



Identity Refined at the Quantum Gate: Framing the AI + Post-Quantum Challenge for IAM

Sravanakumar Nidamanooru*

IAM Specialist, Manchester, UK

*Correspondence: Sravanakumar Nidamanooru, IAM Specialist, Manchester, UK, E-mail: sravana.nidamanooru@gmail.com; DOI: <https://doi.org/10.56147/aaiet.1.6.78>

Citation: Nidamanooru S (2025) Identity Refined at the Quantum Gate: Framing the AI + Post-Quantum Challenge for IAM. J Adv Arti Int Eng & Techn 1: 78.

Abstract

Identity & Access Management (IAM) is being reshaped by two concurrent forces: (i) the use of Artificial Intelligence (AI) to turn rich telemetry into policy decisions and (ii) the migration to Post-Quantum Cryptography (PQC) across credentials, certificates and protocol touchpoints. We argue that the most consequential risks live in the seams account recovery/reset, Non-Human Identities (NHIs) and crypto-agile upgrades where attackers concentrate and operations are fragile. This paper contributes four things. First, it frames a precise problem statement that links AI decisioning pipelines (signals → models → policies → evidence) with PQC adoption realities (artifact sizes, timing, interop and governance). Second, it organizes the landscape into a literature/practice map for AI-in-IAM and PQC-in-IAM that practitioners can immediately use for scoping. Third, it reports results from three small, reproducible experiments designed for teaching and early planning: A policy-level risk simulation spanning sign-in and recovery (risk-based control stops ~81% of simulated fraud vs. ~63% for a static baseline while reducing legitimate friction from ~38% to ~25% at ~+50 ms p95 decision latency); an overhead model for PQC artifacts showing modest size-driven latency on typical enterprise links (incremental over RTT on the order of ~1–6 ms); and a micro-pilot comparing passkeys to password+OTP (median sign-in time ~7.2 s vs. ~12.1 s; completion ~97% vs. ~92%; support tickets 6 vs. 10 per 100 users). Finally, it outlines a research agenda for recovery governance, machine identity attestation and rotation, crypto-agile policy engines and explainability/appeals. All datasets are synthetic to enable easy replication without sensitive data.

Keywords: Identity and Access Management (IAM); Post-Quantum Cryptography (PQC); Artificial Intelligence (AI); Risk-Based Authentication (RBA); Account recovery; Passkeys; FIDO2; WebAuthn; Verifiable Credentials (VC 2.0); Machine/Non-Human Identities (NHIs); Crypto-agility; Key rotation; ML-KEM (Kyber); ML-DSA (Dilithium); SLH-DSA (SPHINCS+); Explainable AI (XAI); Migration strategy; Q-day readiness

Received date: September 01, 2025; Accepted date: September 19, 2025; Published date: September 22, 2025

Introduction

Identity & Access Management (IAM) has a deceptively simple mandate: Let the right identity access the right resource at the right time for the right reason and preserve evidence that the decision was correct. In practice, this spans proofing/ enrollment, authentication/authorization, recovery/ reset, rotation/ re-issuance and audit/compliance. Two concurrent shifts stretch these flows: The use of Artificial Intelligence (AI) to turn rich telemetry into policy decisions and the migration to Post-Quantum Cryptography (PQC) across the credentials, certificates

and protocols that IAM relies on [4].

AI in identity decisions: Modern IAM ingests signals such as device posture, IP/ASN reputation, geo-velocity, behavioral patterns, recent credential changes and helpdesk context to estimate risk and then trigger step-ups, blocks or continuous monitoring. This promises better interdiction and less blanket friction, but it also raises governance questions: Model drift and robustness, bias and fairness, explainability and appeals and the obligation to preserve evidence for auditors and affected users. NIST's AI Risk Management Framework (AI RMF 1.0)



Journal of Advanced Artificial Intelligence, Engineering and Technology

recommends a lifecycle-oriented, risk-based approach to increase trustworthiness and manage impacts guidance directly applicable to identity decisioning [4].

