

The Half-Life of Trust

Hardware-Rooted and Mathematics-Rooted
Foundations for Verifiable AI

Abdelhamid Bakhta

StarkWare

<https://github.com/AbdelStark>

April 2026

Abstract. Verifiable AI infrastructure is being constructed on foundations that age unevenly. The dominant substrate for production-scale verifiable inference—trusted execution environments on Intel, AMD, and NVIDIA silicon—roots its attestation in classical elliptic-curve signatures whose private keys are fused into chips at manufacture. Those roots cannot be rotated in software; migrating them requires a hardware refresh cycle measured in years. A sufficiently large fault-tolerant quantum computer running Shor’s algorithm would render the signing keys of every deployed generation recoverable, turning archival attestation reports into *retroactively forgeable* artifacts. For ephemeral verification this is a manageable migration problem; for evidence that must outlive the hardware that produced it—regulatory records, model provenance, adjudicatory evidence, scientific reproducibility—the foundation is structurally fragile. We contrast this with mathematics-rooted verifiable computation: STARK-family proof systems whose soundness rests on a public, hash-dominated mathematical assumption stack and whose trust ages gracefully under the best known quantum attacks. This is a vision paper. We do not claim that STARK-based verification is production-ready for frontier-scale inference; it is not. We do claim that the *aging behavior* of a trust substrate is a first-class engineering property, that current investment patterns implicitly privilege substrates whose trust decays, and that the asymmetry of consequences argues for a deliberate hybrid trajectory. We contribute a trust-aging taxonomy, a retroactive-forgery threat model, a capability-vs-durability comparison, and a three-layer path forward: hybrid architectures, STARK-friendly ML research, and cryptographic agility in AI evidence regimes.

1. Introduction

Cryptographic systems have a property that is rarely named and almost never engineered for: how their trust ages. Some forms of trust age well. They rest on mathematical assumptions that, when interrogated by successive generations of adversaries, degrade predictably and can be reinforced by parameter growth. Other forms are brittle. They rest on secrets, hardware lifetimes, and assumptions about what future computers cannot do.

For most of the history of applied cryptography, this distinction was an academic curiosity. Protocols were designed, deployed, replaced. The question of how signatures produced in 1998 would fare against 2028 adversaries was not load-bearing, because the artifacts those signatures guarded were themselves ephemeral. Today, two forces make aging a structural concern. First, credible quantum threats on the horizon, crystallized by the standardization of post-quantum signatures in NIST FIPS 204 and FIPS 205 (National Institute of Standards and Technology, 2024a,c), have shifted migration from a theoretical exercise to an operational one. Second, we are entering an era in which the outputs of AI systems—decisions, generations, recommendations—increasingly require independent verification, and that verification must sometimes outlive the system that produced it. Model weights trained today will be litigated for decades. Inference performed today will be evidence in tomorrow’s disputes. Provenance chains anchored today will be interrogated long after the hardware that witnessed them has been recycled.

When we build systems for verifiable AI inference—systems whose job is to convince us that a particular model, on particular inputs, produced a particular output—we are choosing a substrate of trust. That substrate has a half-life. We should choose it with the half-life in mind.

Thesis. We argue three things. First, that the *aging behavior* of a verifiable-AI substrate is a first-class engineering property, distinct from capability, cost, and latency, and that current discourse has not treated it as such. Second, that the dominant substrate for verifiable AI in 2026—hardware-rooted trusted execution environments—is, under its own cryptographic assumptions, structurally fragile for any use case whose verification horizon exceeds the deployed silicon generation. Third, that an alternative substrate based on hash-based cryptography and STARK-family proof systems ages differently, in a manner that aligns with the temporal demands of AI evidence regimes being written now.

