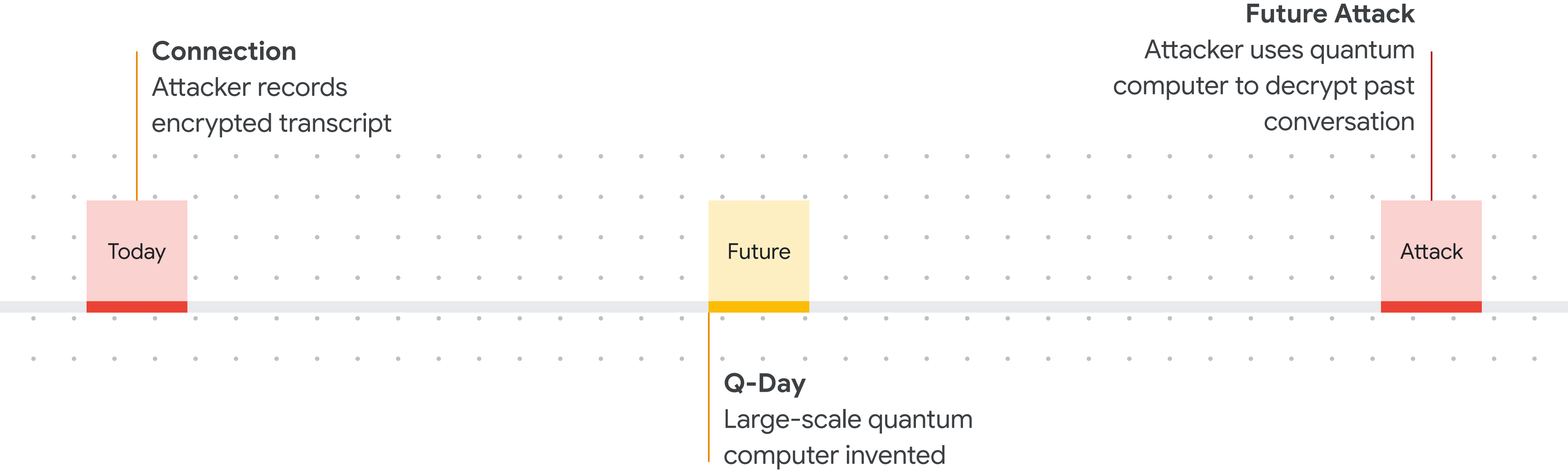
Broken by future quantum computer

Two Threat Models: Key Agreement and Authentication



Until we migrate to post-quantum **key establishment**, current traffic is **vulnerable** to *future* quantum computers

