Post-Quantum

Cryptography Conference

# The internet is ready for some PQC certificates

**Shane Kelly**
Principal Crypto Architect at DigiCert

KEYFACTOR CRYPTO4A SSL.com ENTRUST HID

**October 28 - 30, 2025 - Kuala Lumpur, Malaysia**

PKI Consortium

digicert®

# The Internet Is Ready For Some PQC Certificates

Shane Kelly
shane.kelly@digicert.com

October 28, 2025

# Objective

What are the key points?

1. Start using ML-DSA-44 now.
2. Start with leaf certificates.
3. Work your way up the chain until you think there will be an issue.
4. The future may provide some smaller alternatives.

# Problems with PQC PKI

We don't have nice replacements for ECC and RSA

## Too Big

A pure PQC certificate chain would transfer more than 14 kB of data.

Chrome calls even 7 kB an "implausible" amount to transfer without an actual quantum threat.

Each kB will slow down transactions. Especially for the smaller ones.

## Too Slow

More bytes means slower time to last byte.

Our smaller options take a lot more CPU time than current schemes.

## Too New

These schemes are new. Why replace classical schemes now when we trust them?

# WebPKI change is complicated

There are many different users and use cases. Some systems are very difficult to update. Some on are very old technology (slow).

We can take two things away from these insights:

1. Switching to a whole new PKI infrastructure is not reasonable in our timelines.
2. We must ignore the parts that “can’t” be updated. If you can’t update it’s not going to be secure.

Side Note

# TLS 1.3

- It’s usually implicit that TLS is being used.
- TLS 1.2 isn’t going to support PQC algorithms.
- We/our customers are going to have to use TLS 1.3
- A starting point for “How can we start our PQC journey”.

# >64%

of TLS connections are using TLS 1.3

– Cloudflare

# How can we use PQC in PKI?

What needs to be addressed to start using PQC?

- We have to trust it.
- Transferred data can’t be too big (for some value of big).
- The speed of running/transferring can’t be onerous.

# Now

What PQC can we use today?

(It’s ML-DSA)

# ML-DSA is Secure

For a reasonable definition of secure.

**It’s not new**

For some of us it feels like this thing was release yesterday. It’s actually almost 10 years old.

**Lots of eyes**

Lattices, specifically module-lattices, have come under intense scrutiny in the last 10+ years. There is a large body of academic and industrial research into its security.

**Standardization is arduous**

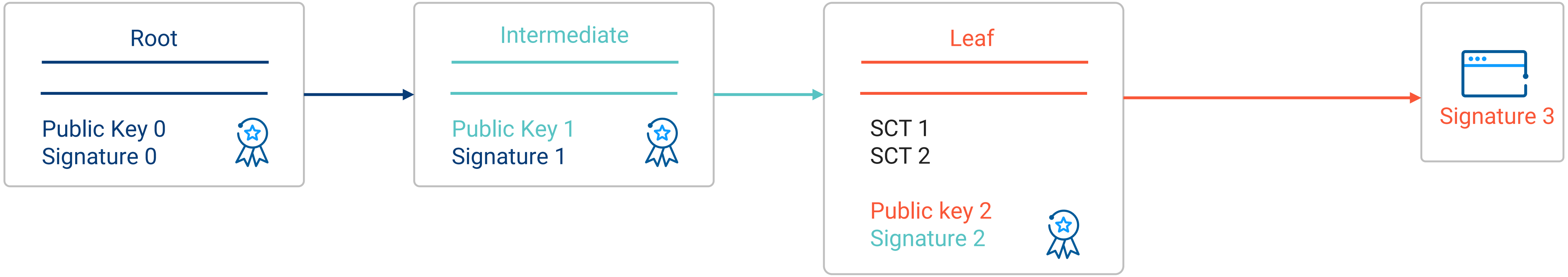
The NIST process may not be perfect, but it sure does subject these schemes to a lot of potential attacks.