We are not arguing that trusted execution environments should be abandoned. They work today. They are the pragmatic production answer for ephemeral verifiable inference at scale, and anyone selling a different solution for that use case is either ahead of the curve or telling a story. We are arguing that the framing *TEEs versus STARKs* is the wrong frame. The two substrates are complementary rather than competitive: they guarantee different properties, age under different assumptions, and fail in different ways, which is precisely why a serious architecture composes them rather than chooses between them. The right frame is *which trust ages with the evidence we want to carry forward*, and under that frame a mature verifiable-AI infrastructure uses both substrates deliberately, with explicit reasoning about the temporal scope of each guarantee.

Scope. By *verifiable AI* we mean the class of infrastructures that produce cryptographic evidence—attestations or proofs—that a specific model, on specific inputs, produced a specific output, and that such evidence can be independently checked by a party that did not witness the computation. This is a vision and position paper. It does not present new cryptographic constructions, new proof systems, or new empirical benchmarks. It synthesizes existing primitives, articulates a property (trust aging) that cuts across them, and argues for an investment and governance posture that follows from taking that property

seriously. We focus on *verifiable inference*—proofs or attestations that a specific model produced a specific output on specific inputs—because it is the verifiable-AI surface being built out most aggressively in 2026. The same aging-behavior arguments apply to verifiable training and verifiable evaluation once those surfaces acquire production substrates; we restrict the present treatment to inference to keep the threat model precise.

Position vis-à-vis adjacent proposals. Three adjacent positions in the literature deserve explicit contrast. First, industry-facing confidential-inference proposals (Anthropic and Irregular, 2025) argue for TEE-based architectures with PQ-signed attestation as the migration path; we agree on the near-term direction but argue that the substrate’s trust-aging properties make the migration insufficient for evidence whose verification horizon exceeds the deployed silicon generation. Second, general PQ crypto-agility guidance (Barker et al., 2021; National Security Agency, 2022) describes how to retire vulnerable algorithms in protocols whose roots are rotatable. The hardware-security-module and trusted-platform-module communities have separately discussed non-rotatable-root migration and long-term evidence archival for more than a decade: the IETF Evidence Record Syntax (Gondrom et al., 2007) specifies how to preserve the evidentiary value of signed data across algorithm failures by chaining archive timestamps over successively stronger hash functions, and NIST Special Publication 800-208 (Cooper et al., 2020) standardized the stateful hash-based signature schemes LMS and XMSS for firmware, boot-chain, and other contexts where keys outlive the signature scheme that produced them. These precedents, together with the long-term archive working-group tradition (LTANS) from which they emerged, treat the non-rotatable-root question as inherited engineering, not a new observation. Our contribution is not to claim that non-rotatable-root reasoning is new, but to identify TEE attestation for verifiable AI as the specific deployment class where the standard migration story does not apply, because the root is physical, the evidentiary horizons are measured against AI-governance regimes that did not exist when those earlier conversations took place, and neither ERS-style countersignature chains nor firmware-signing migrations can retroactively repair attestations produced under a compromised in-silicon key. Third, the zkML benchmarks literature (Liu et al., 2021; Sun et al., 2024) focuses on prover performance at fixed soundness; we frame an orthogonal question of *which soundness substrate ages with the evidence* and argue that performance work must be read against that frame. We scope the aging argument in this paper to hash-rooted proof systems (the STARK family); zkML deployments built on pairing-based SNARKs rely on assumptions that are themselves Shor-vulnerable and therefore inherit an aging profile closer to the hardware-rooted case than to the STARK case.

Relation to complementary work. The argument here is a layer inside the broader *high-assurance AI* agenda we have developed elsewhere (Bakhta, 2026), which treats verifiable computation as one component of a composed assurance stack alongside formal specifications, attestation, privacy-preserving computation, and safety cases. The present paper zooms in on a single load-bearing question within that stack: what cryptographic substrate should carry the evidence forward, and how does the aging of that substrate shape what regulators, courts, auditors, and the public can still verify in ten, twenty, or forty years? We see the two papers as complementary: the assurance-stack argument establishes *that we*

need verifiable artifacts; the half-life argument establishes *which kind*.

