

OPEN SOURCE SecurityCon NORTH AMERICA

Why Post-Quantum Authentication Will Break K8s (and How Merkle Trees Fix It)

Fabian Kammel, ControlPlane

whoami

Fabian Kammel



Worked as

Principal Security Consultant @ ControlPlane

Trainer for K8s & Container Security

Security Engineer on **Confidential K8s**

Systems Architect for **Cloud-Native PKI**

Presented at

+ KubeCon
+ KCD
+ BSides

Guest on

+ Kubernetes Podcast
+ Kubernetes Blog

Contributor to

+ OpenBao (PQC)

@controlplaneio

ControlPlane - We leave it better than we found it

controlplane

Who we are?

Cloud Native and Open Source security consultancy established in 2017. Our diverse culture empowers and develops individuals with talent and integrity.

Accustomed to work in highly-regulated environments and strategic programmes

Industry Leading

Kubernetes (first threat model for Financial Services User Group (FSUG)

Hacking Kubernetes **O'Reilly**, authors and Trainer

Defending Cloud Native - **SANS**, SEC 584 authors and trainers

Future Focussed

Linux Foundation's Security Technical Advisory Group (co-chair, CNCF)

OpenUK (CISO, pro bono)

Community organisers and collaborators (i.e. KubeCon, CloudNativeCon)

Quantum Computing

What is it?

- Classical computers work on discrete binary data
- **Quantum computers** use **quantum superposition**, interference & entanglement for computations

Quantum Computing

What is it?

- Classical computers work on discrete binary data
- **Quantum computers** use **quantum superposition**, interference & entanglement for computations

What is it good for?

- They excel at simulating nature! Run experiments for:
  - Biology, Physics, Chemistry, Material Sciences, ...

Quantum Computing

What is it?

- Classical computers work on discrete binary data
- **Quantum computers** use **quantum superposition**, interference & entanglement for computations

What is it good for?

- They excel at simulating nature! Run experiments for:
  - Biology, Physics, Chemistry, Material Sciences, ...

Why do we care?

- They have one superstar: **Shor's Algorithm**

Quantum Computing

What is it?

- Classical computers work on discrete binary data
- **Quantum computers** use **quantum superposition**, interference & entanglement for computations

What is it good for?

- They excel at simulating nature! Run experiments for:
  - Biology, Physics, Chemistry, Material Sciences, ...

Why do we care?

- They have one superstar: **Shor's Algorithm**. Breaks:
  - RSA & ECC cryptography
  - The Internet
  - The world as we know it!

Quantum Computing

What is it?

- Classical computers work on discrete binary data



  - Cryptography
  - The Internet
  - The world as we know it!

Quantum Computing

What is it?

- Classical computers work on discrete binary data

China breaks RSA encryption with a quantum computer, threatening global data security

Using a quantum annealing processor built by D-Wave Systems, the researchers say they factored a 22-bit RSA integer that had resisted earlier attempts on the same class of hardware. Wang Chao and colleagues at Shanghai University carried out the experiment.

Who was *here* last year?



Key Exchange vs. Signatures

Two basic classes of algorithms

- TLS solves two different challenges

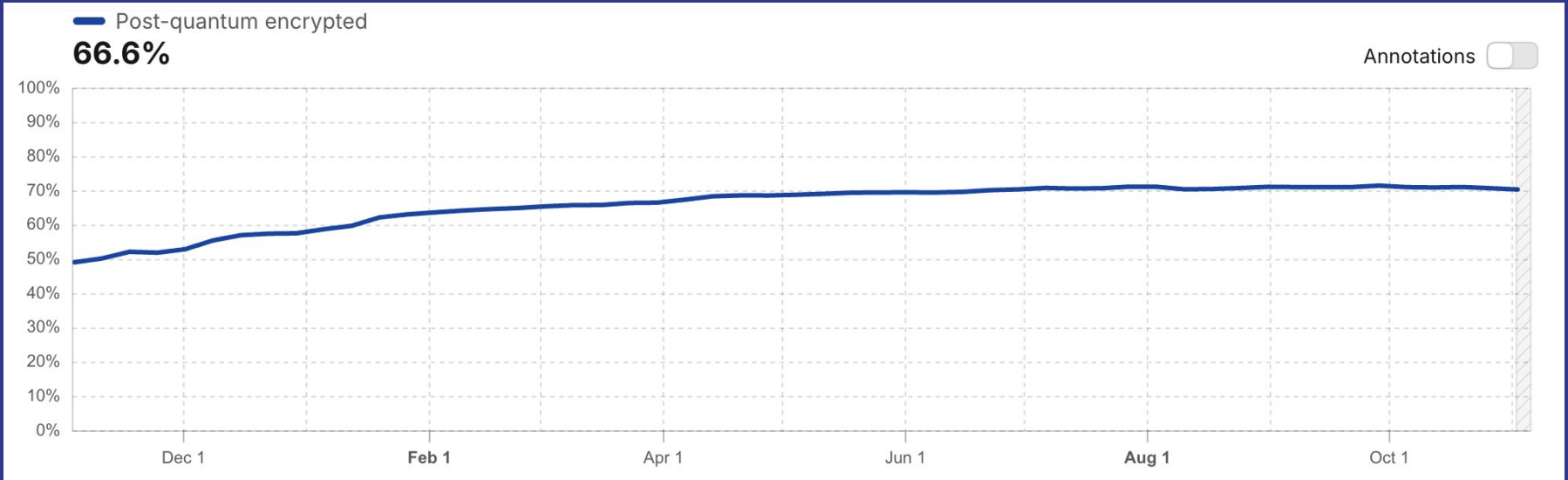
1. Exchange shared secret over **untrusted channel**
2. Establish trust between two **unknown parties**

