sectigo

Merkle Tree Certificates

The architecture behind fast, quantum-safe authentication

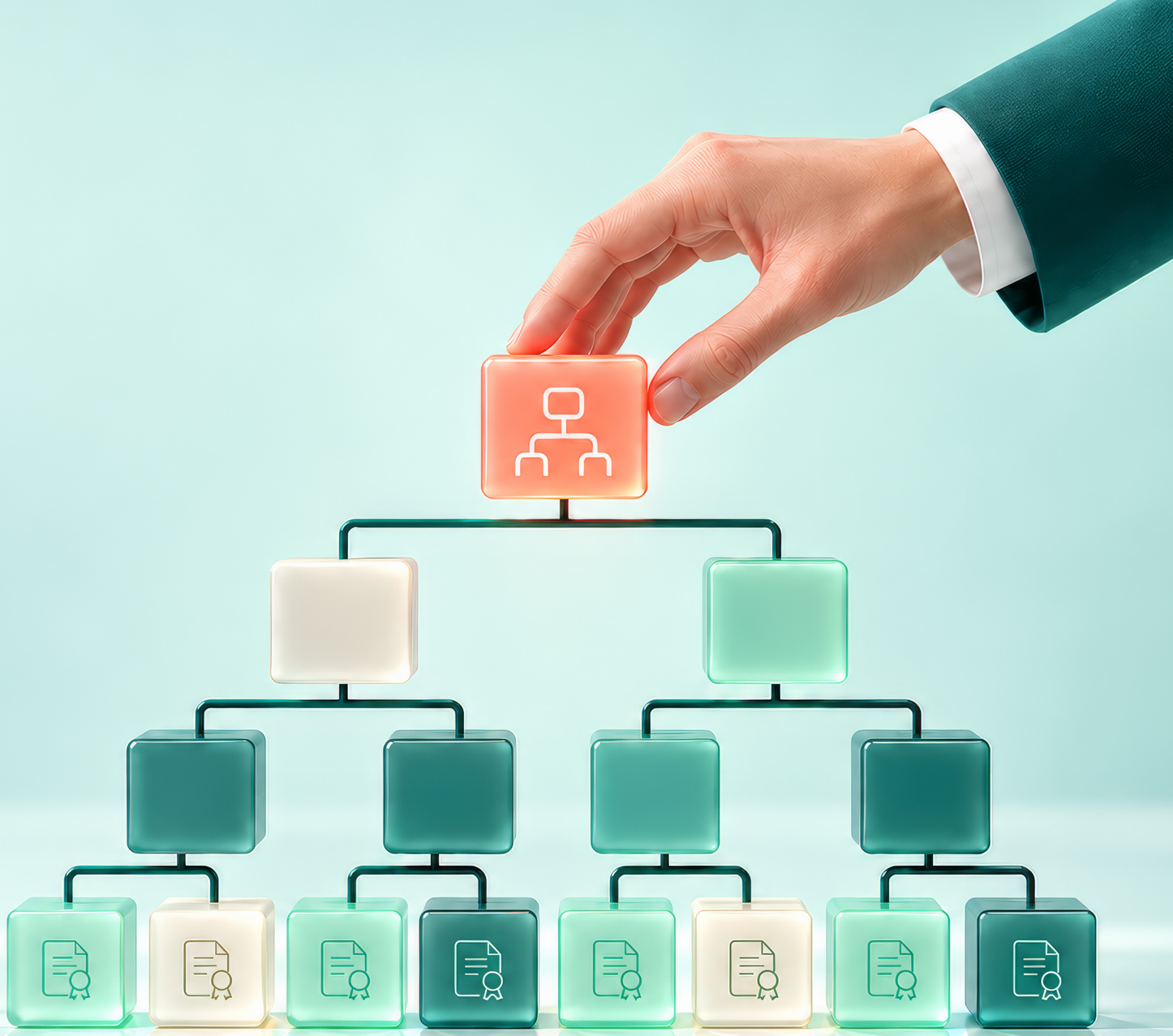


Table of Contents

Introduction 3

Chapter 1 4
The quantum clock is already running

Chapter 2 7
The hidden cost of post-quantum signatures

Chapter 3 9
What is a Merkle Tree Certificate?