Store Now, Decrypt Later

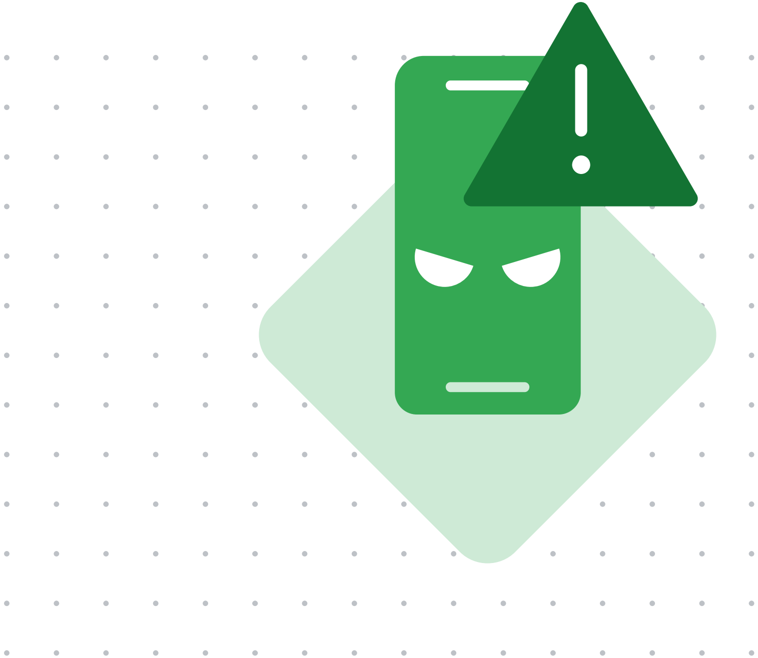

Connection
Attacker records encrypted transcript

Future Attack
Attacker uses quantum computer to decrypt past conversation

Q-Day
Large-scale quantum computer invented

Defense: Use a post-quantum key establishment algorithm *now!*

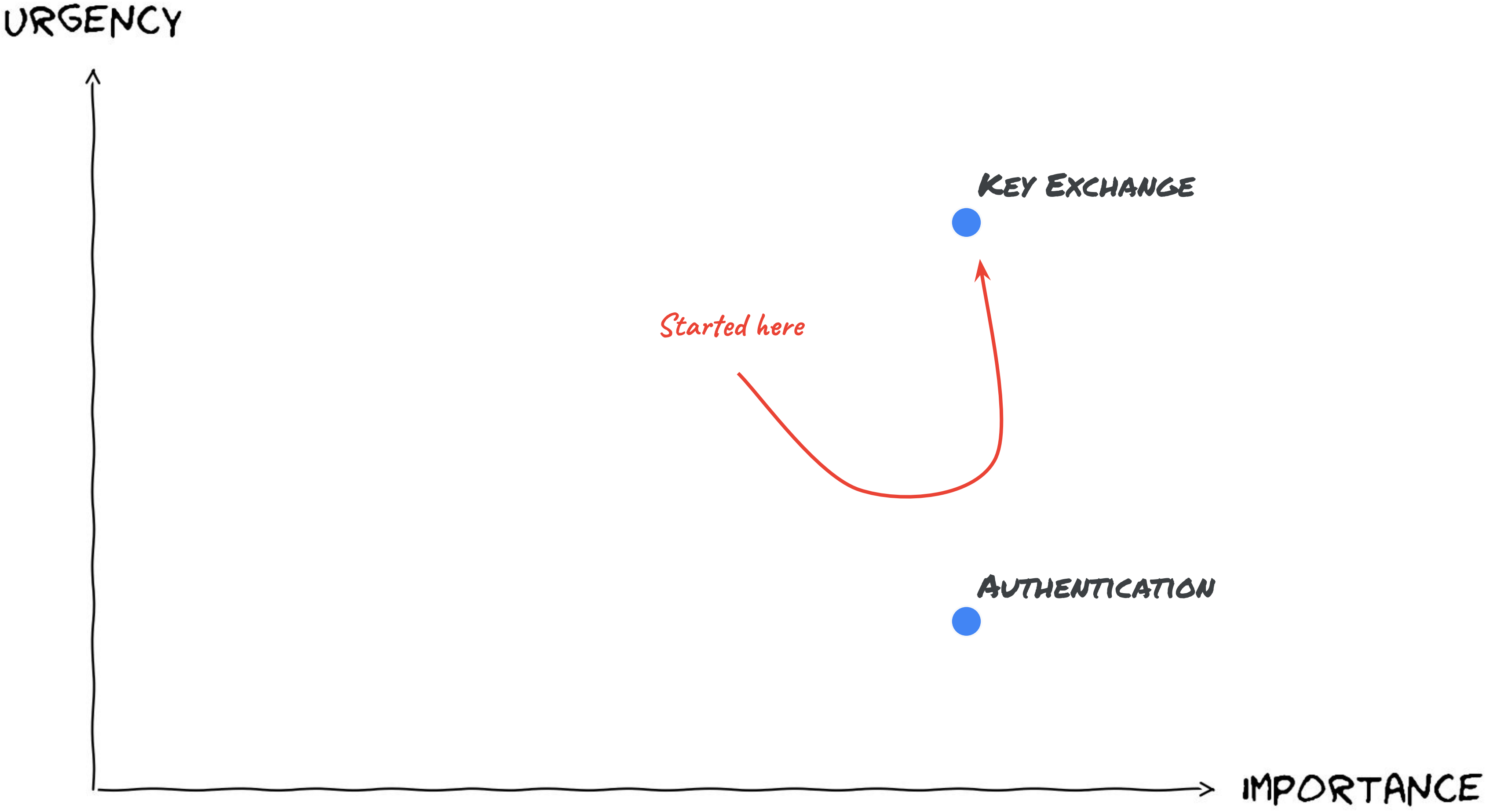
Google



We do not need post-quantum **authentication**, until a quantum computer *actually exists*.

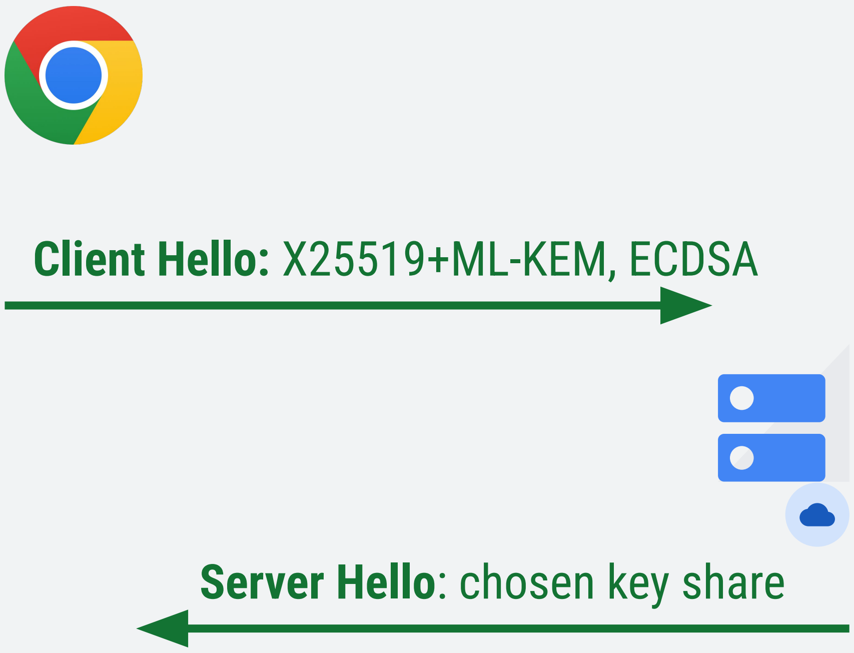
Not Just Tinfoil Hats

- NIST has been running international competitions to select and standardize post-quantum cryptography—Kyber was the winner for **key agreement** [August 2022]
- Chrome 116 deploys experimental support for Kyber in HTTPS [July 2023]
- Signal Messenger deployed post-quantum key agreement in the Signal Protocol [PQXDH][Sep 2023]
- Apple deployed post-quantum key agreement in their latest update to iMessage [PQQ3][Feb 2024]
- Firefox begins experimenting with Kyber on Nightly in Firefox 123 [Feb 2024]
- NIST releases **final Kyber standard, renames to ML-KEM.** Dilithium, the signature algorithm, is renamed to **ML-DSA.** [Aug 2024]
- NSA and GCHQ will require PQC by 2035, EU has a commission [CNSA 2.0][Sep 2022][Dec 2024][GCHQ][March 2025][EU][2024]
- Chrome 131 enables ML-KEM by default [Oct 2024]


URGENCY
KEY EXCHANGE
Started here
AUTHENTICATION
IMPORTANCE

ML-KEM in Chrome

Chrome *offers* hybrid ML-KEM **by default** on desktop platforms since Chrome 131 and Android since Chrome 133

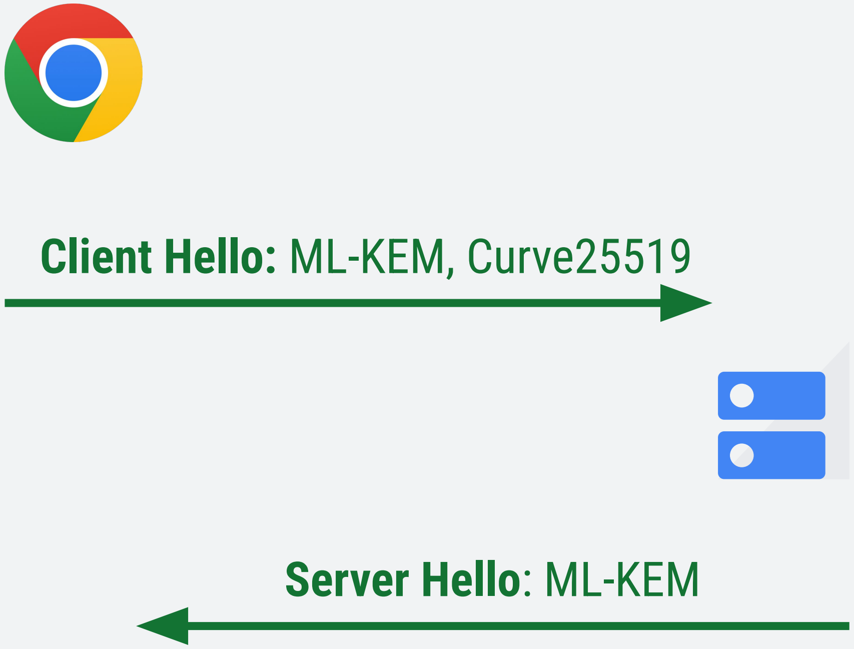


Client Hello: X25519+ML-KEM, ECDSA

Server Hello: chosen key share

ML-KEM at Google

Google Servers *prefer* ML-KEM **by default** for *Google properties* platforms since around the release of Chrome 116.