Key Exchange vs. Signatures

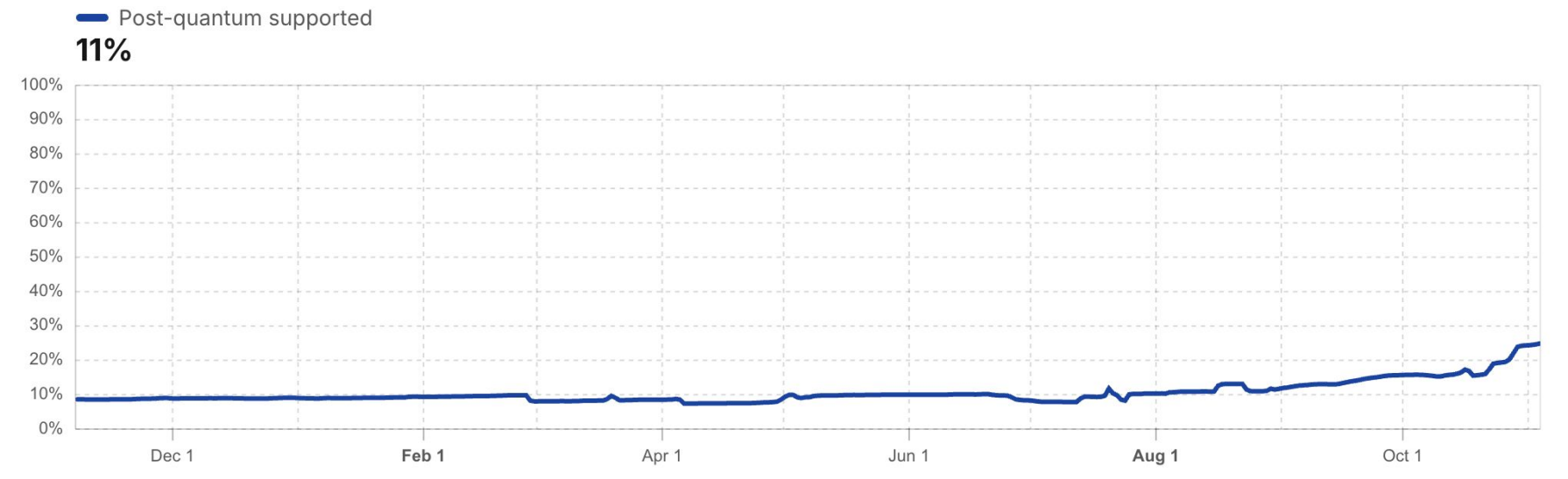
Two basic classes of algorithms

- TLS solves two different challenges
  1. Exchange shared secret over **untrusted channel**
  2. Establish trust between two **unknown parties**
- First is ‘solved’ via Key Exchange Mechanisms (KEM)
  - Important to solve: store-now-decrypt-later

Cloudflare PQC Encryption (%)



Origin Servers PQC (%)



Key Exchange vs. Signatures

Two basic classes of algorithms

- TLS solves two different challenges
  1. Exchange shared secret over **untrusted channel**
  2. Establish trust between two **unknown parties**
- First is ‘solved’ via Key Exchange Mechanisms (KEM)
  - Important to solve: store-now-decrypt-later
- Second is solved via Web PKI and signatures
  - Cannot be manipulated in future
  - More time to build this

What is available today?