Chapter 4 11
Transparency by design

Chapter 5 12
Who's building this, and why it's credible

Chapter 6 13
What changes (and what doesn't) for organizations

Chapter 7 15
Getting ready: Crypto agility as the real prerequisite

Chapter 8 18
What MTCs mean for your certificate management

Chapter 9 19
What to watch next

Closing 20
Sectigo's position

Appendix 21
Glossary

Introduction

Every website that runs HTTPS makes a promise: the site you reached is the site it claims to be. That promise is backed by a digital certificate, and for three decades the math behind it has been quietly reliable. Quantum computers are about to change that math, and the industry has already started moving.

The risk is not hypothetical, and it is not waiting for a working quantum computer to arrive. Adversaries can record encrypted traffic today and decrypt it later, once a sufficiently powerful quantum computer exists; a strategy known as harvest-now, decrypt-later. Data with a long shelf life, financial records, health data, government communications, and long-lived credentials, is exposed the moment it is captured, not the moment it is broken.

Organizations are starting to feel that urgency directly. In Sectigo's 2026 State of Crypto Agility research, 78% of senior IT and cybersecurity decision-makers say today's public key infrastructure will need to change to a great or very great extent to support post-quantum cryptography, not just swap in a new algorithm and move on.





That urgency is why the Internet Engineering Task Force, the body that governs the technical standards the internet runs on, has a working group actively building a new certificate architecture designed for the post-quantum era. It's called a Merkle Tree Certificate, or MTC.

A Merkle Tree Certificate is a new kind of website domain certificate designed to keep secure connections fast even as the signatures behind them grow to resist quantum computers. Instead of asking every certificate to carry its own heavy, quantum-resistant signature, an MTC lets a certificate authority record everything it issues in a public, tamper-evident log, organized as a Merkle tree. The website then presents a short trail of digital fingerprints proving its certificate sits in that log, along with signatures vouching for the log or, for up-to-date browsers, nothing more than the trail itself, checked against log data the browser already holds. The result: a certificate that stays small, resists quantum attacks, and is a certificate that stays small, resists quantum attacks, and is built for public verifiability..

This ebook walks through why that architecture is necessary, how it works, who is building it, and what it means for how organizations manage certificates going forward. Spoiler: the more certificates change shape and cadence, the more automation stops being optional.

The quantum clock is already running

Harvest-now, decrypt-later makes quantum readiness, something organizations used to consider a future problem, a present-day data protection threat. Anything encrypted today with RSA and ECC could be exposed the moment a cryptographically relevant quantum computer exists, regardless of when that day arrives. For data that needs to stay confidential for years, the deadline to act was, in a real sense, yesterday.

Governments and standards bodies have now responded with concrete migration timelines rather than vague warnings. In the United States, federal guidance has pushed agencies and their vendors toward measurable post-quantum migration milestones. Google and Cloudflare, two of the organizations doing the most hands-on engineering in this space, have both pointed to 2029 as a target horizon for meaningful post-quantum deployment on the public web.



Inside the certificate world specifically, the CA/Browser Forum, the industry body that sets the rules browsers and certificate authorities follow, is already compressing maximum certificate lifetimes toward 47 days by 2029. That is no longer a future milestone: the first phase of the reduction, 200-day certificates, is already in effect, with 100-day certificates arriving in March 2027. That timeline is running on its own accelerated schedule, independent of quantum computing, but the two forces compound each other: certificates are being asked to rotate far more often, at the exact moment their signatures are about to get much bigger.

The two pressures compound: certificates must rotate more often, right as their signatures are about to get much bigger.

The pressure isn't only coming from the CA/Browser Forum's own schedule, either. According to Sectigo's 2026 State of Crypto Agility report, organizations increasingly don't control their own PQC timeline: governments, regulators, standards bodies, and major technology providers keep accelerating expectations around migration. Among organizations already tracking PQC developments, 70% say they're aware of Google's own accelerated migration roadmap, which leaves a meaningful 30% still catching up to a moving deadline they don't set.

Why "just swap in bigger keys" doesn't work

The intuitive fix, replace today's signature algorithms with post-quantum ones and move on, sounds simple. It isn't. Post-quantum signature algorithms are not modest upgrades; their signatures and public keys are meaningfully larger than the classical ones they replace. At internet scale, a full TLS handshake carrying full post-quantum signature chains adds real weight to every single secure connection made anywhere on the web, billions of times a day. Bandwidth-constrained mobile networks, satellite links, and IoT devices feel it first and hardest. The industry needed an approach that solved for scale, not just for algorithm strength, and that's the problem Merkle Tree Certificates were built to solve.