Client Hello: ML-KEM, Curve25519

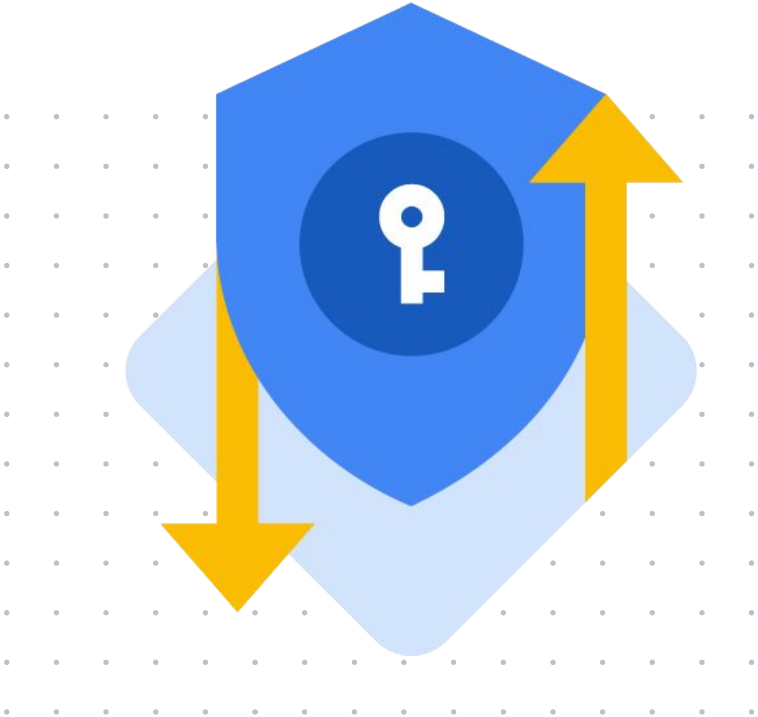
Server Hello: ML-KEM

*Originally, GFEs preferred Kyber, now they prefer ML-KEM. Kyber is no longer supported.

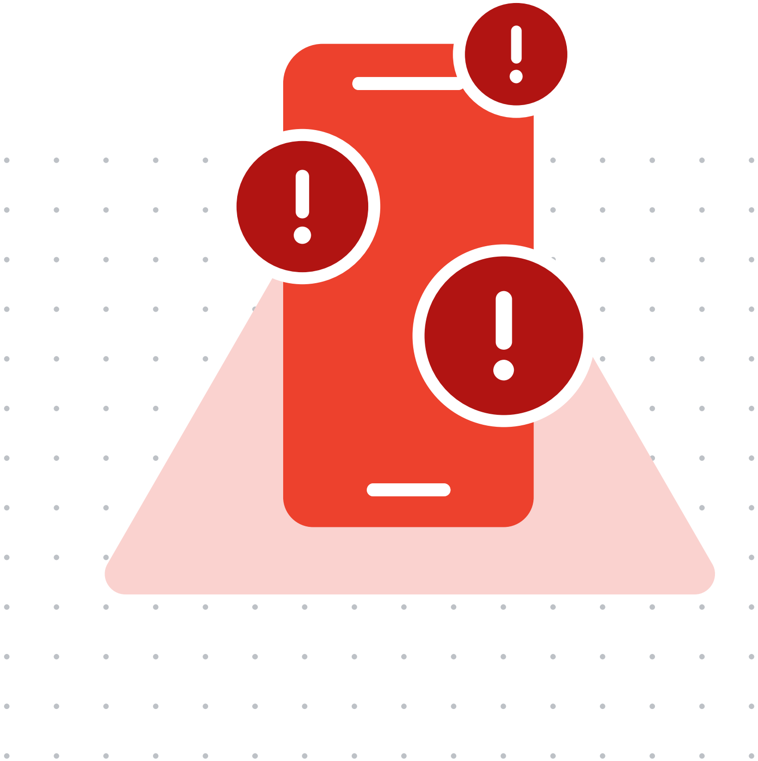
Given all that, let’s talk about the Web PKI.

Post-quantum cryptography…

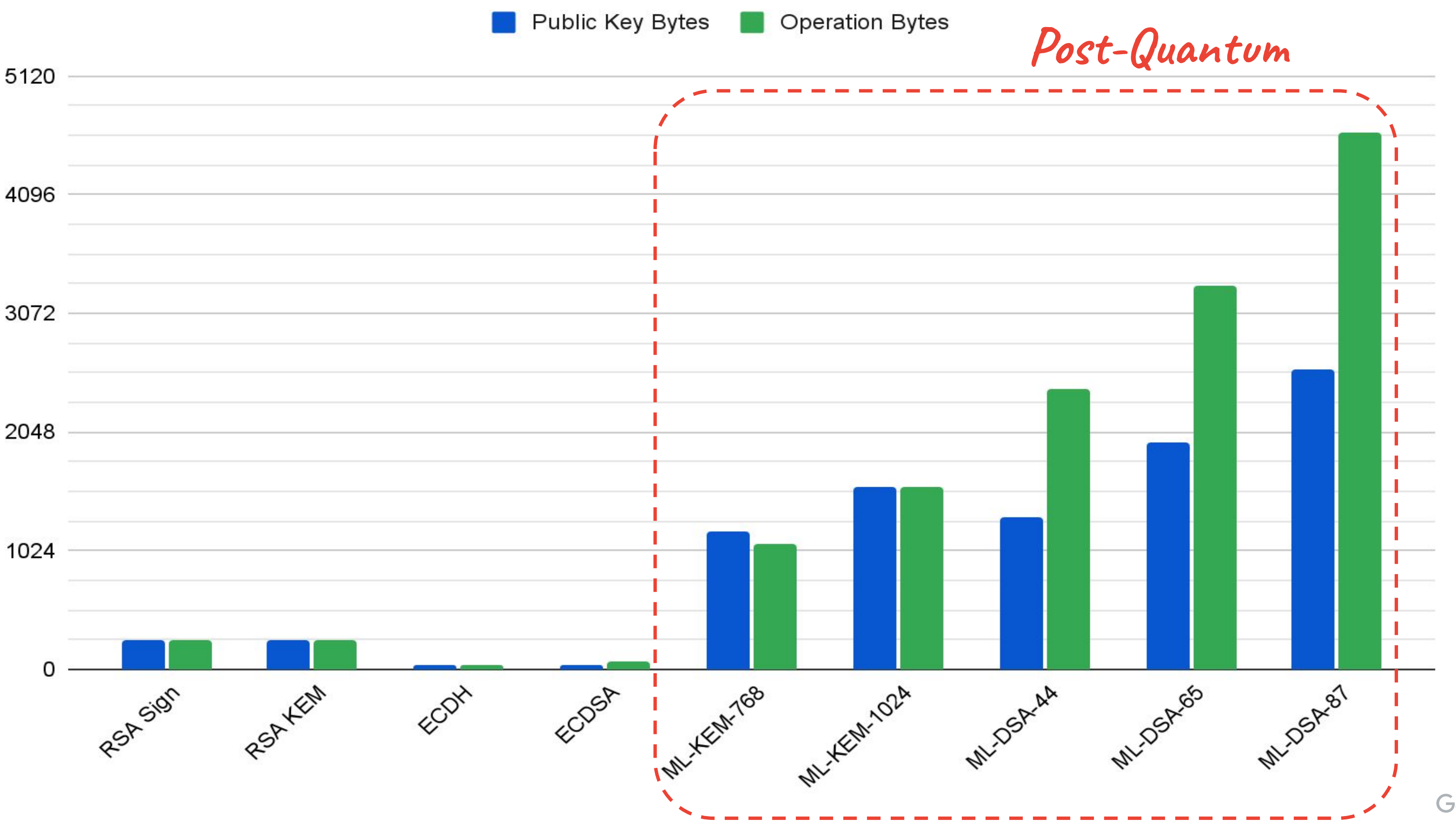
…is really, really big



More bytes =
slow



Greater impact on **mobile** connections, which are a **majority of Chrome users**