Structure. Section 2 characterizes the current hardware-attested substrate for verifiable AI and explains why it is the pragmatic baseline. Section 3 traces the classical cryptographic dependencies embedded in every major TEE attestation protocol. Section 4 explains the architectural constraint—non-rotatable fuse-based roots—that turns those dependencies into a temporal liability. Section 5 formalizes the retroactive forgery threat model—the signature-theoretic analog of *harvest now, decrypt later*, which we name *forge later*. Section 6 develops the mathematics-rooted alternative: hash-based cryptography, STARKs, and why their trust ages differently. Section 7 gives the honest side-by-side: capability today, durability tomorrow. Section 8 addresses two arguments that hold independently of the quantum question: trust topology and side-channel exposure. Section 9 presents a three-layer path forward: hybrid architectures, accelerated investment in STARK-friendly ML, and cryptographic agility in AI evidence regimes. Section 10 argues why this matters beyond cryptography. Section 11 concludes.

2. The Current Substrate: Hardware-Attested AI

The dominant substrate for verifiable AI computation in publicly visible production deployments as of 2026 is the trusted execution environment (TEE). Intel SGX (Costan and Devadas, 2016), Intel TDX (Intel Corporation, 2023), AMD SEV-SNP (Advanced Micro Devices, 2020), and NVIDIA’s Confidential Computing architecture on H100 and successor GPUs (NVIDIA Corporation, 2023) share a common design. Each provides a hardware root of trust, a remote-attestation protocol, and a cryptographic chain that lets a remote verifier gain confidence that a specific piece of code executed on specific hardware in a specific configuration, and that its outputs were produced by that execution rather than by a bystander process, the host kernel, or the cloud operator.

This is genuine engineering progress, and dismissing it would be foolish. TEEs make it possible to run inference on untrusted infrastructure with credible assurance about what executed. They have enabled entire categories of confidential AI workloads: model evaluation behind enterprise firewalls, multi-tenant inference with cryptographic isolation, cross-organization collaboration on sensitive data, and—at the frontier—deployments where a model provider wishes to serve predictions without exposing weights and a user wishes to receive predictions without exposing inputs.

For verifiable inference at production scale today, TEEs are the right answer. The performance overhead on modern confidential-computing GPUs for transformer inference is workload- and configuration-dependent; vendor-reported figures vary by batch size and latency profile, with single-digit overheads at favorable batch sizes and larger overheads under latency-sensitive configurations (NVIDIA Corporation, 2023). We present these figures as illustrative of the regime rather than as an established consensus across the broader ecosystem. Even at the upper end, this is a regime where previously the choice was between no verification and no deployment. The developer experience has matured: SDKs, attestation verification libraries, and cloud-operator integrations now cover the common workflows. The deployment story is, for the first time, end-to-end plausible for a serving stack that a