The hidden cost of post-quantum signatures

To understand why Merkle Tree Certificates exist, it helps to look at the actual numbers behind a post-quantum signature. ML-DSA, the NIST-standardized, lattice-based signature algorithm that is emerging as the leading replacement for today's RSA and ECDSA signatures, produces signatures of roughly 2.4KB at its smallest parameter set, more than 30 times larger than a classical ECDSA signature of around 64 bytes.



A single signature is one thing. A certificate chain is another. A fully post-quantum TLS handshake using ML-DSA-44 carries public keys, CA signatures, two Certificate Transparency signatures, and a handshake signature, roughly 14,700 bytes in total. Merkle Tree Certificates replace three of those signatures, the CA's and both transparency signatures (7,260 bytes), with an inclusion proof as small as 736 bytes in the draft's example. That's almost ten times smaller. The public key and handshake signature remain, so the full exchange shrinks by around 70%.

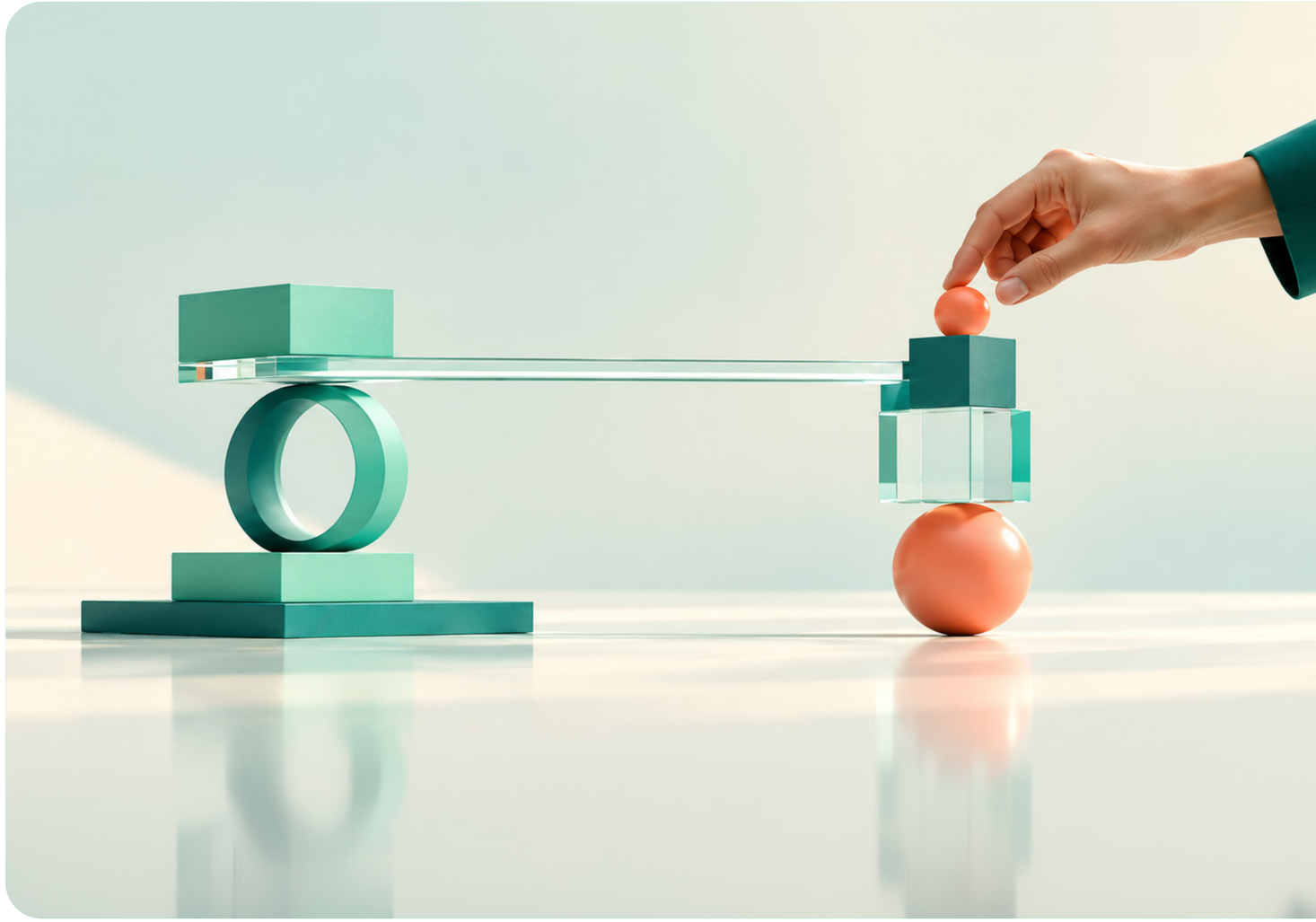
What that weight actually costs

Bandwidth: full certificate-authenticated handshakes carry the extra payload.