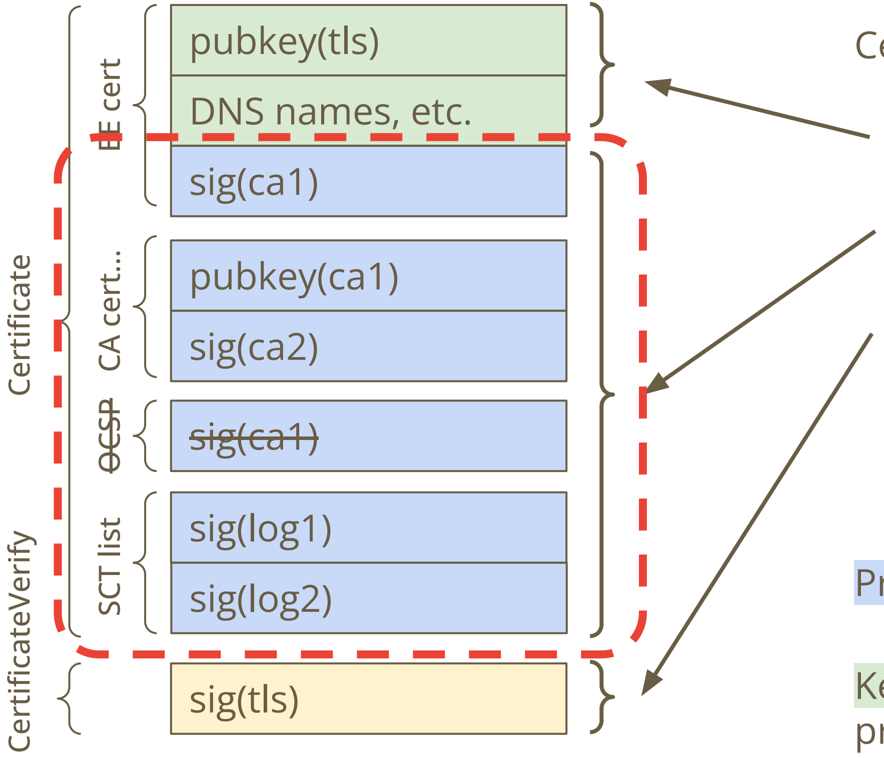

🟦 Public Key Bytes
🟩 Operation Bytes
Post-Quantum
5120
4096
3072
2048
1024
0
RSA Sign
RSA KEM
ECDH
ECDSA
ML-KEM-768
ML-KEM-1024
ML-DSA-44
ML-DSA-65
ML-DSA-87

“Rip and Replace” is too damn big.

- Deploying ML-KEM was **9% latency hit** at 1.1KB in the `ClientHello`.
- Greater impact (50-100%) on very low bandwidth connections (BRICS, sub-saharan Africa)
- Swapping all public keys and signatures to the minimal size ML-DSA-44 with no other changes (intermediates, 2 ML-DSA SCTs) would be an **additional 16KB of data** in the handshake, which would add 40-130% latency.
- Even worse for ML-DSA-87 (CNSA2 required) at 33KB.

See https://dadrian.io/blog/posts/pqc-signatures-2024/
and https://blog.cloudflare.com/pq-2024/

Keys and Signatures in TLS handshake



Certificate and CertificateVerify contain:

A TLS key and application identifier

Proof from the PKI that identifiers match the key

Using the TLS key

Proof is determined by the PKI

Key, identifiers, and usage are determined by the protocol

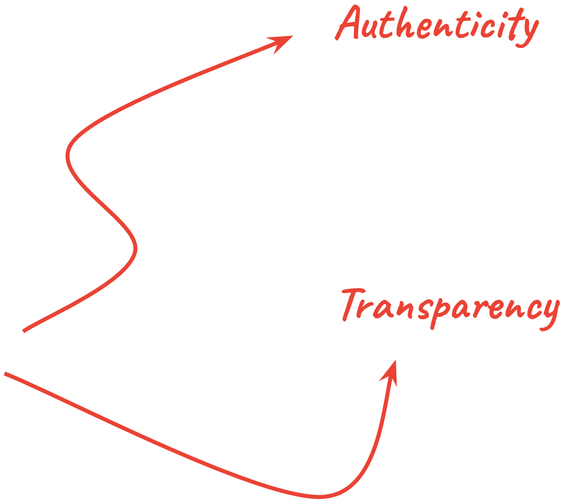
Quantum Threat

Quantum computers will break classical forms of public/private key (asymmetric) cryptography.

Encryption/Decryption. Encode messages such that a secret key is required to decode the message.
AES, ChaCha-Poly, Simon/Speck ✓

Key Establishment. Securely select a key to use for encryption and decryption Diffie-Hellman, RSA Encrypt, ECDH ✗

Authentication. Ensure the other party is the real thing, not an imposter.
Signatures, RSA Sign, ECDSA ✗



Two Threat Models: Key Agreement and Authentication

State of Authenticity

ML-DSA in IETF LAMPS and TLS

Using ML-DSA in X.509 and TLS is still under standardization at the IETF.

Resolve “Hybrid or Not”

There is no consensus on hybrid-or-not. Different compliance regimes have conflicting requirements.

HSM support and FIPS validation

ML-DSA only recently was defined in a FIPS standard, which is a requirement for FIPS validation.

Availability for servers

Without standards, implementations are primarily available in non-standard software packages.

State of Transparency

FIPS-validated algorithms are not required by FIPS / CNSA / etc. Not aware of any compliance obligations for transparency.

We have three options:

1. Keep using *classical* signatures in SCTs even for PQC certs
2. Migrate to *UOV* (66KB keys, 96 byte signatures, non-FIPS)
3. Something completely different

The authentication deadlines are all 2030+...

...which is still far away

Chrome’s Priorities

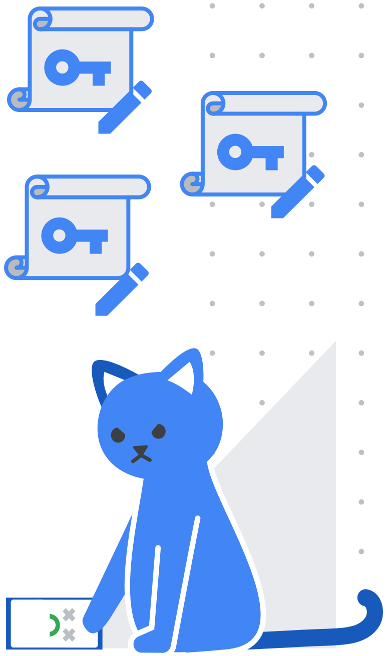
Experimentation.

Enabling Experimentation

The main capability we see as required for *enabling experimentation* is some form of **Trust Anchor Negotiation** (certificate negotiation).