# ML-DSA is big

| Scheme | Public Key | Signature | Normalized Signature |
|---|---|---|---|
| **RSA 2048** | 256 | 256 | 4 |
| **RSA 4096** | 512 | 512 | 8 |
| **ECC 25519** | 32 | 64 | 1 |
| **ML-DSA-44** | 1312 | 2420 | 38 |

For common certificate chains, 5 signatures and 2 public keys are transmitted.

# Common Cert Chain



Root

Public Key 0
Signature 0

Intermediate

Public Key 1
Signature 1

Leaf

SCT 1
SCT 2

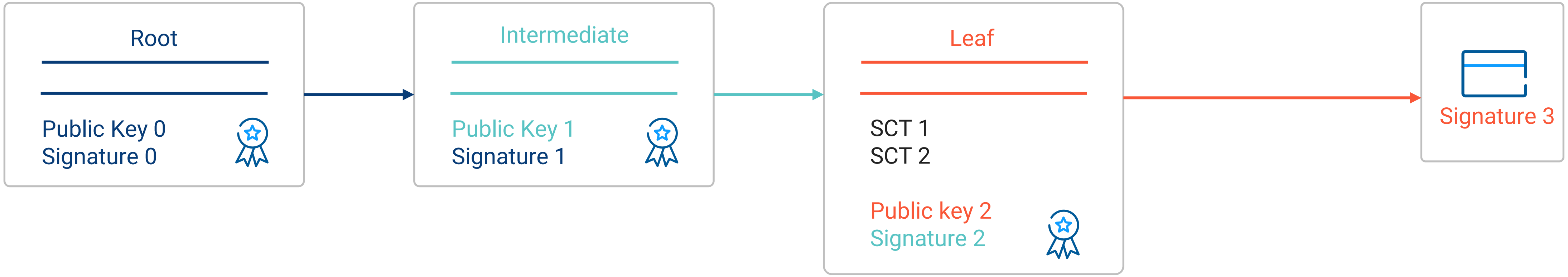
Public key 2
Signature 2

Signature 3

| Signature 1 | Public Key 1 | Signature 2 | Public Key 2 | Signature 3 | SCT 1 | SCT 2 |
|---|---|---|---|---|---|---|
| RSA-4096 | RSA-2048 | RSA-2048 | RSA-2048 | RSA-2048 | secp256r1 | secp256r1 |
| 512 | 256 | 256 | 256 | 256 | 64 | 64 |

**Total: 1664**

# ML-DSA Cert Chain



Root

Public Key 0
Signature 0

Intermediate

Public Key 1
Signature 1

Leaf

SCT 1
SCT 2

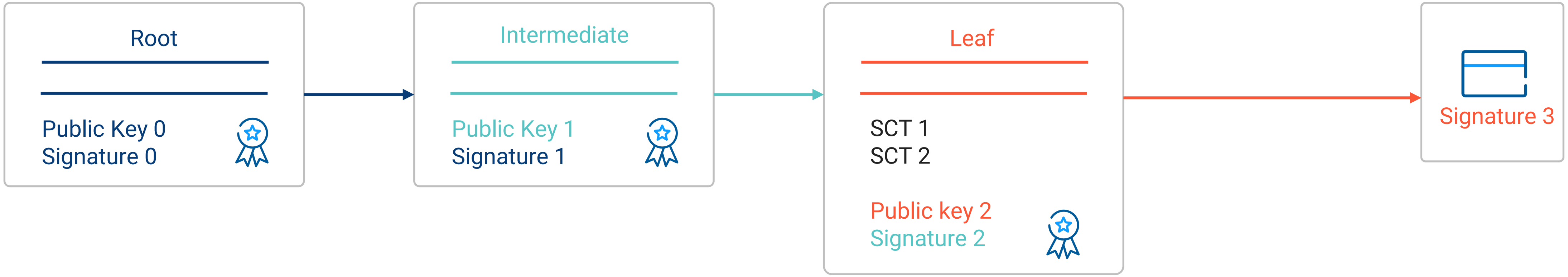
Public key 2
Signature 2

Signature 3

| Signature 1 | Public Key 1 | Signature 2 | Public Key 2 | Signature 3 | SCT 1 | SCT 2 |
|---|---|---|---|---|---|---|
| ML-DSA-44 | ML-DSA-44 | ML-DSA-44 | ML-DSA-44 | ML-DSA-44 | ML-DSA-44 | ML-DSA-44 |
| 2420 | 1312 | 2420 | 1312 | 2420 | 2420 | 2420 |