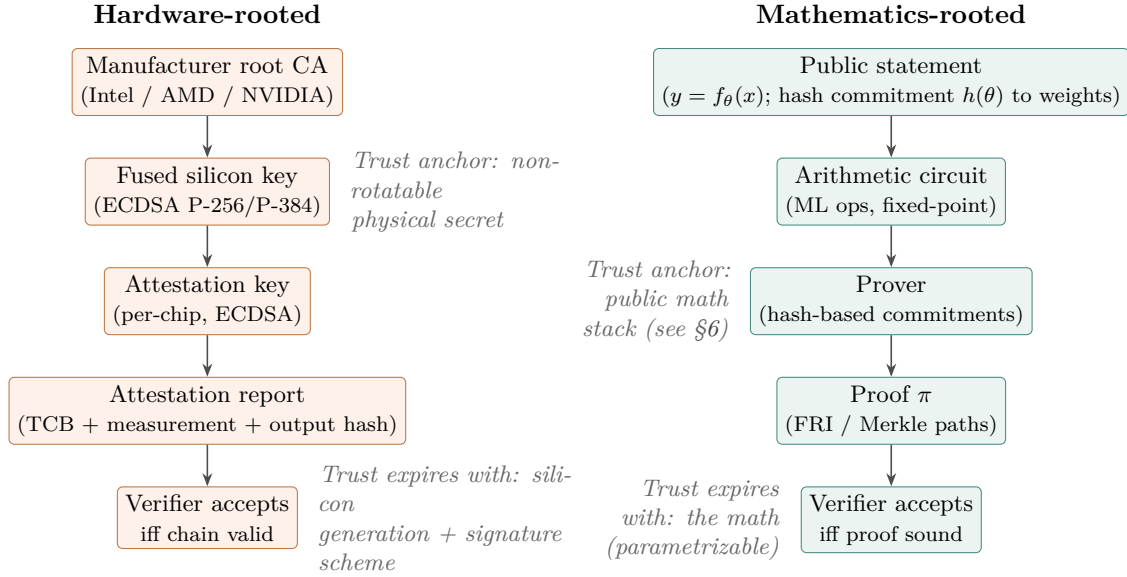


Figure 1: Two substrates for verifiable AI. The hardware-rooted chain delegates trust from a manufacturer’s certificate authority through a fused silicon key down to a per-chip attestation key that signs reports about what ran. Every link is a classical ECDSA signature. The mathematics-rooted chain produces a non-interactive proof whose soundness rests on a public, mathematics-only assumption stack—hash commitments, Fiat–Shamir in the random-oracle model, IOP/low-degree-test soundness, and concrete parameters, developed in §6—rather than on a single hardware secret. Aging behavior differs accordingly: Section 4 develops the hardware case, Section 6 the mathematics case.

large production deployment would accept.

This is the honest pragmatic baseline. We state it explicitly because the argument that follows is sometimes misread as advocacy for abandoning TEEs. It is not. The TEE ecosystem is the verifiable-inference infrastructure that exists, it does real work, and the engineers who built it have earned the production share they hold.

The question we raise is not whether TEEs work today. They do. The question is what assumptions we are encoding into the foundations of an AI infrastructure we will need to trust for years, possibly decades, after the hardware that underwrote the original attestations has been decommissioned.

3. The Cryptographic Dependencies Hidden in Attestation

Every major TEE attestation protocol deployed in production AI infrastructure today depends on classical public-key cryptography whose hardness reduces to the discrete logarithm problem over elliptic curves.

Intel SGX’s data-center attestation (DCAP) uses ECDSA over the NIST P-256 curve (Costan and Devadas, 2016; Intel Corporation, 2020). AMD SEV-SNP attestation reports are signed with the Versioned Chip Endorsement Key (VCEK), an ECDSA P-384 key, and the certificate chain terminates at the AMD Root Key (ARK) also instantiated as an elliptic-

curve signing key (Advanced Micro Devices, 2020). NVIDIA H100 confidential computing attestation follows the same architectural pattern: a certificate chain signed with elliptic-curve primitives, rooted at a manufacturer’s certificate authority, and consumed by verifier libraries that check the chain and the measurement of what executed in the confidential computing region (NVIDIA Corporation, 2023). Intel TDX composes a hardware-signed quote over the TDX measurement and delivers it to a remote verifier through a similar ECDSA-rooted chain (Intel Corporation, 2023).

Every one of these schemes is, under current cryptanalytic understanding, broken in the following technical sense by a sufficiently large fault-tolerant quantum computer running Shor’s algorithm (Shor, 1997): the private key corresponding to any issued public key can be recovered in polynomial time in the curve size. Once the key is recovered, an attacker can produce signatures that verify as legitimate under that public key. The attacker does not need access to the original signer. They do not need access to the original hardware. They need only the historically disclosed public key, which by design of any public-key system is public.