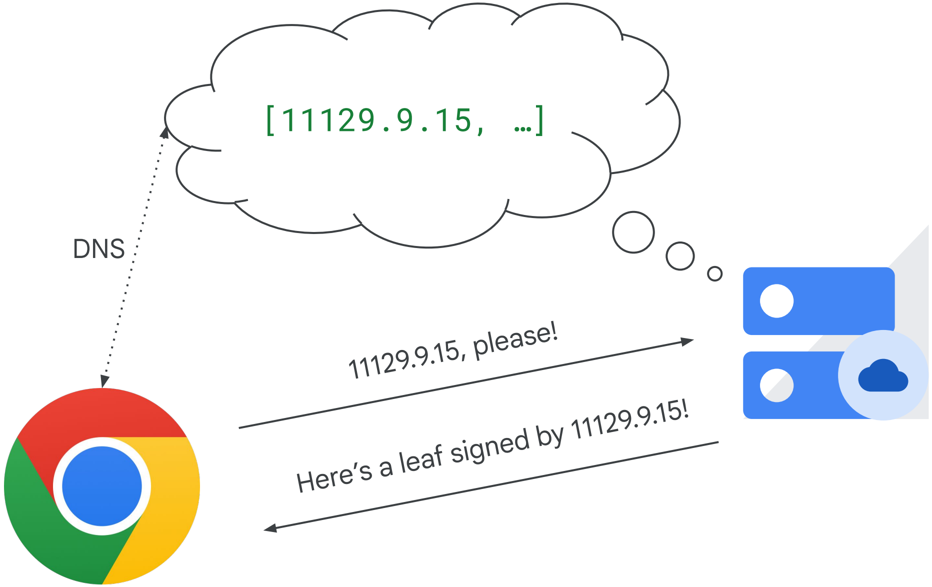
This will enable new clients to experiment with new hierarchies and new authentication schemes without requiring all clients to be updated at the same time.

Continued expansion of **automation** will allow more site operators to participate in experiments.



Trust Anchor Identifiers

- Assign Trust Anchor Identifiers (TAIs) to intermediates and roots
- Advertise in DNS as part of the HTTPS RR
- Clients can optionally pick a TAI in the ClientHello
- Draft RFC adopted by the IETF TLS working group



https://github.com/tlswg/tls-trust-anchor-ids

Trust Anchor Negotiation Benefits

1

Elide intermediates for up-to-date clients

Transmitting intermediate certificates wastes bandwidth, even more so for long chains or post-quantum algorithms. What if we could avoid this?

2

Experiment with post-quantum authentication

Enable support for experimental post-quantum schemes only supported by a subset of new clients, without ossifying on to the first attempt.

3

Solve the problem of root store divergence

Adds a well-lit path for a single hostname to support a set of clients that have no intersection in root store contents and requirements.

How can I participate?

Now

- Chrome: Adding support for TAI, working on experimenting with server partners
- **CAs: Further encourage automation among subscribers**

Eventually, *dependent on experimentation and standardization*

- Will need Private Enterprise Number (PEN) from IANA
- Assign OIDs under the PEN to your hierarchies

😢

Our Expectations for PQC

We anticipate that

- …in the public PKI, there will be demand for a new certificate type that mitigates the performance issues by **unifying authenticity and transparency**
- …in the private PKI, there will be demand for **large** ML-DSA X.509 certificate chains

Reimagining PQC CAs

Previously, had “proposed” Merkle Tree Certificates. We have an updated draft we refer to as Photosynthesis*.

Key insights:

- Each CA runs a tiled log (cheap) of its own issued certificates
- Fast issuance—certificates are signed by the logs and mirrors (3 signatures)
- Slow issuance—certificates are batched into a hash-based inclusion proof (0 signatures)

Photosynthesis Introduction on IETF TLS WG

Photosynthesis

Aiming to prototype an experimental deployment with Cloudflare by Q1 2026.

- Usage is negotiated via Trust Anchor Identifiers

For the experiment, domain validation continues to be provided by existing CAs.

- Must be a 1:1 correspondence between Photosynthesis and existing Web PKI certificate (enforced by Google)

Chrome's Actions

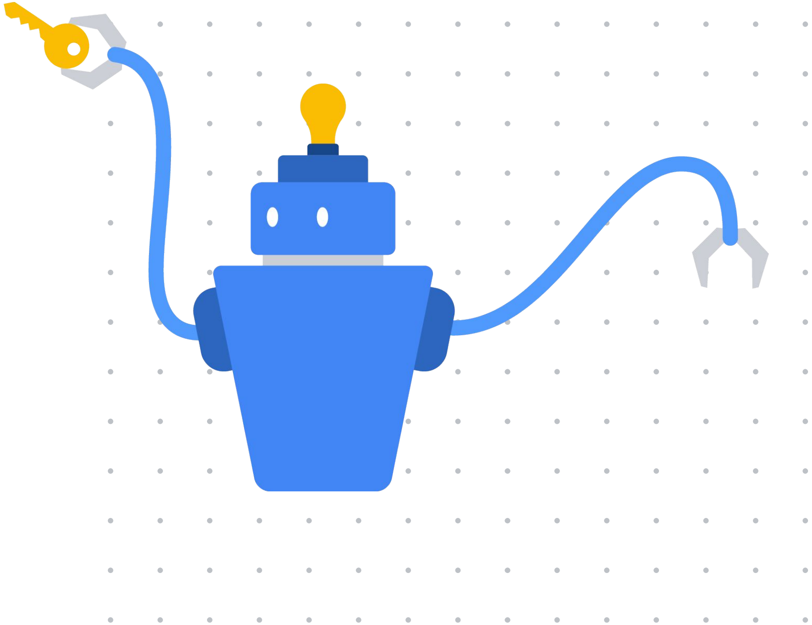
We plan to take Photosynthesis to the IETF.

- We expect there will be opinions
- We plan to focus on real-world experimentation and running code
- We expect any solution will rely on some form of Trust Anchor Negotiation as a building block

But what about a post-quantum Chrome Root Store?

We are confident that we could spin up a policy for post-quantum X.509 roots quickly, should the need arise.

We're equally confident in CAs' ability to spin up a quantum-resistant hierarchy.



A post-quantum root store would skip to the end state of "Moving Forward, Together".

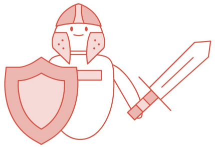
Post-Quantum Root Store Expectations

- New, clean, quantum-resistant, serverAuth only, flat hierarchies.
- Emphasis on automation, short-lived certificates only.
- Chrome Root Program provides a CP/CPS.
- Leverage the CCADB for any additional disclosures and self-attestations.
- Focus heavily on automated, externally-verifiable requirements, e.g. reproducible domain validation, CA key attestation, linting

No ETA, not a current priority, non-normative. Focus is on **experimentation** with *new systems* that reduce the performance impact.

Summary

Our priority is **experimenting** with new structures for unified issuance transparency and authenticity.



We are optimistic that we can add flag-gated ML-DSA support for **private, non-publicly trusted** PKIs in late 2026, depending on IETF progress.

We ask CAs continue to **encourage automation among their subscribers** to better prepare for lifetime reduction *and* post-quantum.