Latency: mobile and high-latency networks feel the extra round-trip weight disproportionately, and it lands hardest on the users least able to absorb it.

Infrastructure breakage: legacy servers, load balancers, and middleboxes with hardcoded buffer sizes or handshake assumptions can fail outright when certificate messages grow past what they expect.

Taken together, these costs make one thing clear: simply swapping classical signature algorithms for post-quantum ones inside the existing certificate model doesn't scale. The industry didn't need a bigger version of the same architecture. It needed a different architecture, one where the certificate no longer has to carry the full weight of proving itself on every connection. That's the thesis behind Merkle Tree Certificates.





What is a Merkle Tree Certificate?

The core idea behind a Merkle Tree Certificate is a shift from signing every certificate individually to batching certificates into a single, signed structure: a Merkle tree. Instead of a certificate authority producing one heavy, standalone signature per certificate, it commits to a batch of certificates at once by signing a compact snapshot of its log, known as a checkpoint, and the subtrees that cover each new batch.

A handful of hashes instead of a chain of signatures

Once a certificate is included in that tree, proving it belongs there doesn't require redoing the certificate authority's full signature. It only requires an inclusion proof: a short sequence of hash values, sometimes described as a handful of digital fingerprints, that lets anyone reconstruct the path from that one certificate up to the publicly known tree head. Verifying the proof is fast, and, critically, it stays small no matter how large the underlying post-quantum signature algorithm gets.

Picture a family tree of hashes: each certificate is a leaf, pairs of leaves combine into a branch, branches combine into bigger branches, and the whole structure collapses into a single value at the top. Proving your certificate belongs in the tree only takes the handful of sibling values along your specific branch, not the whole tree.

The size math

This is where the architecture earns its keep. In the streamlined, landmark-relative case, the 7,260 bytes of signatures that vouch for a certificate today give way to an inclusion proof of under 1KB. The exact size depends on how many certificates the tree covers. The server's public key, certificate fields, and handshake signature still travel with every connection, so the total exchange drops from roughly 14,700 bytes to around 4,500. Post-quantum authentication doesn't become free. But a cost that would have taxed every connection on the internet becomes one the web can realistically absorb.

It's worth being precise about what makes this possible: the certificate authority still signs, but it signs batches, not individual certificates. A single signed subtree can cover thousands of certificates, and a landmark can cover millions. The expensive operation happens once per batch instead of once per certificate, and what the website presents to a visiting browser is the cheap part: the proof that its certificate is one of the leaves underneath that signed root.



Transparency by design

Certificate Transparency, the existing system of public logs that record issued certificates so anyone can audit them, has done real work catching misissued and fraudulent certificates over the last decade. But it was layered onto the certificate system after the fact. A certificate authority signs a certificate first, then submits it to logs, and the system relies on separate enforcement to make sure that second step happens. Transparency is checked rather than built in.

The MTC inversion

Merkle Tree Certificates change that sequence. Issuance and logging are no longer two separate steps that can drift apart; they are the same step. A certificate cannot be a valid Merkle Tree Certificate without already being in the CA's issuance log, because the inclusion proof that makes it usable only exists once the certificate has been added to the tree.

That closes a real gap in today's model, where logging happens after issuance and is enforced through policy. But being in a log isn't the same as being publicly visible. Whether monitors can actually retrieve and audit a log's contents still depends on the log staying available. It also depends on independent cosigners vouching that the log is operated consistently, and relying parties set both through their own policies. The draft also permits CA-only validation for applications without transparency requirements, such as some private deployments. Where browser policy requires independent cosigners, MTCs make transparency far harder to skip. It's a stronger structural foundation, not a guarantee that holds on its own.