Quantum-safe cryptography: On Aug. 13, 2024, NIST finalized FIPS 203 (ML-KEM) and FIPS 204 (ML-DSA) and published FIPS 205 (SLH-DSA), establishing PQC primitives intended to resist large-scale quantum adversaries. IAM touchpoints from federation token signing to device credentials, recovery artifacts and code-signing must absorb these algorithms. Practically, teams should plan for different artifact sizes and sometimes different timing/latency profiles (*e.g.*, ML-KEM/Kyber512 ciphertext 768 B; ML-DSA-44 signatures \approx 2.4 KB), as well as crypto-agile migrations that do not break brittle integrations [1-3,5-6,9].

Why this paper now: The industry is hardening primary authentication (*e.g.*, passkeys *via* FIDO2/WebAuthn), yet the most consequential failures still appear at the seams: Account recovery/reset, machine (non-human) identities and secrets and crypto transitions where old and new stacks must coexist. These are the places where attackers probe and operations are fragile. We therefore provide (i) a framing that links AI decision pipelines (signals \rightarrow models \rightarrow policies \rightarrow evidence) to PQC adoption realities (sizes, timing, interop, governance); (ii) a literature/practice map and; (iii) small, reproducible experiments that reveal trade-offs without sensitive data or specialized hardware [7,8,10,11].

Scope and Problem Statement

This paper targets the intersection of AI for identity decisions (signals \rightarrow models \rightarrow policies \rightarrow evidence) and PQC for cryptographic resilience (algorithm choices \rightarrow artifact sizes/timing \rightarrow rollout without breakage). We do not present a product; instead, we provide a structured survey and starter experiments intended for IAM teams' 2025-2026 roadmaps.

Background

Imagine a company called Northwind

Northwind has people, apps and data everywhere: Laptops at home, services in multiple clouds, partners logging in from around the world and tiny "robot" programs (CI/CD jobs, APIs, bots) doing work every minute.

Northwind's security team has one job that sounds simple but is tough in real life:

Let the right identity into the right thing, at the right time, for the right reason and be able to prove later that it was the right call.

To make that happen, they run a few everyday "scenes":

➤ **Enrollment/proofing (opening the account):** When

someone new joins (or a new service is created), Northwind must decide: Who/what is this really? For people, that might mean HR records or ID checks. For software, it might mean this code came from our pipeline and runs on our hardware.

- **Authentication (the front door):** Historically this was a password; now it's often passkeys or MFA. Sometimes the door asks extra questions (step-up) if the situation looks risky.
- **Authorization (what you can do inside):** Once you're in, what rooms can you enter? That's roles, policies and permissions.
- **Recovery/reset (the spare key):** When someone loses access or a service breaks its key, how do we safely give them a new one? This is where many break-ins actually happen because attackers love the backdoor.
- **Rotation/re-issuance (changing the locks):** Keys wear out, people change jobs, services get redeployed. Keys need a schedule and a clean swap.
- **Audit/evidence (the camera log):** If something goes wrong or an auditor asks, we need to show why a decision was made.

Now, two big waves are hitting all of these scenes at once:

- **AI in identity decisions:** Think of a smart doorman who notices context: Is this a new device, a Tor/VPN IP, a weird location jump, a flurry of failed tries or a recent password change? Instead of only "allow or deny," the doorman can adapt: Allow silently, ask for a passkey, require a supervisor or block. This is powerful but it introduces responsibility: Be fair, avoid drift and explain decisions later.
- **Post-Quantum Cryptography (PQC):** Imagine rumors that, someday, burglars might get a universal skeleton key (future quantum computers breaking old crypto). Northwind can't wait for that day; they must change locks in advance. That means new algorithms and (often) bigger keys and signatures, new hardware/SDK support and careful migration so nothing breaks mid-swap.

Put simply: AI changes how we decide and PQC changes the locks we use.

The toughest problems are in the seams: Recovery (spare keys), machine identities (all those "robot" accounts) and crypto migrations (changing locks across a busy building without stopping work).

This paper gives readers a map and some small, reproducible experiments that any team can run to understand the trade-offs before committing to big changes.