**Total: 14724**

# Big RSA Cert Chain



Root

Public Key 0
Signature 0

Intermediate

Public Key 1
Signature 1

Leaf

SCT 1
SCT 2

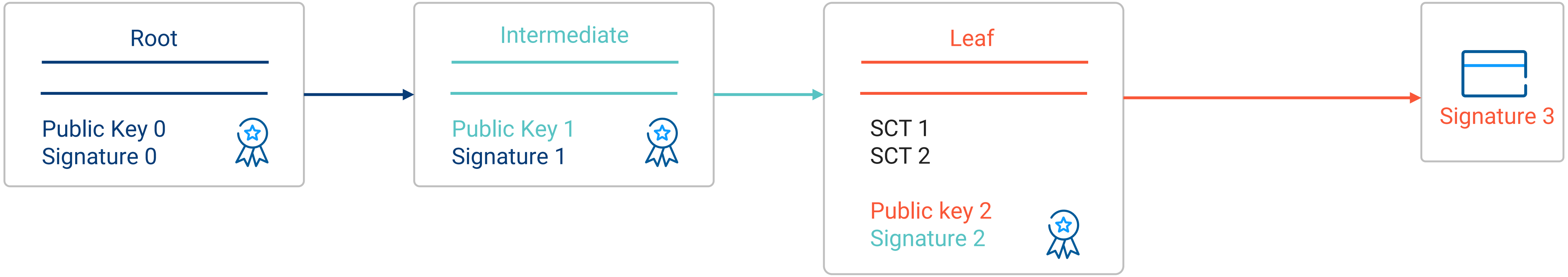
Public key 2
Signature 2

Signature 3

| Signature 1 | Public Key 1 | Signature 2 | Public Key 2 | Signature 3 | SCT 1 | SCT 2 |
|---|---|---|---|---|---|---|
| RSA-4096 | RSA-4096 | RSA-4096 | RSA-4096 | RSA-4096 | secp256r1 | secp256r1 |
| 512 | 512 | 512 | 512 | 512 | 64 | 64 |

**Total: 2688**

# ECC Cert Chain



Root

Public Key 0
Signature 0

Intermediate

Public Key 1
Signature 1

Leaf

SCT 1
SCT 2

Public key 2
Signature 2

Signature 3

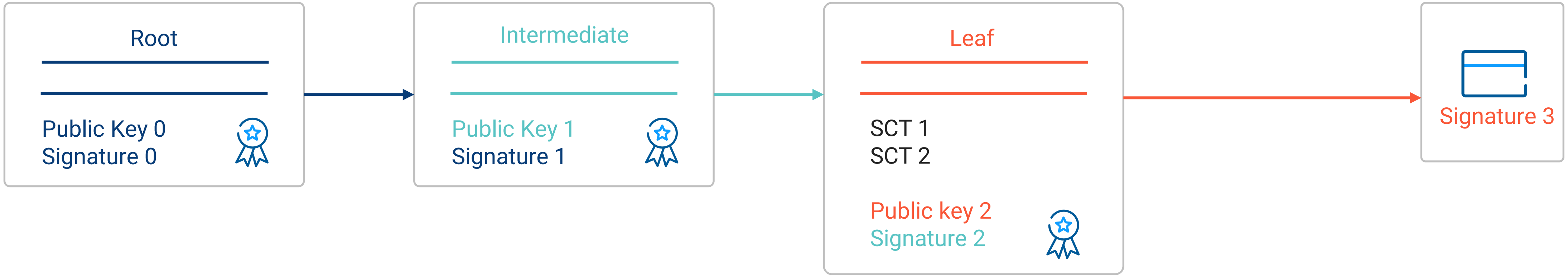
| Signature 1 | Public Key 1 | Signature 2 | Public Key 2 | Signature 3 | SCT 1 | SCT 2 |
|---|---|---|---|---|---|---|
| 25519 | 25519 | 25519 | 25519 | 25519 | secp256r1 | secp256r1 |
| 64 | 32 | 64 | 32 | 64 | 64 | 64 |