The standard response to this observation is that the field will migrate. NIST has finalized ML-DSA (FIPS 204) (National Institute of Standards and Technology, 2024a) and SLH-DSA (FIPS 205) (National Institute of Standards and Technology, 2024c) precisely so that protocol designers can rotate signature schemes before capable quantum machines exist. TLS, code signing, software update infrastructure, and identity systems have clear migration paths. For each of these, the migration is a protocol upgrade: clients and servers negotiate a new suite, certificates are reissued, old credentials are retired. The existing installed base does not need to be physically replaced.

The quantum resource frontier has tightened sharply. Two developments over 2024–2025 have moved the cryptanalytically-relevant-quantum-computer question from “whether the architecture scales” to “how many logical qubits the scaled architecture needs.” Google Quantum AI’s *Willow* processor demonstrated below-threshold scaling of the surface code: as the code distance grew from 3 to 5 to 7, the logical error rate dropped by roughly a factor of two per step, the exponential suppression the surface code predicts but had not previously exhibited at cryptographically interesting scales (Google Quantum AI, 2025). This is the threshold the fault-tolerance community had been trying to cross since Shor’s algorithm was published; it does not deliver a cryptographically relevant machine, but it closes a standing doubt about whether the surface-code approach can in fact scale. In parallel, the concrete resource estimate for factoring RSA-2048 has collapsed by roughly an order of magnitude in a single paper. The 2021 Gidney–Ekerå estimate (Gidney and Ekerå, 2021) put the cost at 20 million noisy qubits for an eight-hour run and was, for several years, the reference number. In 2025, Gidney showed that RSA-2048 could be factored with *fewer than one million* noisy qubits in under a week, via tighter magic-state distillation and improved modular-arithmetic layouts (Gidney, 2025). The same techniques tighten estimates for the elliptic-curve discrete-logarithm problem that underwrites every TEE attestation chain discussed above. None of this is a demonstration that a cryptanalytically relevant quantum computer exists; what it is, is a reminder that the resource frontier is a moving target, and the movement over the past four years has been in the direction of shorter

adversary timelines.

Industry migration timelines bound the planning window. The public roadmaps for post-quantum migration have converged on *2030–2035* as the window in which classical-ECC signatures should be deprecated and disallowed. The NSA Commercial National Security Algorithm Suite 2.0 (National Security Agency, 2022) requires exclusive use of post-quantum algorithms in new national-security software and firmware signing from 2025, in new networking and custom equipment by 2030–2031, with full enforcement by 2035. The U.S. White House National Security Memorandum 10 (The White House, 2022) directed federal agencies to inventory classical-cryptographic systems and prepare migration plans on the same schedule. The UK National Cyber Security Centre’s 2024 guidance (UK National Cyber Security Centre, 2024) frames 2024–2028 as a discovery and inventory phase, 2028–2031 as the high-priority migration phase, and 2031–2035 as the full-migration phase. NIST’s initial public draft IR 8547 (Moody et al., 2024) proposes to deprecate ECDSA and RSA for signatures in 2030 and disallow them after 2035. These schedules do not describe when capable quantum computers will arrive; they describe the latest dates by which classical-cryptographic artifacts produced today can be expected to retain their security properties. A 2026-era TEE attestation whose admissibility horizon runs to 2045 is, against these schedules, already outside the protected window.

Mosca’s inequality and long-horizon AI evidence. Mosca (Mosca, 2018) formalized the migration-timing question with a short inequality. Let X be the time required to migrate a system to post-quantum cryptography, Y the security lifetime demanded of the data or evidence the system produces, and Z the time until a cryptographically relevant quantum adversary exists. The system is at risk whenever $X + Y > Z$. For TLS sessions or short-lived access tokens, Y is seconds to minutes and the inequality is dominated by Z ; the migration question reduces to “can we rotate primitives before Z ?” For a TEE attestation that certifies a regulated AI decision whose admissibility horizon is twenty years, Y is twenty years, and X is bounded below by a hardware refresh cycle. Even on conservative priors about Z , $X + Y$ can exceed Z for artifacts produced today, which is the temporal condition under which long-horizon attestations are already at risk before any capable quantum computer has been built (Section 5 names this condition formally). The public roadmaps above are coherent with Mosca’s framing only under the implicit assumption that Y for the data they govern is short. For the AI evidence regimes this paper addresses, that assumption does not hold.