What are my options?

- As with AES, NIST kicked-off a standardization process for post quantum algorithms in 2016
- 4 winners were selected in 2024

Algorithm	Type	Underlying Math
ML-KEM (FIPS-203)	Key Exchange Mechanism	Lattice
ML-DSA (FIPS-204)	Signature	Lattice
SLH-DSA (FIPS-205)	Signature	Hash
FN-DSA (pending..)	Signature	Lattice

Signature Zoo (NIST)

Algorithm	Public Key (b)	Signature (b)	Signing (t)	Verification (t)
Ed25519	32	64	0.15	1.3
ML-DSA	1,312	2,420	1	1
SLH-DSA	32	7,856	14,000	40
FN-DSA	897	666	3 (!)	0.7

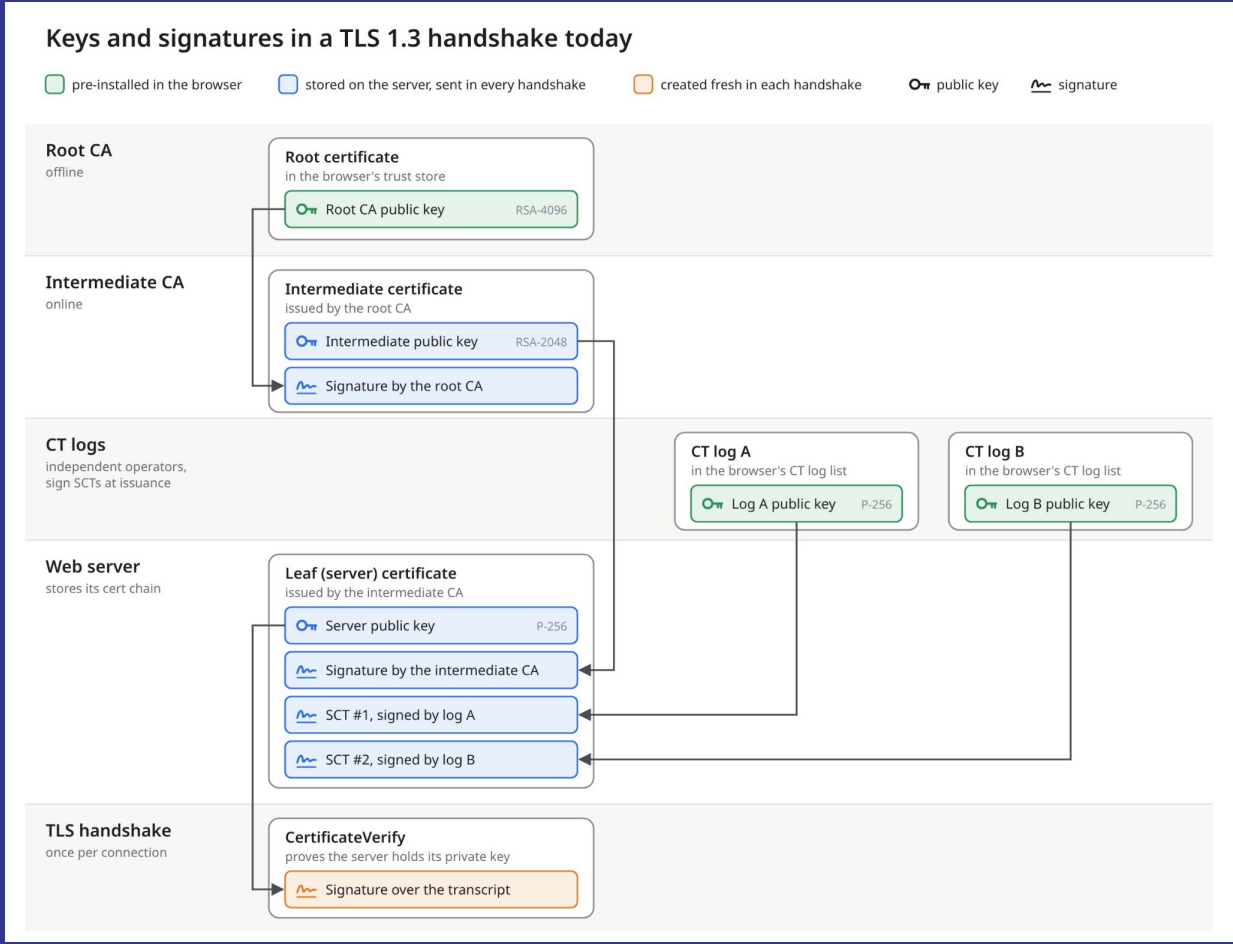


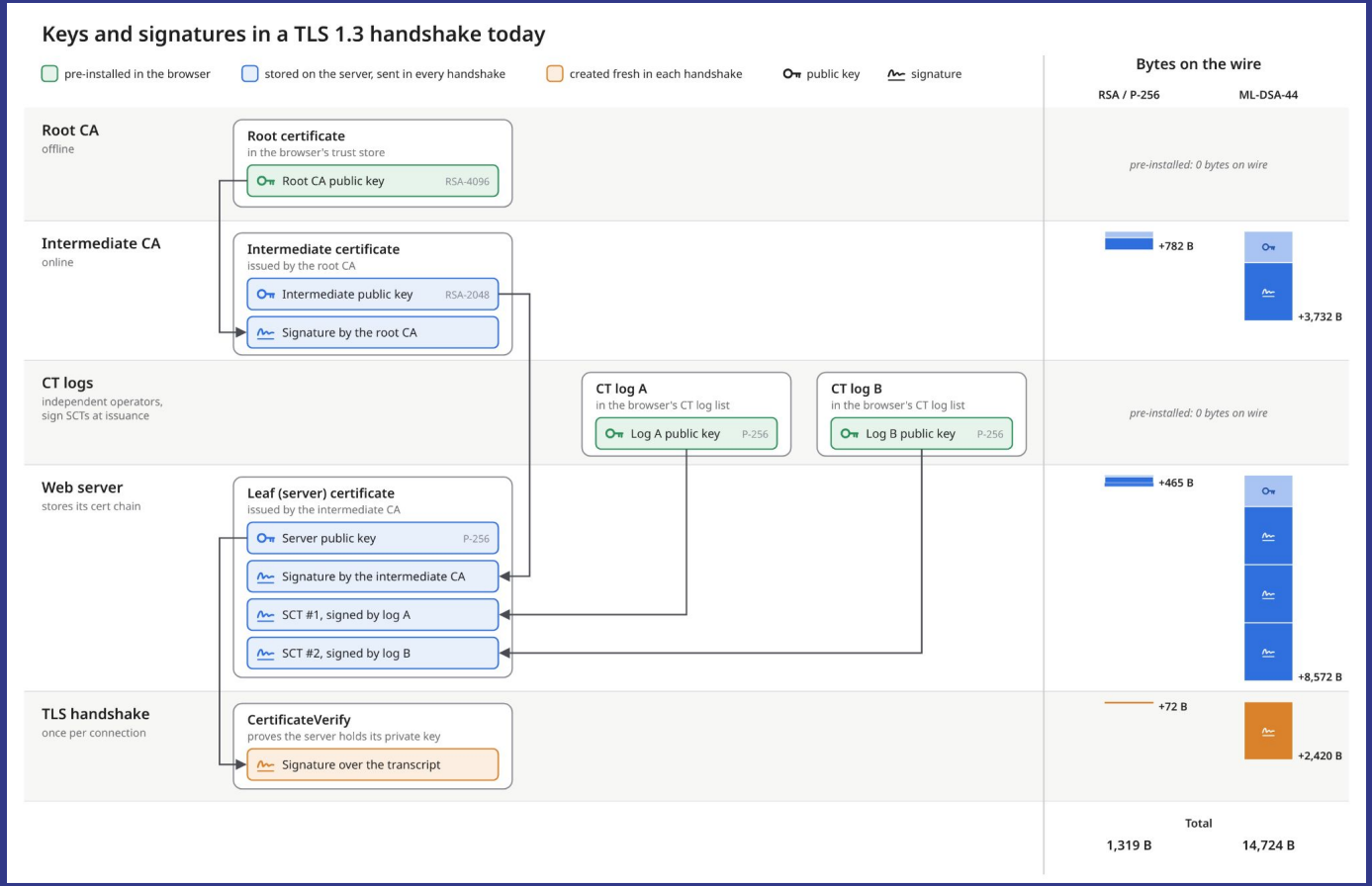
Signature Zoo (NIST)