Who's building this, and why it's credible

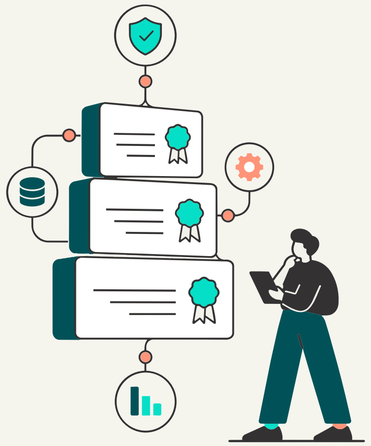
Merkle Tree Certificates are not a single vendor's proposal circulating for feedback. The specification, drafted as draft-ietf-plants-merkle-tree-certs, moves through the IETF's PLANTS working group, PKI, Logs, and Tree Signatures, chartered specifically to reduce the overhead post-quantum signatures create for certificate transparency-backed public key infrastructure.

The authorship reflects the parties with the most at stake in getting this right: engineers from Google, Apple, and Cloudflare, alongside independent cryptographic contributors, co-author the draft, and Sectigo is engaged in that same standards conversation as a certificate authority with direct operational stake in how issuance changes under this model.

A signal, not a side project

Browser vendor involvement matters here in a specific way. When the organizations that decide what a browser will and won't trust are directly engineering the replacement architecture, that's a strong signal about where the industry's post-quantum certificate strategy is actually heading, not just where one vendor wants it to head.

This also isn't purely theoretical work sitting in a specification document. Live feasibility testing against real internet traffic is underway, the kind of testing that surfaces the operational rough edges a whiteboard design can't. That combination, a multi-stakeholder standards process, direct browser vendor engineering investment, and testing against real traffic, is what separates a credible architectural shift from a proposal that never leaves the drafting stage.



What changes (and what doesn't) for organizations

It's worth being direct about scope: Merkle Tree Certificates are a coexistence model, not a replacement event. Today's directly-signed certificates are not going away overnight, and organizations do not need to treat this as an imminent cutover.

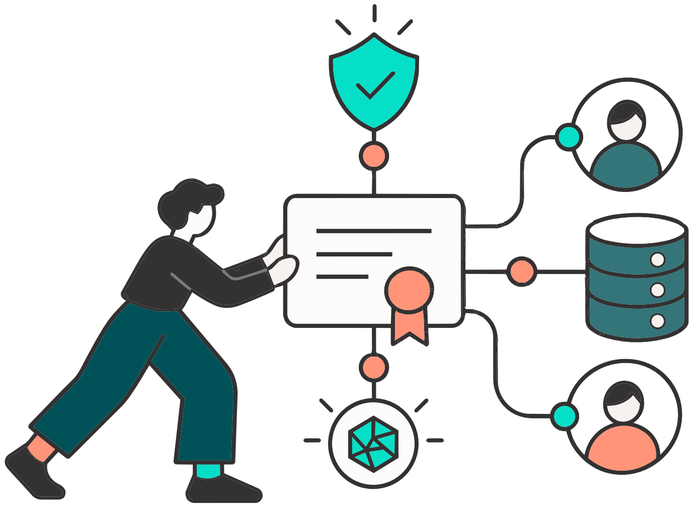


Two forms, one operational reality

Part of what makes MTCs practical for a gradual rollout is that they come in two forms. A standalone certificate carries its inclusion proof along with signatures from the certificate authority and independent cosigners, so it can be issued with minimal delay and works for any client that trusts those parties. A landmark-relative certificate drops the signatures entirely for an even smaller footprint, but it only becomes available once the log's next landmark, a periodic checkpoint the log publishes, covers it, and it only works for browsers whose predistributed landmark data is current.

Servers are expected to hold both forms simultaneously, and each TLS connection negotiates which one to present based on what the connecting client supports. That's a meaningfully different operational posture than swapping one certificate for another: it's maintaining two live certificate types per domain, tracking which one a given landmark cycle makes available, and serving the right one on the fly.

This kind of added operational complexity is landing on organizations that are already stretched thin managing the certificates they have today. Sectigo's 2026 State of Crypto Agility report finds that 61% of organizations experienced a certificate-related outage or incident in the past year, and 61% remain very or extremely concerned about the business impact of shorter certificate lifespans, up from 52% in 2025. Confidence is climbing alongside that concern, complete certificate inventory jumped from 28% to 62% year over year, but the two data points together tell an honest story: organizations are getting better prepared while the ground keeps shifting under them.