The protocol-upgrade path available to TLS, code signing, and identity systems is the escape hatch classical-ECC deployments in those domains can reach for. TEE attestation cannot reach for it. The reason is not a failure of foresight on the part of TEE designers; it is architectural, and we develop it in the next section.

4. The Silicon Problem: Non-Rotatable Roots

The trust anchors for hardware attestation are not configuration values living in software. They are derived from physical secrets written into the silicon at manufacturing time.

Intel’s Provisioning Certification Key, the cryptographic identity at the base of the SGX attestation chain, is derived from values written into the chip during fabrication (Costan and Devadas, 2016; Intel Corporation, 2020). AMD’s chip endorsement keys are anchored in fuses set at wafer test (Advanced Micro Devices, 2020). NVIDIA’s confidential computing root of trust follows the same pattern (NVIDIA Corporation, 2023). In each case, the root value is inaccessible to software, inaccessible to the firmware update path, and (by intent) inaccessible to the manufacturer themselves after provisioning.

This was a deliberate and defensible design choice. Rooting trust in physical secrets, sealed during fabrication and never exposed to software, is precisely what makes hardware attestation *hardware*. If the root of trust could be rotated by a firmware update, an attacker who compromised the firmware could rotate it. Immutability of the root is the property that defends against the compromise scenarios the TEE architecture was built to withstand. The price of that immutability is the property we are now surfacing: the root cryptographic scheme, the curve, the signature algorithm, the hash—are all locked in at tape-out.

To migrate the cryptographic root of trust, one therefore does not ship a patch. One designs new silicon, ships it through a fabrication cycle, deploys it into data centers, and decommissions the old generation. Data-center CPU and GPU refresh cycles, measured from architectural freeze to broad installed-base deployment, run on multi-year timescales, and full retirement of an older generation takes longer still. Fab capacity, supply-chain constraints, customer adoption velocities, and the lifetime of serviced contracts all extend the tail.

Two consequences follow. First, any post-quantum migration of TEE attestation must be timed so that new silicon is in place *before* capable quantum machines exist, not after, because retrofit is impossible. Second, and more importantly for our argument, the attestation reports produced by current-generation hardware do not automatically benefit from any future migration. A report signed in 2026 by a P-384 VCEK is still a P-384-signed report in 2040, regardless of what silicon is shipping then. Whether that report still means what it claimed depends on whether the cryptographic assumption under which it was produced still holds.

What we ask in this paper is precisely what happens to those archival reports, and to the ones current hardware will keep producing until the next generation takes over, the day after a sufficiently large quantum computer exists.

5. Harvest Now, Forge Later

Most public discussion of the quantum threat to cryptography has focused on confidentiality. The phrase *harvest now, decrypt later* captures the immediate concern: encrypted traffic captured today can be archived and decrypted in the future once quantum capability arrives. This has driven the post-quantum KEM standardization effort (FIPS 203 / ML-KEM) (National Institute of Standards and Technology, 2024b) and the broad migration of transport security.

For *attestation*, the structurally analogous threat is more subtle and, we argue, more consequential for the evidence regimes being built around AI. Call it **forge later**.

Threat Model 1 (Retroactive attestation forgery). Let S denote an attestation signing scheme instantiated with a classical public-key algorithm vulnerable to quantum cryptanalysis (e.g. ECDSA over a standardized curve). Let \mathbf{pk} be a public key whose corresponding \mathbf{sk} is sealed in hardware at time t_0 and used to sign reports until $t_1 > t_0$. Consider an adversary \mathcal{A} with the following capabilities:

1. \mathcal{A} has access to any signature ever produced under \mathbf{pk} —or even only to \mathbf{pk} itself—collected before or after t_1 .
2. At some time $T > t_1$, \mathcal{A} gains access to a fault-tolerant quantum computer of sufficient scale to run Shor’s algorithm against S .
3. A verifier evaluates a claim about events in the interval $[t_0, t_1]$ at a time $T' \geq T$ using only the public key \mathbf{pk} and its embedding in a manufacturer certificate chain.