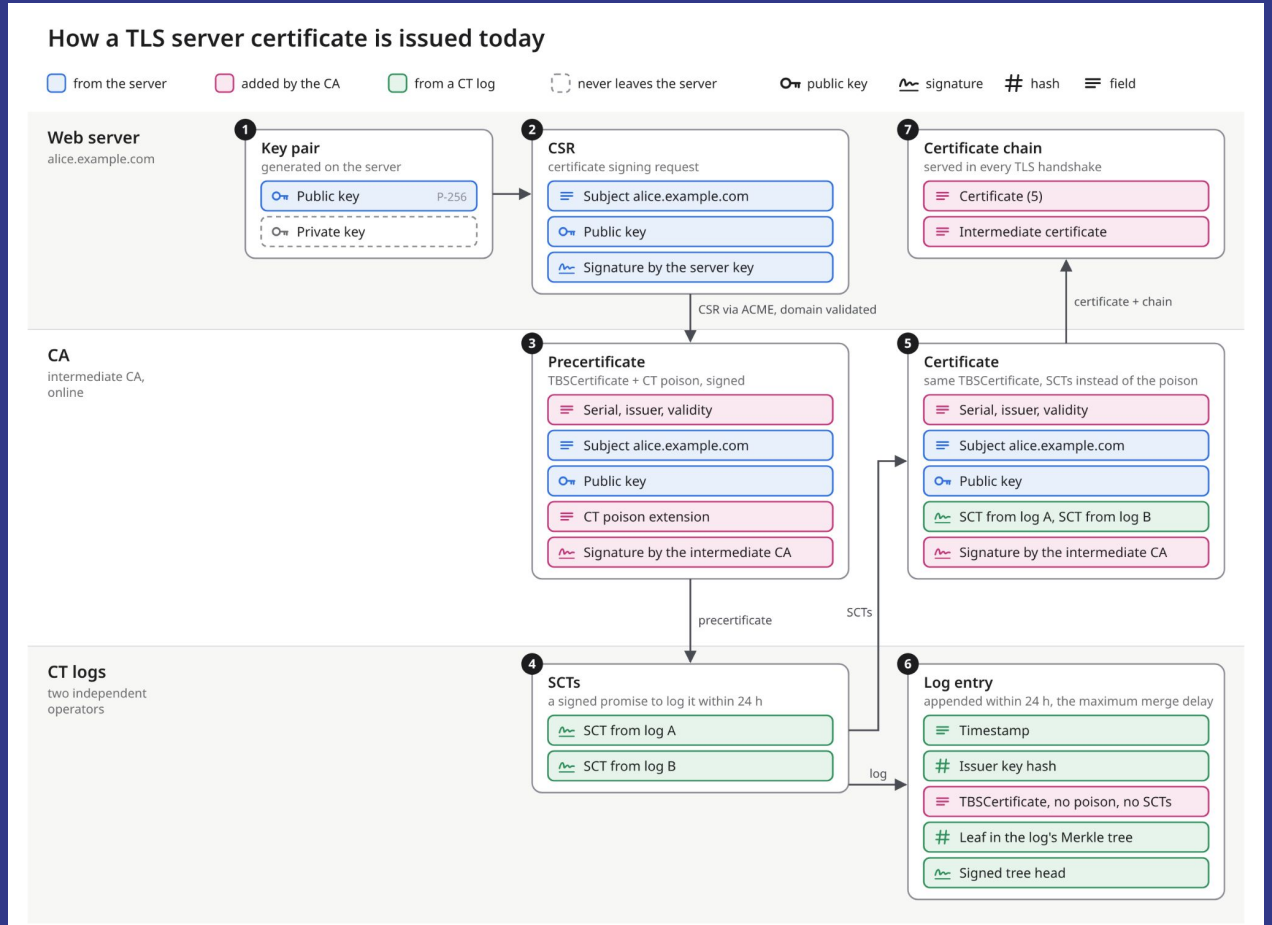
Algorithm	Public Key (b)	Signature (b)	Signing (t)	Verification (t)
Ed25519	32	64	0.15	1.3
ML-DSA	1,312	2,420	1	1
SLH-DSA	32	7,856 ~120x	14,000 ~100000x	40
FN-DSA	897	666	3 (!)	0.7

Signature Zoo (NIST)

Algorithm	Public Key (b)	Signature (b)	Signing (t)	Verification (t)
Ed25519	32	64	0.15	1.3
ML-DSA	1,312 ~40x	2,420 ~40x	1	1
SLH-DSA	32	7,856	14,000	40
FN-DSA	897	666	3 (!)	0.7