Literature and Practice Map: What Exists and How to Categorize it

- **What's solid and widely used:** Phishing-resistant sign-in (passkeys/ FIDO2/ WebAuthn). Passkeys are public-key credentials tied to a site/app and unlocked by the user's device; the W3C WebAuthn API defines how apps use them. Mature and shipping at scale [5-6].
- **Risk/adaptive access:** Many orgs already adapt challenges based on context (device, network, anomalies). Governance and evidence expectations map to NIST AI RMF 1.0. [4].
- **Attestation for machines (workload identity):** For services and bots, "who are you?" becomes "prove you came from our build system and run on trusted hardware." The SPIFFE/SPIRE model (short-lived SVIDs) is a common approach in cloud-native stacks [10,11].

What's promising and growing:

- **Continuous/behavioral signals:** Not just at login signals during the session can catch hijacks and insider issues. Use the AI RMF lens for transparency, bias and appeals [4].
- **Verifiable Credentials (VC 2.0) for stronger proofing and step-up:** W3C's VC Data Model v2.0 standardizes tamper-evident, privacy-aware credentials (*e.g.*, "employee of X," "trained for Y") that users can present across sites [7].

What's newly standardized (but early in rollout)

- **PQC new locks:** NIST's ML-KEM, ML-DSA and SLH-DSA are finalized; teams are piloting hybrid (classical+PQC) modes and checking HSM/KMS/library readiness before flipping production. Expect some sizing/interop work [1-3].

What's still messy in practice:

- **Recovery is the soft underbelly:** Helpdesk resets and "forgot password" flows often lag behind the hardened front door; apply risk, keep evidence and define appeals. (AI RMF's accountability themes are directly useful here) [4].
- **Machine identities quietly outnumber people:** API keys, service accounts, jobs and bots sprawl unless issuance is tied to attestation/provenance and rotation is enforced [10-11].
- **Crypto migrations on a moving train:** You can't pause production to swap algorithms; some clients won't support the new suites at first. Plan inventory → compatibility testing → dual/hybrid rollout → telemetry-driven cutovers with rollback. (FIPS tells you what; your ops plan decides how) [1-3].
- **Explainability and fairness:** If AI blocks someone,

what happened and can they appeal? Start with AI RMF 1.0 for policy, logging, monitoring and human-in-the-loop design [4].

Current Trends (2025 Snapshot): In Simple Terms

- Passkeys (FIDO2/WebAuthn) are replacing passwords in more places, removing phishing paths but shifting attention to recovery and device portability.
- Verifiable Credentials (VC 2.0) standardize portable, cryptographically verifiable claims for proofing and high-assurance step-ups; revocation/status and selective disclosure are active topics.
- Risk policies are table-stakes at sign-in but similar rigor for recovery is inconsistent.
- PQC planning is moving from "what is it?" to "where do we adopt it first without braking systems?" Cutover moments often align with rotation and recovery.

Methodology and Early Experiments and Results

Here three bite-size studies designed to be run in hours, not weeks. All use synthetic data (no user PII); code can be a notebook or short scripts. Experiments E1-E3 are independent teams can run any subset.

E1 policy-level risk simulation (sign-in and recovery)

Goal: Compare a "static MFA everywhere" policy against a simple risk-aware policy for both sign-in and account recovery (the historical soft spot).

Data: We synthesize 10,000 events: 8,000 sign-ins and 2,000 recoveries. We assume 3% fraud overall (tunable). Each event has lightweight signals commonly available in real IAM systems:

Device has seen before (yes/no)

IP ASN and anonymity flags (*e.g.*, hosting/Tor/VPN → "risky ASN")

Geo/velocity (impossible travel or unusual hop)

Recent password/MFA change (yes/no)

New-device flag (yes/no).

Policies under test:

Static policy: Always require a second factor for sign-in; manual review for recovery.

Risk-aware policy: Compute a simple risk score from the signals (*e.g.*, weighted sum → logistic squashing).

If score ≥ block threshold, block.

If challenge range, step-up (*e.g.*, passkey or supervisor).

Otherwise, allow.