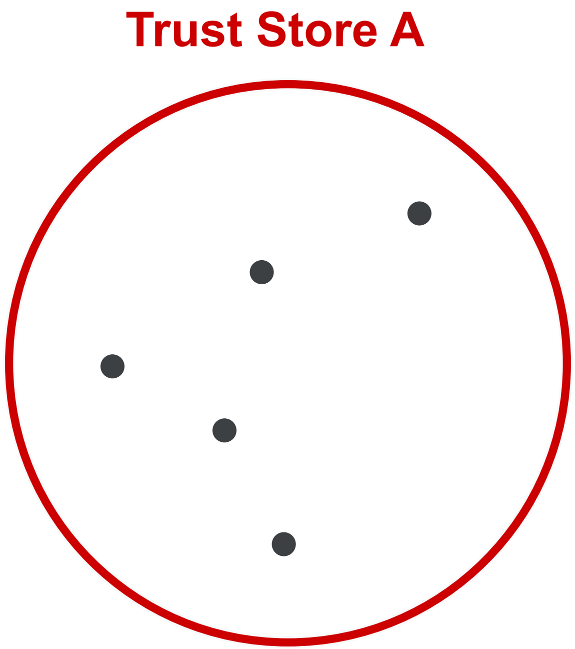
Chrome PQC Update

June 10, 2025

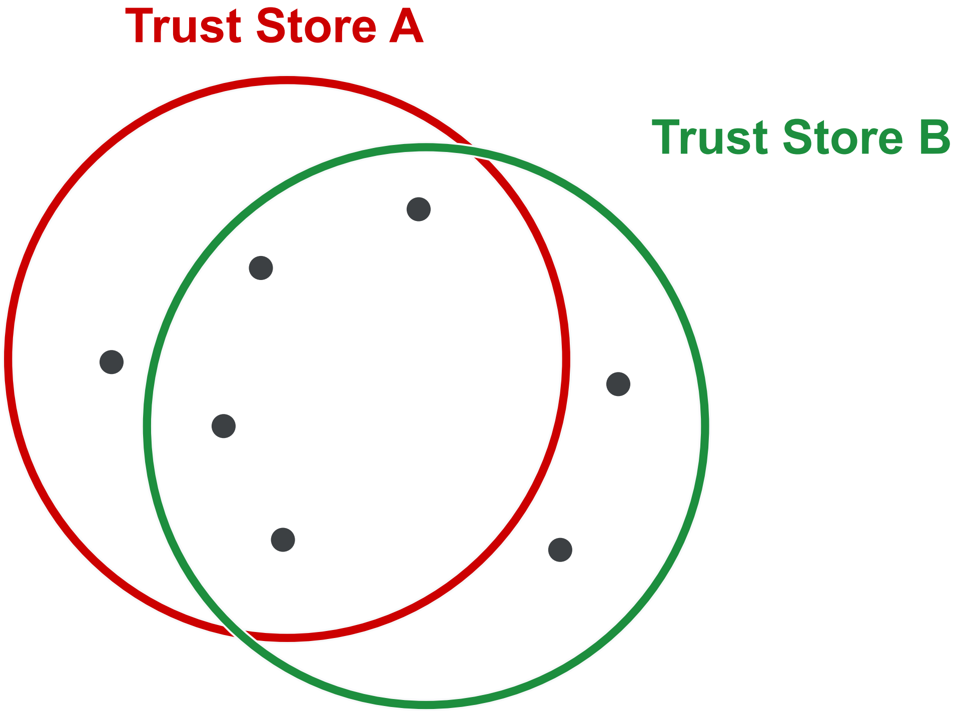
chrome

Appendix

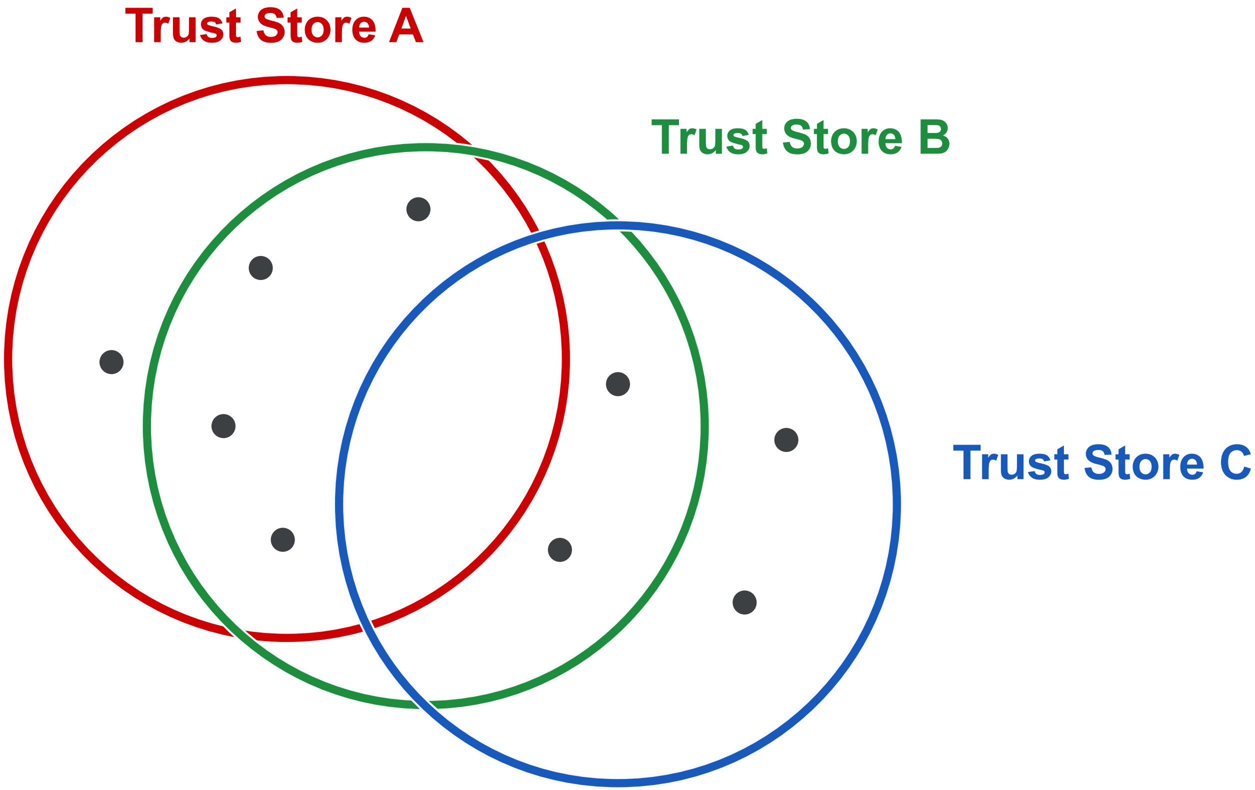
Challenges with Trust Stores: Client Divergence