**Total: 384**

# What else can we try?

We don’t need all or nothing.

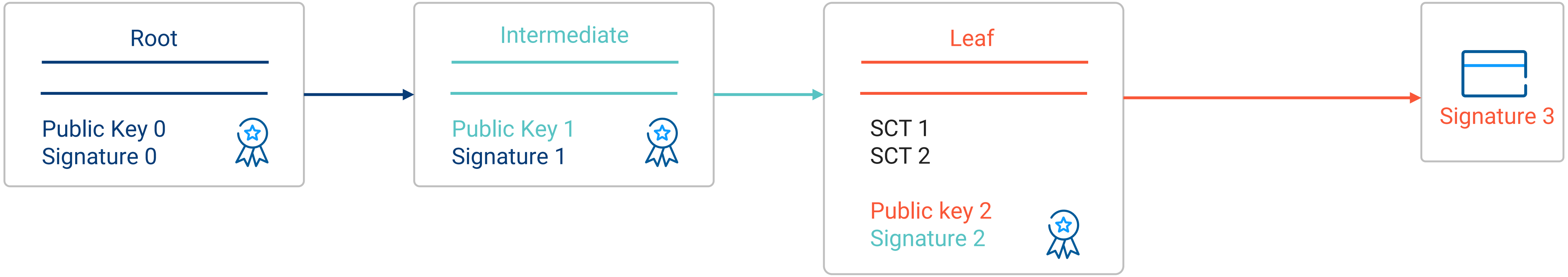
# Start at the Leaf



| Signature 1 | Public Key 1 | Signature 2 | Public Key 2 | Signature 3 | SCT 1 | SCT 2 |
|---|---|---|---|---|---|---|
| RSA-4096 | RSA-2048 | RSA-2048 | ML-DSA-44 | ML-DSA-44 | secp256r1 | secp256r1 |
| 512 | 256 | 256 | 1312 | 2420 | 64 | 64 |

**Total: 4484**

# Small PQC Cert Chain



| Signature 1 | Public Key 1 | Signature 2 | Public Key 2 | Signature 3 | SCT 1 | SCT 2 |
|---|---|---|---|---|---|---|
| 25519 | 25519 | 25519 | ML-DSA | ML-DSA | secp256r1 | secp256r1 |
| 64 | 32 | 64 | 1312 | 2420 | 64 | 64 |

**Total: 4020**

# Transmission Comparison

| Cert Chain | Size | Difference |
|---|---|---|
| **RSA** | 1664 | Reference |
| **Big RSA** | 2688 | 1024 |
| **ML-DSA Leaf** | 4484 | 2820 |
| **Small(er) ML-DSA Leaf** | 4020 | 2356 |

This is going to be acceptable for most use cases.