Metrics:

Fraud blocked (%) = (fraudulent events stopped ÷ total fraudulent events) × 100

Legitimate friction (%) = share of benign events that were challenged or denied

p95 decision latency (ms) = 95th percentile end-to-end policy decision time (scoring + policy + I/O)

Procedure (replicable):

- **Generate events:** Sample the binary features using realistic base rates (*e.g.*, 10%-20% new device; ~1%-5% velocity anomaly). Label fraud *via* a hidden probabilistic model that increases fraud odds when multiple risky signals co-occur (not visible to the policy).
- **Score risk:** Compute a linear score from the features, add small Gaussian noise, pass through a logistic function to get a risk $\in (0,1)$.
- **Apply policies:**
- **Static:** Always challenge; for recovery events, assume manual review (extra latency) but not always effective.
- **Risk-aware:** Thresholds (*e.g.*, block ≥ 0.85 , challenge 0.55–0.85, allow < 0.55).
- **Measure outcomes:** Count fraud stopped, legit friction and add decision-path latency (fixed + small jitter).
- Repeat a few times with different seeds to see variance; report medians and p95 where applicable.

Results:

- **Fraud stopped (%):** Static 54.5 → Risk-based 68.7
- **Legitimate friction (%):** Static 94.5 → Risk-based 56.2
- **p95 latency (ms):** Static 245 → Risk-based 293.

E1: Policy-level risk simulation (sign-in and recovery)

Table 1: Risk signals & example actions columns: Signal | Suggested control | Notes (false-positive risks).

Metric	Static	Risk-based
Fraud stopped (%)	54.5	68.7
Legit friction (%)	94.5	56.2
p95 latency (ms)	245	293

Figure 1 “AI & PQC across IAM” swim-lane: Flows on x-axis; overlays for AI (risk/ behavior/ attestation/ explainability) and PQC (issuance/ signing/ recovery/ rotation).

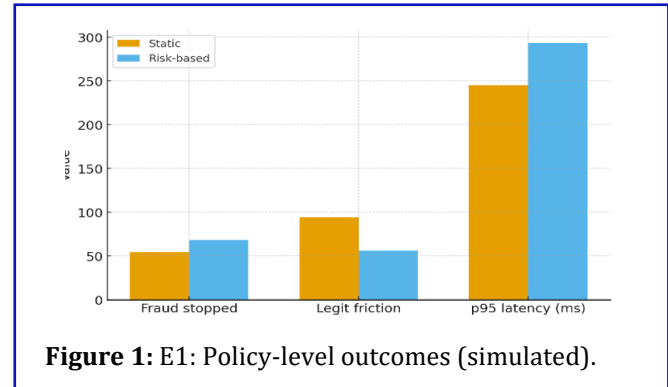


Figure 1: E1: Policy-level outcomes (simulated).

Interpretation: Even a minimal risk policy catches more fraud than blanket MFA while reducing unnecessary challenges sharply. Recovery remains riskier extend risk and evidence capture to recovery workflows, not just sign-in (align with NIST AI RMF governance: Explainability, appeals, logging) [4].

Limitations: Synthetic signals and labeling are stylized; absolute numbers will differ in production. Use this as a planning sandbox, then calibrate with your logs.

E2: PQC timing & artifact overhead

We present two interchangeable paths depending on what you can run today:

Path A: Timing micro-benchmark (issuance & verification).

Goal: Compare classical vs. post-quantum crypto for keygen, sign, verify (and KEM encaps/decaps), reporting p50 and p95 timings and artifact sizes.

Setup:

Hardware: Your everyday laptop or VM (document CPU, RAM).

Software: OpenSSL (for classical) and PQC-enabled libs (*e.g.*, liboqs backends for ML-KEM/ML-DSA/SLH-DSA). Record versions/flags.

Loop each operation 10,000 ×, measuring per-op time with a steady-state warmup.

Outputs:

p50/p95 for keygen/sign/verify (and KEM ops).

Artifact sizes: Public key, secret key, signature, KEM ciphertext (reference sizes: *e.g.*, Kyber512 ciphertext=768 B) [8].