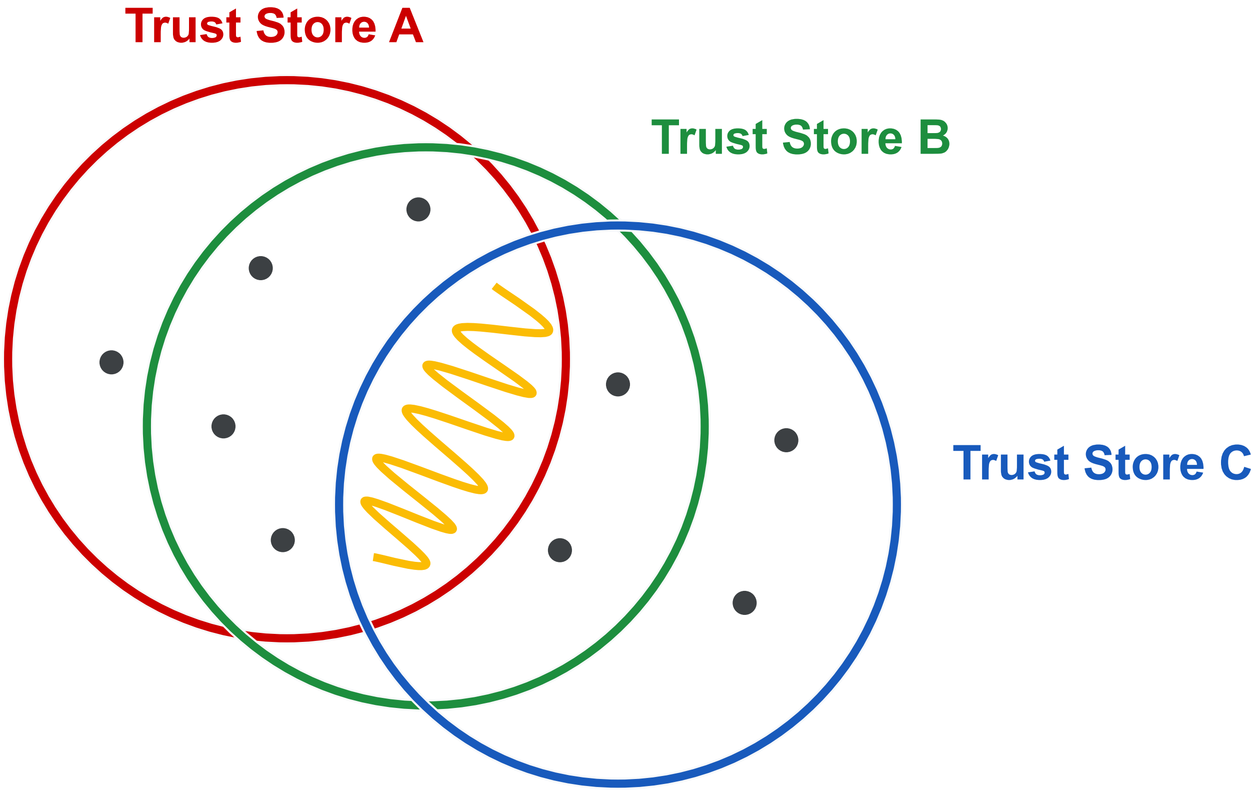
Challenges with Trust Stores: Client Divergence



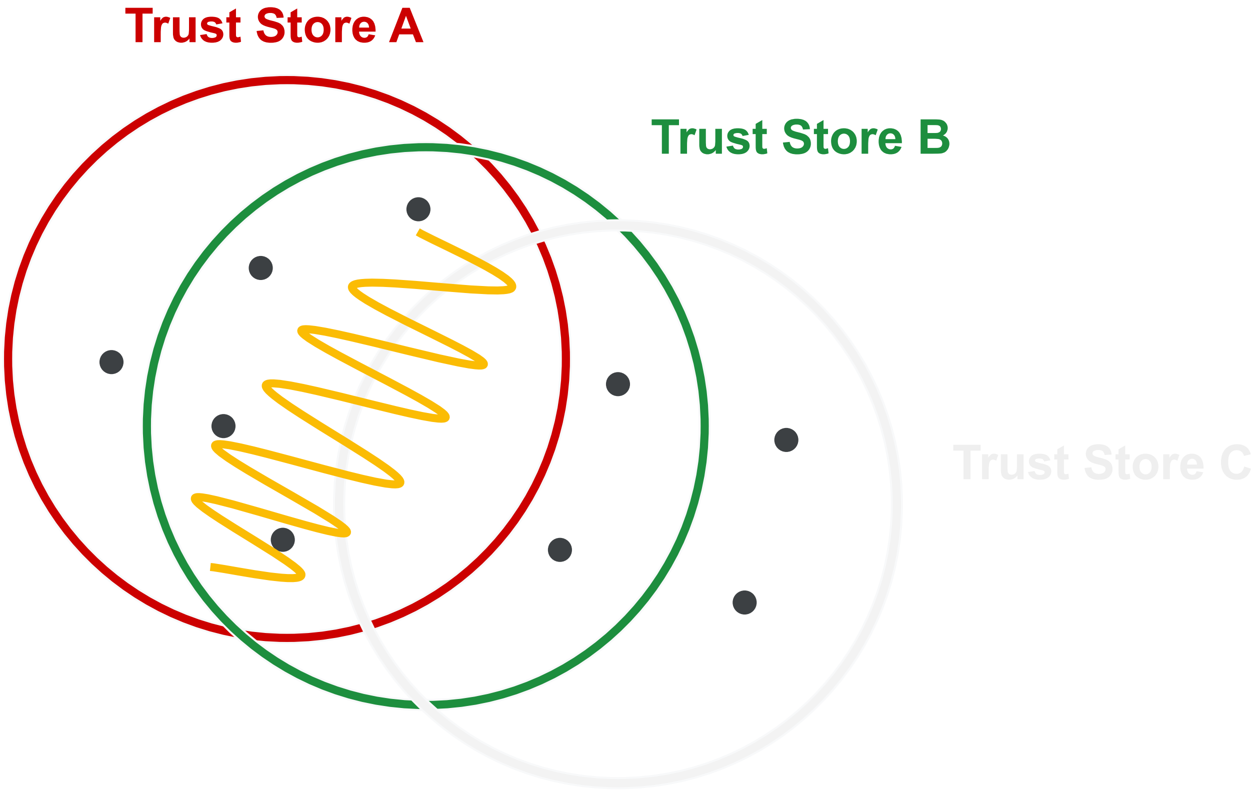
Challenges with Trust Stores: Client Divergence



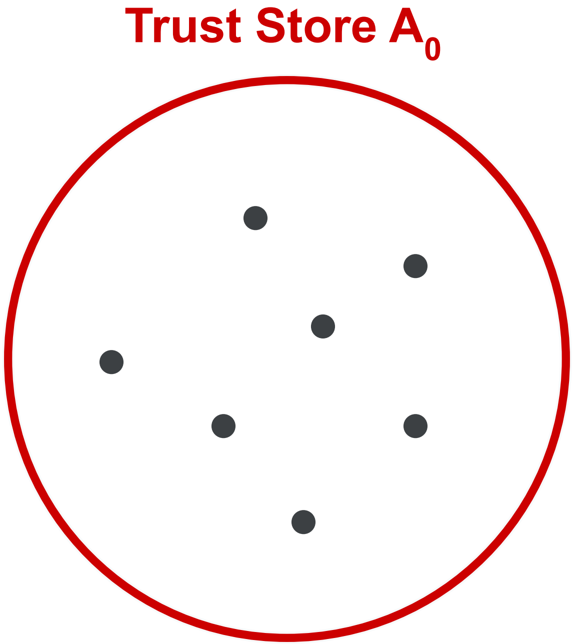
Challenges with Trust Stores: Client Divergence