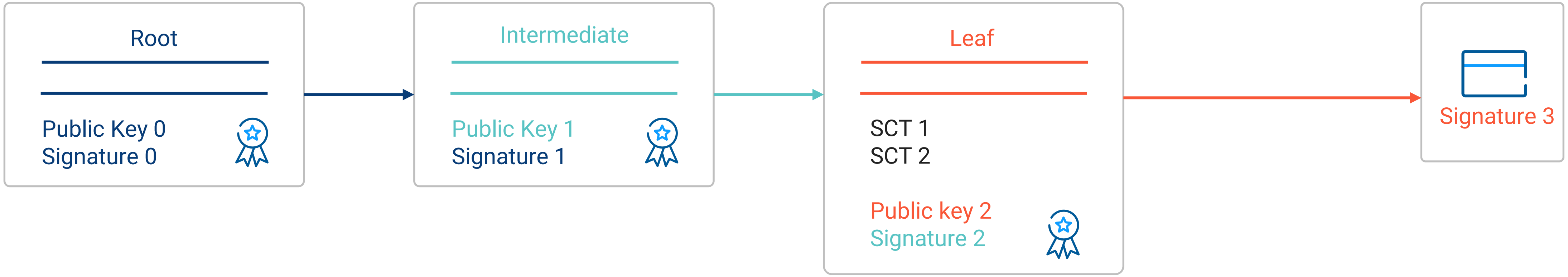
# The Number 47

Arbitrary? Yes!

Useful? Also, yes!

- Switching leaf certificates to have 47 day lifetimes is a great time to start swapping in ML-DSA-44 leaf certificates.
- We don't want to be revoking certificates just to switch algorithms. Use the natural lifecycle ending to switch to PQC.
- Having a 47 day limit means that you can run a test trial for ~47 days. If things prove problematic just issue a classic certificate while you address the problem.

# Bonus Points: Work Up the Chain



Root

Public Key 0
Signature 0

Intermediate

Public Key 1
Signature 1

Leaf

SCT 1
SCT 2

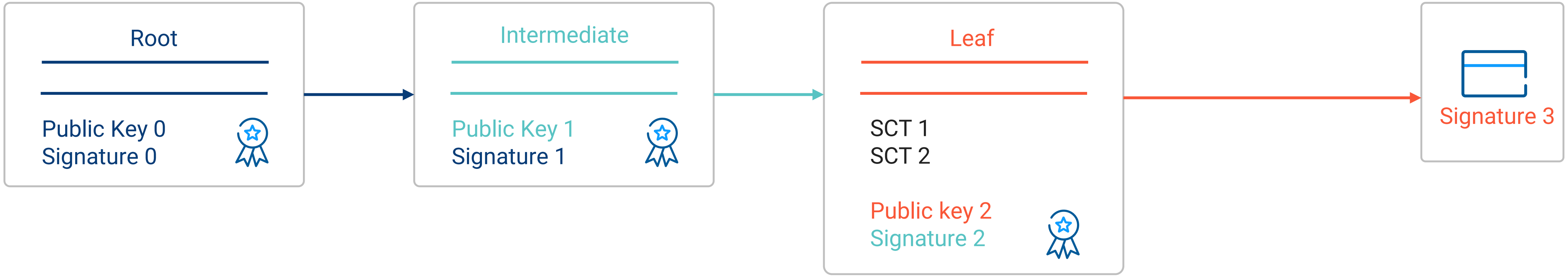
Public key 2
Signature 2

Signature 3

| Signature 1 | Public Key 1 | Signature 2 | Public Key 2 | Signature 3 | SCT 1 | SCT 2 |
|---|---|---|---|---|---|---|
| RSA-4096 | ML-DSA-44 | ML-DSA-44 | ML-DSA-44 | ML-DSA-44 | secp256r1 | secp256r1 |
| 512 | 1312 | 2420 | 1312 | 2420 | 64 | 64 |

**Total: 8104**

# And Up



| Signature 1 | Public Key 1 | Signature 2 | Public Key 2 | Signature 3 | SCT 1 | SCT 2 |
|---|---|---|---|---|---|---|
| ML-DSA-44 | ML-DSA-44 | ML-DSA-44 | ML-DSA-44 | ML-DSA-44 | secp256r1 | secp256r1 |
| 2420 | 1312 | 2420 | 1312 | 2420 | 64 | 64 |