At time T , \mathcal{A} recovers \mathbf{sk} from \mathbf{pk} and can produce attestation reports, indistinguishable from legitimate ones under \mathbf{pk} , that claim arbitrary measurements and outputs for arbitrary timestamps in $[t_0, t_1]$. No access to the original hardware, to the original enclave, or to any private state of the original signer is required.

The consequence for a verifier at time $T' \geq T$ is qualitative: the verifier can no longer distinguish legitimate attestation reports from forgeries that claim to be from the same hardware. Every report produced under \mathbf{pk} is, from the verifier’s perspective, indistinguishable from a counterfactual report \mathcal{A} could have produced after T .

This is a stronger statement than the usual framing. It is not that the signatures stop verifying; they remain mathematically valid under the original public key. What they cease to be is *unforgeable*, which is the property the verifier actually cared about.

Key-hierarchy refinement. Threat Model 1 is stated for a single public key \mathbf{pk} ; real TEE deployments use multi-level certification chains with different forgery economics at each level. In AMD SEV-SNP, for example, attestation reports are signed by a per-chip, per-TCB Versioned Chip Endorsement Key (VCEK) whose certificate is issued under the AMD Signing Key (ASK), which in turn chains to the AMD Root Key (ARK) (Advanced Micro Devices, 2020). Key generation and certification are distinct here: the VCEK is not derived from the ASK, but its certificate is, and verifiers follow the ARK–ASK–VCEK chain to accept a report. An adversary can compromise evidence in two regimes: *global*, by recovering a manufacturer root (one quantum computation against the ARK or ASK, every chip’s historical attestations forgeable through a fabricated certificate); or *per-chip*, by recovering individual VCEKs (one quantum computation per chip, per-chip forgery only). Intel SGX DCAP and NVIDIA H100 CC have analogous structures. The economics differ; the conclusion does not. Under either regime, a sufficiently motivated adversary eventually reaches the target evidence, because the public keys required to run the attack are public by design. The global regime is catastrophic and fast; the per-chip regime is expensive and slow. Neither is blocked by any protocol move available after the hardware has shipped.

When the threat binds. For many use cases, this is a manageable problem. Real-time fraud detection, live API serving, short-window trading, on-the-fly model-vs-user negotiation: the attestation must be valid at the moment of verification, and that moment is now. By the time quantum capability arrives, the verification is long finished. The report has done its job and been retired.

For other use cases, the temporal structure is precisely the reverse. Regulatory compliance records, court-admissible evidence of what an AI system decided, provenance chains for generative media, scientific reproducibility artifacts, audit trails for credit or medical decisions, cross-border accountability records—these are categories where the verification horizon is measured in years or decades, and where the evidentiary question posed later is often the key question. If in 2035 a regulator, a judge, or a counterparty needs to verify that a particular model produced a particular decision in 2026, and the only verification artifact is a TEE attestation signed under ECDSA-P-384, the answer available at that later date depends on whether the quantum-adversary scenario has materialized. If it has, the report is unfalsifiable in exactly the wrong direction: anyone with the recovered key could have produced it.

What makes this architectural rather than implementational. This is not a bug in any vendor’s attestation implementation. It is a structural property of any attestation system that satisfies two conditions simultaneously: (i) it ties evidence to a non-rotatable cryptographic root, and (ii) it instantiates that root with a signature scheme in a complexity class that contains efficient quantum algorithms for key recovery. Any system with both properties has an evidence- expiration date bounded by the slowest-aging assumption in its chain.