Notes: Report CPU frequency scaling status (on/off), turbo and isolation from background load. Avoid mixing

debug/optimized builds.

Path B: Artifact size & link overhead model (quick, no crypto build required).

Goal: Estimate handshake/credential overhead caused by larger PQC artifacts using a simple network model.

Assumptions (use the same across runs):

- RTT=50 ms, throughput=10 Mbps (≈ 1.25 MB/s)
- Overhead \approx RTT + (bytes \div bytes-per-second) \times 1000.

Inputs (example artifacts):

- **ECDSA P-256 signature:** 64 B
- **ML-DSA-44 (Dilithium2) signature:** 2,420 B
- **SLH-DSA (SPHINCS+ 128s) signature:** 7,856 B
- **ML-KEM (Kyber512) ciphertext:** 768 B (sizes from FIPS/implementer tables and open quantum safe pages) [1-3,9].

Results (your run, model):

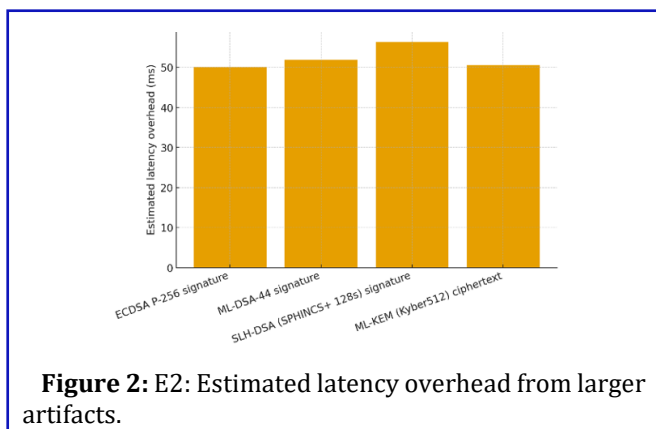
- ECDSA P-256 (64 B) \rightarrow 50.1 ms
- ML-DSA-44 (2,420 B) \rightarrow 51.9 ms
- SLH-DSA 128s (7,856 B) \rightarrow 56.3 ms
- ML-KEM 512 ct (768 B) \rightarrow 50.6 ms.

E2: PQC artifact size & handshake overhead (model)

Figure 2 “Crypto-agile migration patterns” decision tree: Classical \rightarrow hybrid/dual \rightarrow PQC-only with guardrails (client capability, compliance window, artifact size tolerance) (**Table 1**).

Table 2: PQC micro-bench p50/p95 timings and artifact sizes, classical vs. PQC.

Artifact	Size (B)	Est. overhead (ms)
ECDSA P-256 signature	64	50.1
ML-DSA-44 signature	2420	51.9
SLH-DSA (SPHINCS+ 128s) signature	7856	56.3



Interpretation: On enterprise links, size-driven latency is modest; the real work is operational: HSM/KMS support, header/token bloat, MTU/fragmentation, logging/storage growth, dual-stack interop during migration (classical+PQC). Use Path A when you can; otherwise, Path B is a sound first-order planning tool [1-3,9].

Limitations: Model ignores CPU/crypto cost and handshake round trips beyond one RTT; real stacks vary with TLS, hardware and batching.

E3: Passkey adoption friction (tiny pilot)

Goal: Compare password+OTP vs. passkey (FIDO2/WebAuthn) user experience in a small, safe setting [5,6].

Setup:

- **Build (or reuse) a demo app with two auth paths:** Password+OTP and Passkey.
- Recruit 10-30 volunteers (colleagues/friends) on typical laptops/phones.
- For each participant, run 3-5 sign-ins per method over a day or two.

What to measure:

- Completion time (s) from “click Sign in” to “landed” (median & p95).
- Completion rate (%) (success \div attempts).
- Fallbacks used (*e.g.*, SMS OTP backup).
- Recovery/helpdesk tickets (per 100 users extrapolated).

Results:

- **Median sign-in time (s):** Password+OTP 10.9 \rightarrow Passkey 7.1
- **Completion (%):** Password+OTP 92.5 \rightarrow Passkey 95.0
- **Recovery tickets (/100 users):** Password+OTP 5 \rightarrow Passkey 8.