Challenges with Trust Stores: Client Divergence

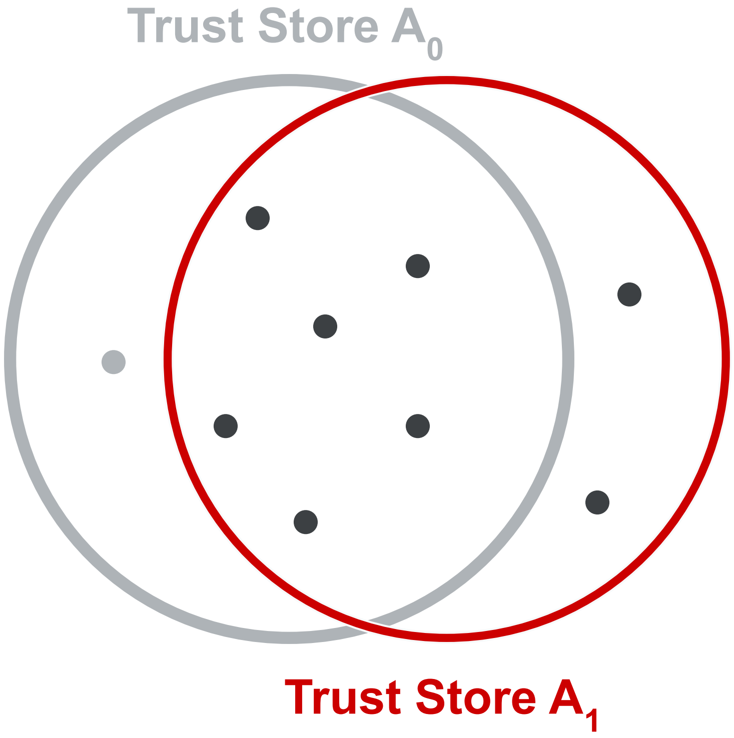


Challenges with Trust Stores: Temporal Divergence

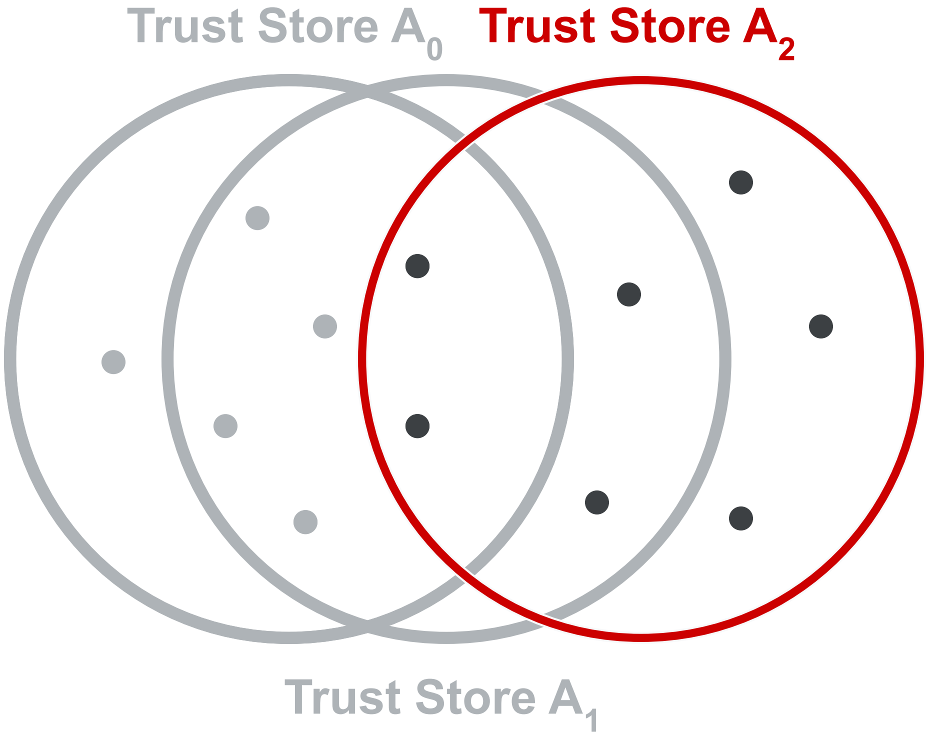


Trust Store A_0

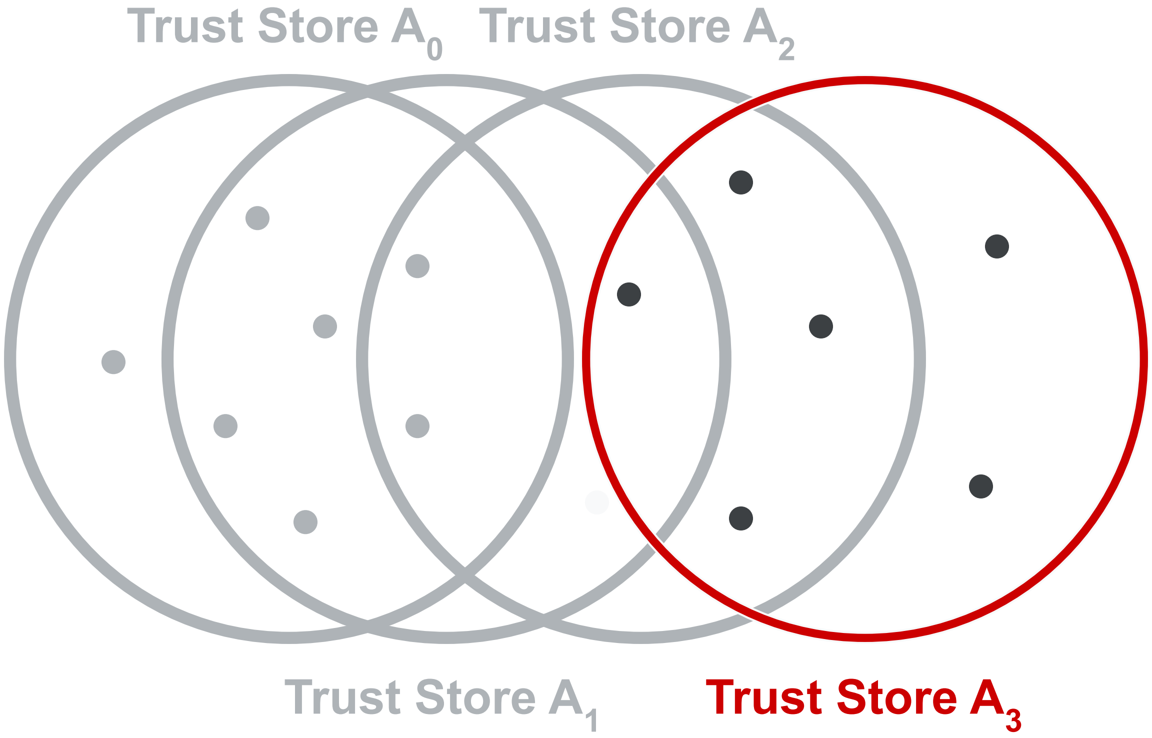
Challenges with Trust Stores: Temporal Divergence



Challenges with Trust Stores: Temporal Divergence



Challenges with Trust Stores: Temporal Divergence


Trust Store A_0
Trust Store A_2
Trust Store A_1
Trust Store A_3