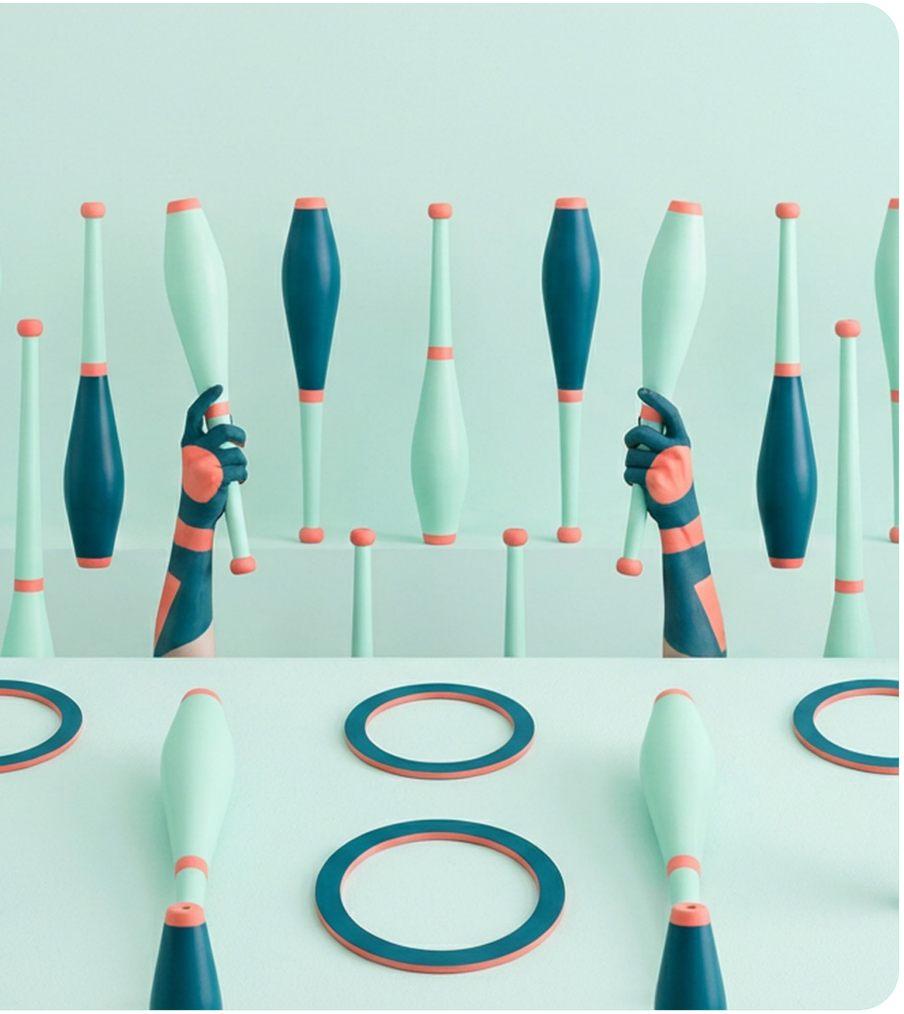
What this means, by role

Certificate authorities: take on issuance log operation, batch-signing, collecting cosignatures from independent cosigners, and allocating and publishing landmarks, on infrastructure that can update Merkle trees at issuance speed.

Browsers and relying parties: need to ingest and keep current the landmark data that lets them verify landmark-relative certificates, and need handshake logic that can negotiate between certificate forms.

Organizations running servers: need visibility into which certificate form is live, when the next landmark update lands, and whether their infrastructure is current enough to serve the smaller, signatureless form when it's available.

The early momentum behind MTCs is in the public web PKI, where browser vendors are driving adoption. The specification itself also accommodates private deployments, such as an organization's internal services, and the same crypto-agility discipline applies there too.



Getting ready: Crypto agility as the real prerequisite

Step back from the specifics of Merkle Tree Certificates for a moment, and a broader pattern emerges. Certificates are arriving in stages. They're getting swapped as landmarks roll forward on a cadence measured in on a frequent cadence that will make manual issuance and renewal unfeasible. And they carry lifetimes the CA/Browser Forum is already shrinking toward 47 days by 2029. Manual renewal was already showing its age before quantum computing entered the picture; against this cadence, it's not a viable operating model.

That's true regardless of which post-quantum certificate format ultimately wins broad adoption. Whether the industry converges fully on Merkle Tree Certificates, settles on a different hybrid approach, or runs multiple formats in parallel for years, the organizations in the strongest position share the same three characteristics: they have complete visibility into every certificate they've issued, they can inventory and classify that certificate population instantly, and they can rotate certificates through automation rather than through a person remembering a renewal date.