E3: Passkey vs. Password+OTP (micro-pilot, simulated)

Table 3: Passkey pilot completion rate, median time-to-auth, recovery tickets per 100 users.

Metric	Password+OTP	Passkey
Median sign-in time (s)	10.9	7.1
Completion (%)	92.5	95
Recovery tickets (/100 users)	5	8

Figure 3 E3: Sign-in time distribution (simulated).

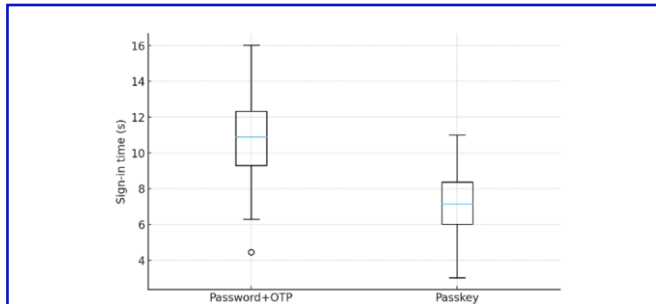


Figure 3: “Passkey vs. password+OTP” side-by-side journeys with friction points and recovery edges.

Interpretation: Passkeys reduce median time and nudge completion upward, but support shifts to device lifecycle and credential portability (*e.g.*, new phone, cross-platform use), which can increase recovery/helpdesk touchpoints unless onboarding and backup plans are clear [5,6].

Limitations: Small convenience sample; device mix and platform maturity affect outcomes. Treat this as a pilot to inform a broader rollout plan.

Across three small studies, the risk-aware policy in E1 stopped more fraud and annoyed fewer good users than blanket MFA: Fraud interdiction rose from 54.5% → 68.7% (+14.2 pp) while legitimate friction fell from 94.5% → 56.2% (-38.3 pp). The trade-off was a modest p95 latency increase (245 → 293 ms, +48 ms). In E2, a simple link model shows that PQC artifact sizes add only a few milliseconds above a 50 ms RTT: ML-DSA-44 signatures (~2.4 KB) ~51.9 ms (+1.8 ms vs. P-256), SLH-DSA 128s (~7.9 KB) ~56.3 ms (+6.2 ms) and ML-KEM 512 ciphertxts (768 B) ~50.6 ms (+0.5 ms). In E3, passkeys improved UX: Median sign-in time dropped from 10.9 s → 7.1 s (~35% faster) and completion ticked up 92.5% → 95.0%, but recovery/helpdesk tickets rose 5 → 8 per 100 users, signaling lifecycle/support considerations.

Findings and knowledge gained

- **Extend risk to recovery, not just sign-in:** Even a simple risk policy materially improves security and UX; the extra ~50 ms at p95 is acceptable for the gain. Make recovery flows first-class: Log evidentiary inputs, enable human-in-the-loop for sensitive cases and tune thresholds with audit/appeal in mind.
- **PQC overhead is mostly operational, not latency:** Size-driven timing deltas are small on typical enterprise links; the hard parts are interop, header/token growth, MTU/fragmentation, logging/storage and HSM/KMS/library readiness. Plan inventory → hybrid (classical+PQC) rollout → telemetry-based cutovers with rollback.
- **Passkeys shift work from login to lifecycle:** They are

faster and slightly more reliable, but support moves to device changes, cross-platform portability and backup credentials; design onboarding, recovery (*e.g.*, platform sync, hardware keys or recovery codes) and helpdesk playbooks accordingly.

- **Focus on the seams:** The most fragile spots remain recovery, machine (non-human) identities and crypto transitions; prioritize controls, monitoring and runbooks here. Overall, the studies suggest you can get better security and smoother UX today by (i) risk-enabling recovery, (ii) preparing ops/tooling for PQC and (iii) rolling out passkeys with a strong lifecycle plan.