**Total: 10012**

# The Future

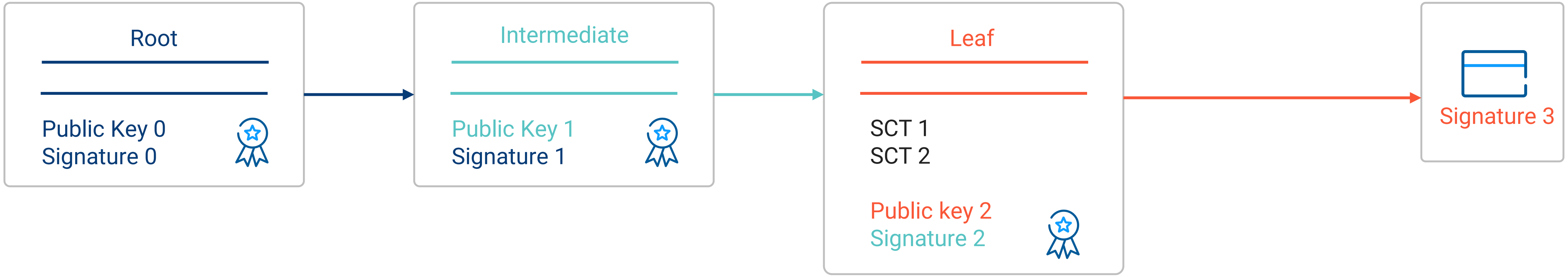
Let’s try and predict a future utopian certificate chain.

# NIST Additional Signatures

We’ll probably be getting additional special purpose PQC signature schemes in the next 5 years.

- There are no additional general purpose schemes.
- What we need to push for are the special purpose schemes that solve our size issue.
- SQIsign is slow but it has very attractive public key and signature sizes.
- Because SCTs don’t send the public key they are prime candidates for a multivariate scheme.

# Tortoise.0



Root

Public Key 0
Signature 0

Intermediate

Public Key 1
Signature 1

Leaf

SCT 1
SCT 2

Public key 2
Signature 2

Signature 3

| Signature 1 | Public Key 1 | Signature 2 | Public Key 2 | Signature 3 | SCT 1 | SCT 2 |
|---|---|---|---|---|---|---|
| $\text{SQIsign}_1$ | $\text{SQIsign}_1$ | $\text{SQIsign}_1$ | ML-DSA | ML-DSA | $\text{QRUOV}_{(127, 156, 54, 3)}$ | $\text{QRUOV}_{(127, 156, 54, 3)}$ |
| 148 | 65 | 148 | 1312 | 2420 | 200 | 200 |

**Total: 4493**

# That’s not so bad

4493 Bytes

- Remember the RSA chain with an ML-DSA leaf cert was 4484 bytes.
- The Small PQC Chain with ECC was 4020 bytes.
- This sits somewhere between ECC and RSA + ML-DSA in terms of size.
- “Start with the leaf certificates!”
- The size of a Photosynthesis solution would be 4500 bytes.

# It’s not perfect either

All the math!

- Even with improvements to SQIsign, it’s going to be slower than an all ML-DSA chain.
- However, transfer size is the problematic factor.
- There are three different types of algos in this chain. The code complexity and code size is very high.
- Multivariate …

# Summary

What are the key takeaways?

- Use ML-DSA-44 now, wherever you can get it to work.
- For PKI start with the leaf certificates.
- Test then move up the certificate chain if you are comfortable.
- If SQIsign and a multivariate scheme survives we can send “only” 4.5 k.

# What can you do?

- Contact your local IETF and CA/B forum representative and ask them to approve ML-DSA now.
- If you are using a private CA, start using ML-DSA-44 now.
- Shift the mindset from ML-DSA being experimental and PQC-specific to being the current best tool for signing leaf certificates.
- Work on being able to change certificate algorithms quickly. Use the 47 day transition as a way to verify you can quickly switch algorithms.
- Get comfortable with mixed algorithm certificate chains.
- If you want to play around with PQC certs: https://www.digicert.com/digicert-labs

digicert®

www.digicert.com