The data backs up how much ground most organizations still have to cover on that third point. Per Sectigo's 2026 State of Crypto Agility report, only 4% of organizations describe their certificate

management as fully automated today, even though 73% plan to increase certificate automation specifically because of shorter certificate lifespans. Renewals remain the most automated task, but every other certificate management function, discovery, issuance, revocation, reporting, showed year-over-year automation growth. The direction is right; the finish line is still a long way off for most.

The same report finds that organizations are starting to see these initiatives as one effort rather than several: 96% see at least some overlap between their preparedness for shorter certificate lifespans and their readiness for post-quantum cryptography. That overlap is the practical case for treating crypto agility as core infrastructure rather than a series of separate projects.

One investment, three returns: 47-day readiness, machine identity management, and post-quantum readiness all draw on the same automated foundation.





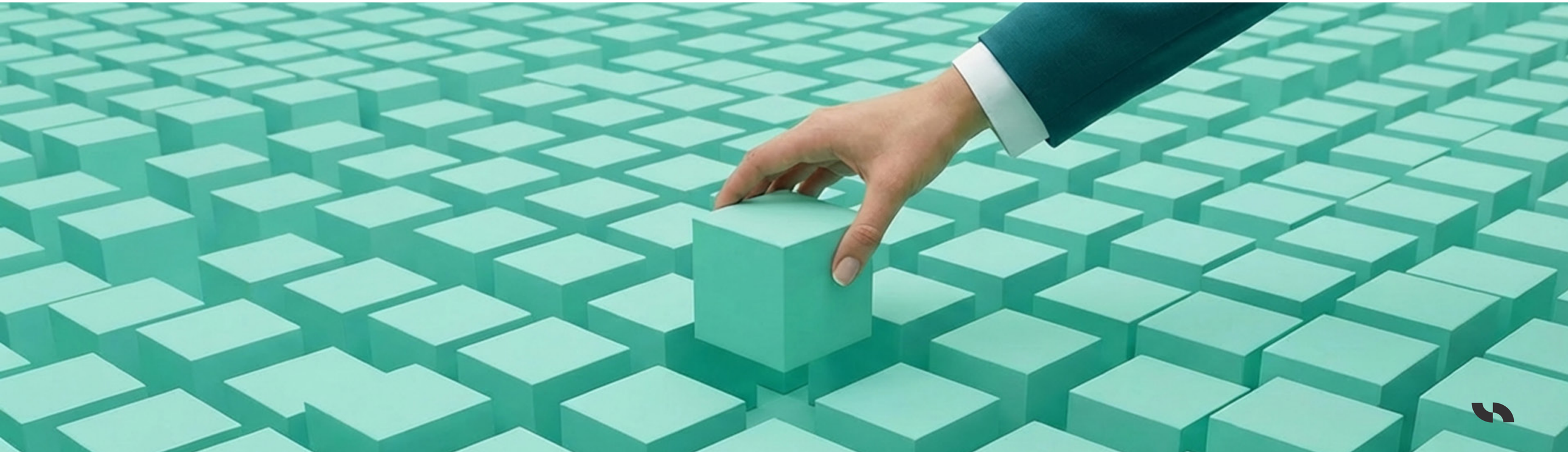
Practical readiness checklist

- Inventory every certificate your organization has issued or relies on, including ones issued outside central IT's visibility.
- Automate issuance and renewal end to end, so no certificate's validity depends on a person remembering a date.
- Build monitoring for certificate and log health so a missed rotation or a transparency gap surfaces immediately, not after an outage.
- Treat crypto agility as an operating capability, not a one-time project, so the next algorithm or certificate format change doesn't require rebuilding the process from scratch.

This is the connective tissue between the 47-day mandate, machine identity management, and post-quantum readiness. They read as three separate initiatives on a roadmap, but they draw on the exact same underlying capability: automated, policy-driven certificate lifecycle management that doesn't depend on manual intervention to keep pace.