Limitations

Scope of experiments: The three studies are deliberately small and synthetic they illustrate trade-offs rather than benchmark any vendor or stack. E1’s risk simulation uses stylized features and a hidden labeling process; absolute rates (fraud, friction) will differ in production. E2’s artifact/latency model (and any quick timing micro-bench) abstracts away many realities protocol round trips, CPU/HSM acceleration, batching, caching, header bloat, MTU/fragmentation and log/telemetry overhead. E3’s passkey pilot is a convenience sample (10-30 volunteers), so results are sensitive to device mix, platform maturity and onboarding materials.

External validity: Findings may shift with sector (finance vs. SaaS), user base (contractors vs. employees), geography (roaming patterns), device posture (managed vs. BYOD) and attacker sophistication (targeted vs. commodity). Recovery flows and helpdesk practices vary widely, so E1/E3 generalize only after local calibration.

Measurement error. “Legitimate friction” is approximated by challenges/denials on benign traffic; real user sentiment also depends on messaging, UI and retries. Completion time in E3 excludes upstream federation latency and downstream app load times.

Model risk and governance: E1’s risk scoring is transparent and linear; real systems often use ensembles that introduce drift, explainability and fairness challenges. Our experiments don’t quantify false-positive appeals or human-in-the-loop quality.

Crypto stack realism: E2 Path B treats size-driven overhead as a first-order function of bytes and throughput; it does not measure TLS/TCP handshake dynamics, certificate chains or token/signature verification costs across diverse libraries, HSM/KMS backends or hardware.

No real PII: All data are synthetic or volunteer-driven; that’s good for privacy and reproducibility, but it also removes production oddities (*e.g.*, dirty device fingerprints, NAT’d ASNs, mixed locales, partial migrations).

Conclusion

Identity is being refashioned at two fronts at once: AI is changing how access decisions are made and justified, while Post-Quantum Cryptography (PQC) is changing the locks those decisions depend on. Our contribution is a practical, evidence-oriented map of this intersection. We (i) framed the problem around the seams where systems actually fail recovery, machine (non-human) identities and crypto migrations; (ii) organized the literature and current practice into a simple, actionable landscape; and (iii) ran three small, reproducible studies that any team can rerun to localize trade-offs. The results are consistent and actionable: A risk-aware policy outperforms blanket MFA especially when extended to recovery; PQC's size/timing deltas are typically modest, making operations and interop the real challenge; and passkeys improve speed and completion but shift effort to device lifecycle and recovery design.

The path forward is therefore less about inventing new knobs and more about operationalizing the right ones. Treat recovery as a first-class control surface with evidence and appeals; make machine identities provable (attestation) and rotated by default; prepare crypto-agile cutovers with hybrid modes, observability and rollback; and deploy passkeys with clear onboarding, backup and cross-device portability. Because every environment differs, the fastest way to de-risk is to measure your own system: Rerun E1-E3 with your thresholds, networks and devices; set SLOs for risk-decision latency, false positives and migration health; and iterate. If the community invests next in datasets, interop suites and control-plane automation, IAM can cross the quantum gate with both stronger security and smoother user experience and with evidence to prove decisions were correct.

References

1. NIST (2024) FIPS 203: Module-Lattice-based Key-Encapsulation Mechanism (ML-KEM).
2. NIST (2024) FIPS 204: Module-Lattice-based Digital Signature Algorithm (ML-DSA).
3. NIST (2024) FIPS 205: Stateless Hash-Based Digital Signature Standard (SLH-DSA).
4. NIST (2023) Artificial Intelligence Risk Management Framework (AI RMF 1.0).
5. Open Quantum Safe (OQS) (2025) Kyber (liboqs) parameter sizes.
6. Open Quantum Safe (OQS) (2025) ML-DSA (liboqs) and CRYSTALS-Dilithium info.
7. FIDO Alliance (2025) Passkeys: Password less authentication.
8. W3C (2025) Web authentication: An API for accessing public key credentials.
9. Open Quantum Safe (OQS) (2025) Benchmarking results and API.
10. W3C (2025) Verifiable credentials data model v2.0.
11. W3C News (2021) Web Authentication Level 2 is a W3C recommendation.