Keys and signatures in a TLS 1.3 handshake today
pre-installed in the browser
stored on the server, sent in every handshake
created fresh in each handshake
public key
signature
Root CA
offline
Root certificate
in the browser's trust store
Root CA public key
RSA-4096
Intermediate CA
online
Intermediate certificate
issued by the root CA
Intermediate public key
RSA-2048
Signature by the root CA
CT logs
independent operators,
sign SCTs at issuance
CT log A
in the browser's CT log list
Log A public key
P-256
CT log B
in the browser's CT log list
Log B public key
P-256
Web server
stores its cert chain
Leaf (server) certificate
issued by the intermediate CA
Server public key
P-256
Signature by the intermediate CA
SCT #1, signed by log A
SCT #2, signed by log B
TLS handshake
once per connection
CertificateVerify
proves the server holds its private key
Signature over the transcript


Keys and signatures in a TLS 1.3 handshake today
pre-installed in the browser
stored on the server, sent in every handshake
created fresh in each handshake
public key
signature
Bytes on the wire
RSA / P-256
ML-DSA-44
Root CA
offline
Root certificate
in the browser's trust store
Root CA public key
RSA-4096
pre-installed: 0 bytes on wire
Intermediate CA
online
Intermediate certificate
issued by the root CA
Intermediate public key
RSA-2048
Signature by the root CA
+782 B
+3,732 B
CT logs
independent operators,
sign SCTs at issuance
CT log A
in the browser's CT log list
Log A public key
P-256
CT log B
in the browser's CT log list
Log B public key
P-256
pre-installed: 0 bytes on wire
Web server
stores its cert chain
Leaf (server) certificate
issued by the intermediate CA
Server public key
P-256
Signature by the intermediate CA
SCT #1, signed by log A
SCT #2, signed by log B
+465 B
+8,572 B
TLS handshake
once per connection
CertificateVerify
proves the server holds its private key
Signature over the transcript
+72 B
+2,420 B
Total
1,319 B
14,724 B


How a TLS server certificate is issued today
from the server
added by the CA
from a CT log
never leaves the server
public key
signature
hash
field
Web server
alice.example.com
1
Key pair
generated on the server
Public key
P-256
Private key
2
CSR
certificate signing request
Subject alice.example.com
Public key
Signature by the server key
CSR via ACME, domain validated
7
Certificate chain
served in every TLS handshake
Certificate (5)
Intermediate certificate
certificate + chain
CA
intermediate CA, online
3
Precertificate
TBSCertificate + CT poison, signed
Serial, issuer, validity
Subject alice.example.com
Public key
CT poison extension
Signature by the intermediate CA
precertificate
5
Certificate
same TBSCertificate, SCTs instead of the poison
Serial, issuer, validity
Subject alice.example.com
Public key
SCT from log A, SCT from log B
Signature by the intermediate CA
SCTs
CT logs
two independent operators
4
SCTs
a signed promise to log it within 24 h
SCT from log A
SCT from log B
log
6
Log entry
appended within 24 h, the maximum merge delay
Timestamp
Issuer key hash
TBSCertificate, no poison, no SCTs
Leaf in the log's Merkle tree
Signed tree head

Merkle Tree

Demo Time!