What MTCs mean for your certificate management

Merkle Tree Certificates (MTCs) are likely to change how certificates are managed, not just how they're built. Under the MTC model, a subscriber may receive more than one form of certificate over time: a larger certificate that can be used right away, and a smaller, more efficient one that arrives later. The subscriber must deploy the right certificate at the right moment, so manual handling becomes difficult or even impractical. The current draft defines extensions to ACME, the protocol behind most automated certificate issuance today, to deliver both forms. Those details may still shift as the specification matures, but the direction is clear: **organizations planning to adopt MTCs should expect certificate lifecycle management and automation to be essential, and the days of manual certificate renewal may be coming to an end.**



What to watch next

Merkle Tree Certificates are moving through an active standards process, not a finished spec waiting for adoption. A few signals are worth tracking over the coming year:

NIST algorithm standardization status: continued guidance and any updates to the standardized post-quantum signature algorithms that MTCs and other post-quantum efforts build on.

IETF specification maturity: as of October 2026, draft-ietf-plants-merkle-tree-certs is at revision -07 and remains a working group document. Each new revision can shift implementation details that matter operationally.

Browser roadmap signals: public statements and shipped code from major browser vendors indicating timelines for supporting Merkle Tree Certificates in production, not just in test channels.

None of these signals require an organization to wait before acting. The infrastructure work, visibility, automation, and inventory, is valuable under any outcome these signals reveal.



Closing: Sectigo's position

Sectigo is tracking and directly contributing to the standards conversation shaping Merkle Tree Certificates, because the organizations we serve shouldn't have to choose between staying current on emerging PKI architecture and keeping their day-to-day certificate operations running. Those are the same job.

Sectigo's own 2026 State of Crypto Agility research puts a number on the trajectory: 38% of organizations have now conducted a full assessment of their quantum-vulnerable systems, up from just 14% in 2025, and 83% have budget allocated to quantum-safe security initiatives in the year ahead. Progress is real. But as that same research makes clear, progress and operational resilience aren't yet the same thing, and closing that gap is exactly the work Sectigo is built to help with.

Whatever certificate format ultimately carries the industry through the post-quantum transition, the operational foundation underneath it doesn't change: visibility into every certificate, automated issuance and renewal, and the crypto agility to adapt when the next standard shifts. That foundation is what Sectigo Certificate Manager is built to provide today, ahead of the mandates that will make it non-negotiable.

If your organization is evaluating where it stands on 47-day readiness, machine identity management, or post-quantum crypto agility, that's a conversation worth having now, while there's still runway to build the automated foundation rather than retrofit it under deadline pressure.



Appendix

Merkle tree	A tree-shaped data structure where each leaf represents one item (here, a certificate) and each parent node is a hash of its children, collapsing an entire batch of items into a single root value.
Inclusion proof	A short sequence of hash values that lets anyone verify a specific certificate is included within a given checkpoint or subtree, without needing the rest of the tree.
ML-DSA	Module-Lattice-Based Digital Signature Algorithm; the NIST-standardized (FIPS 204) post-quantum signature algorithm, formerly known as CRYSTALS-Dilithium.
Cosigner	A party that signs snapshots of a CA's issuance log to vouch that it is operating correctly. The CA operates one; independent cosigners add assurance beyond trusting the CA alone.
Transparency log	A public, append-only record of issued certificates that anyone can audit, letting domain owners and researchers detect misissued or unauthorized certificates.
Landmark	A periodic checkpoint a Merkle tree log publishes; landmark-relative certificates only become usable once a browser's predistributed data covers the relevant landmark.
Issuance log	An append-only log a certificate authority maintains of every certificate it issues. Under MTCs, a certificate is only valid once it is in the log.

Further Reading

- Sectigo, State of Crypto Agility Report 2026
- Root Causes podcast, episode 611
- Root Causes podcast, episode 616
- Root Causes podcast, episode 618
- IETF PLANTS working group draft: draft-ietf-plants-merkle-tree-certs

About Sectigo

Sectigo is a leading provider of Certificate Lifecycle Management (CLM), helping organizations secure human and machine identities while reducing the hidden risks of modern digital trust. As certificate volumes grow and lifespans shrink, unmanaged certificates are becoming a source of outages, compliance exposure, and operational disruption. Sectigo uses Orchestrated Automation to deliver Simplicity at Scale through its cloud native CLM platform, bringing visibility, control, and lifecycle management together across complex environments. With more than 700,000 customers, including 65% of the Fortune 500, Sectigo is one of the largest and longest standing certificate authorities, with over 20 years of delivering unparalleled digital trust. For more information, visit www.sectigo.com or follow us on LinkedIn.

© 2026 Sectigo. All rights reserved.