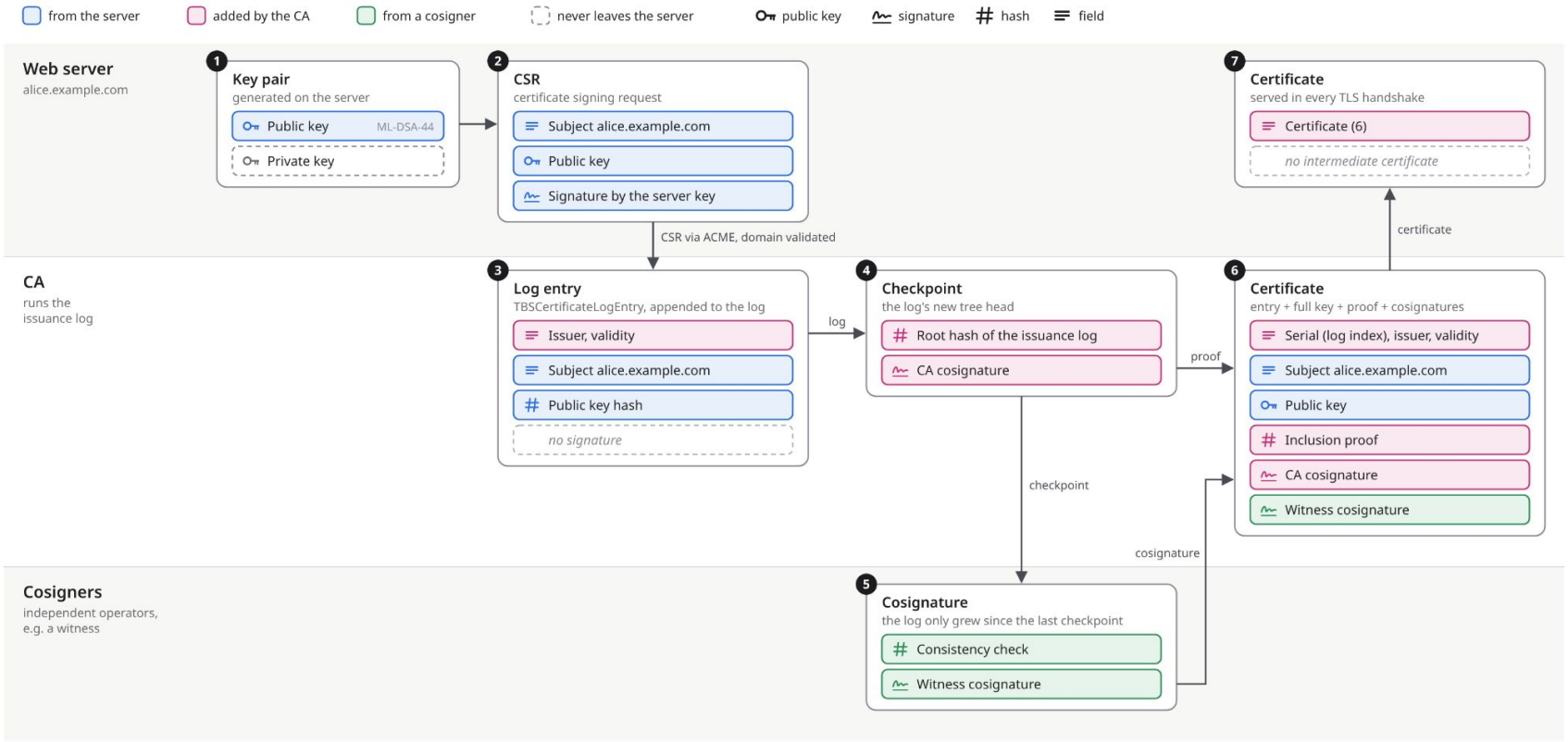
Merkle Tree

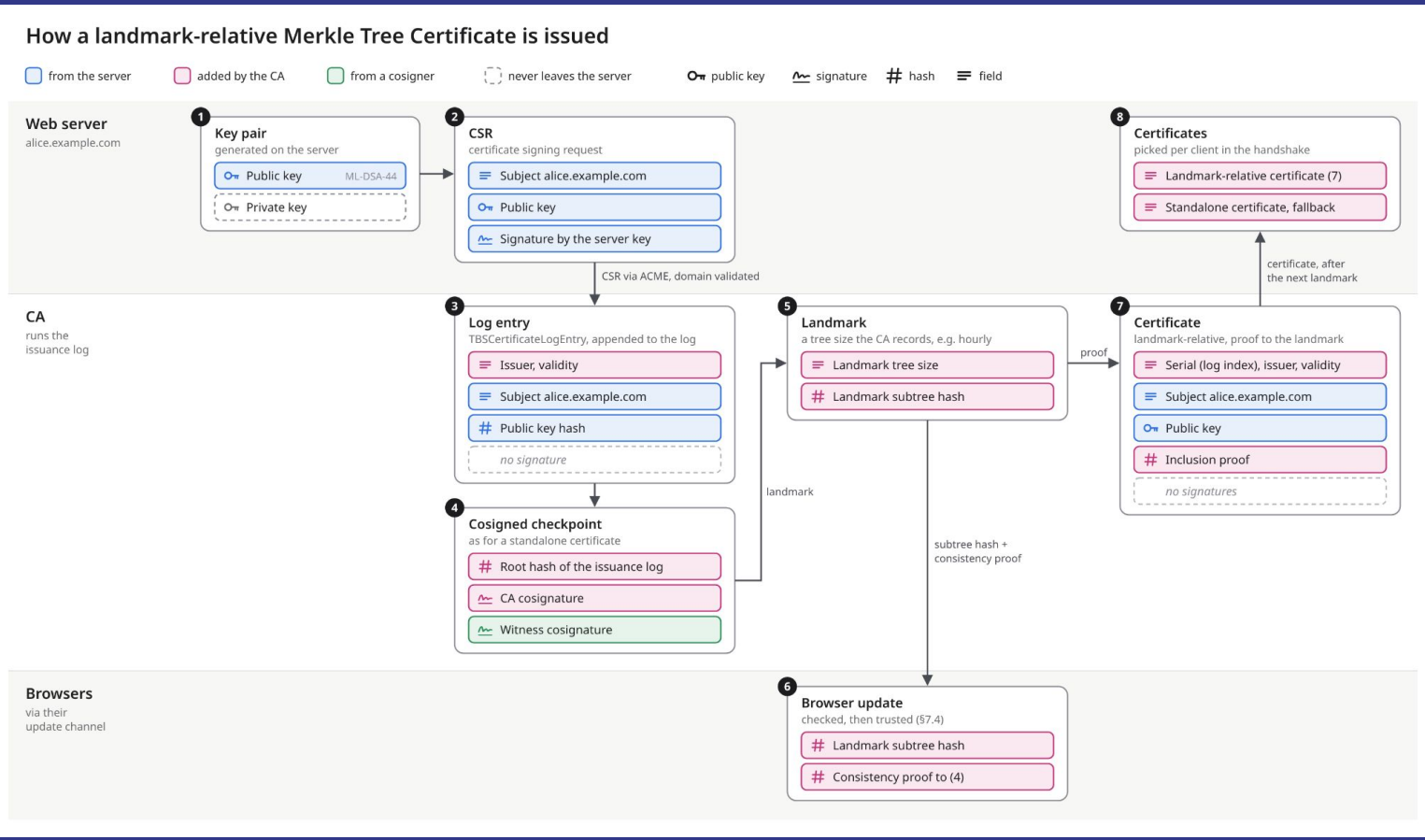
Demo Time!



controlplane

How a Merkle Tree Certificate is issued (standalone)




How a landmark-relative Merkle Tree Certificate is issued
from the server
added by the CA
from a cosigner
never leaves the server
public key
signature
hash
field
Web server
alice.example.com
1
Key pair
generated on the server
Public key
ML-DSA-44
Private key
2
CSR
certificate signing request
Subject alice.example.com
Public key
Signature by the server key
CSR via ACME, domain validated
8
Certificates
picked per client in the handshake
Landmark-relative certificate (7)
Standalone certificate, fallback
certificate, after
the next landmark
CA
runs the
issuance log
3
Log entry
TBSCertificateLogEntry, appended to the log
Issuer, validity
Subject alice.example.com
Public key hash
no signature
4
Cosigned checkpoint
as for a standalone certificate
Root hash of the issuance log
CA cosignature
Witness cosignature
5
Landmark
a tree size the CA records, e.g. hourly
Landmark tree size
Landmark subtree hash
proof
7
Certificate
landmark-relative, proof to the landmark
Serial (log index), issuer, validity
Subject alice.example.com
Public key
Inclusion proof
no signatures
landmark
subtree hash +
consistency proof
Browsers
via their
update channel
6
Browser update
checked, then trusted (§7.4)
Landmark subtree hash
Consistency proof to (4)

Takeaways

What have we learned today?

- The post-quantum crypto transition is well underway
  - 2030~2035 is a reasonable target
- 2025 saw a lot of adoption of ML-KEM due to transparent updates in many tools and libraries
  - Mitigates store-now-decrypt-later
  - The tail is long…
- Signatures remain the focus area to further adoption. New standards and more automation:
  - Merkle Tree Certificates (IETF PLANTS)
  - ACME Renewal Information (RFC 9773)
  - Shorter Lifetimes, 45 days by 2029

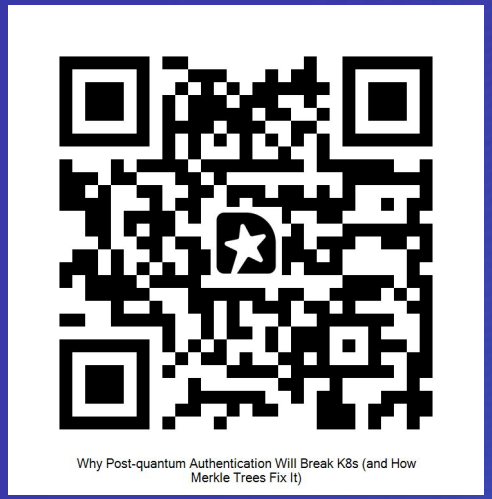
Sources to watch

Now with 100% less clickbait

- Bas Westerbaan @ Cloudflare
  - 2029 Migration Target
  - State of PQC 2025
- Scott Aaronson @ UT Austin
  - Shor's Algorithm
- Filippo Valsorda
  - ML-KEM implementation
  - ML-DSA implementation
- Jan Schaumann
  - Sites using PQC (Feb 2026)

Thanks so much for listening!

Any questions?



controlplane