Why post-quantum silicon does not retroactively help. A future generation of TEE hardware using ML-DSA as its attestation signature algorithm eliminates the forge-later threat for reports produced by *that* hardware. It does not improve reports produced earlier by the classical-ECC generation, because those earlier reports remain verified against the earlier public keys. The retroactive forgery surface is, by construction, a property of the historical keys, not a property of the current verifier. It cannot be closed from the future.

Countermeasures that help partially, and why they are not enough. Three partial countermeasures deserve mention.

Timestamping and long-term signatures. Evidence can be countersigned by a long-term timestamping service that itself migrates to post-quantum signatures on a schedule faster than the quantum threat. If the countersignature binds the classical attestation to a timestamp *before* quantum capability emerges, a forged attestation produced *after* that capability cannot acquire a matching countersignature. This is the standard long-term validation pattern from archival document signing (European Telecommunications Standards Institute, 2016). It helps. It requires that the timestamping authorities were themselves trustworthy, that their keys have not been compromised, and that the countersignature was actually obtained contemporaneously. For attestation data not originally designed to be countersigned, these conditions often do not hold.

Anchoring digests to a broadly witnessed public log. If every attestation report is hashed and its digest committed to an append-only log published widely enough that the log itself is beyond forgery, then the *existence* of the original report at the committed time is evidenced, even if the attestation signature later loses its unforgeability. Concrete instantiations range from Certificate-Transparency-style logs (Laurie et al., 2013) to cryptographic timestamping services such as OpenTimestamps (Todd, 2016), which aggregate digests into a Merkle tree and anchor its root in the Bitcoin blockchain (Nakamoto, 2008). The durability of such an anchor rests on hash-based ordering and the long-term availability and consensus security of the underlying log or chain, rather than on a classical signing key: Bitcoin’s spending signatures are themselves ECDSA over secp256k1 and therefore Shor-vulnerable, but the block-header chain over SHA-256 is cemented by cumulative proof-of-work, and the ordering of anchored digests ages under Grover rather than Shor, provided the log remains publicly accessible and its consensus remains intact. A report whose digest is anchored in a block at height H_0 carries a verifiable claim that the report existed by the time H_0 was mined, and that claim survives the later compromise of the attestation signing key, because a retroactively forged report cannot retroactively appear in a block that was already mined. This reduces the problem from *could the report be forged* to *was the report anchored at the right time*. It helps. Its residual limit is subtle and worth stating precisely: the anchor proves the report existed at time t , but cannot exclude that the signing key was already compromised at t , because key recovery leaves no forensic trace in the signed artifact itself. It also does not help if the attestation in question was never anchored, and the anchor is evidentiary only for attestations produced before quantum capability emerges.

Switching to hash-based signatures at the root. One can, in principle, design hardware whose root of trust uses a post-quantum signature scheme such as the hash-based SLH-DSA or a lattice-based ML-DSA. This is the correct long-term solution for *new* silicon, and post-quantum migration is broadly on the agenda of the confidential-computing ecosystem; which specific family—lattice-based or hash-based—any given vendor adopts at the attestation root is an open engineering choice that we do not try to predict here. For *existing* silicon, the root cannot be retrofitted.

The combination of these partial countermeasures is substantial. Anyone deploying production TEE-attested AI systems today with long-horizon evidentiary requirements should be implementing all three; together, they constitute an operationally available hybrid defense that narrows—but does not close—the retroactive-forgery window for already-deployed silicon. What they do not change is the underlying structure: a verification chain whose deepest assumption is classical-ECC unforgeability is a chain with an expiration date the protocol cannot move. Closing that window, rather than narrowing it, requires either new silicon with a post-quantum root or a mathematics-rooted substrate that does not depend on a non-rotatable physical root at all.

6. Mathematics-Rooted Verification

A second family of cryptographic primitives ages differently. It is built almost entirely on collision-resistant hash functions and information-theoretic arguments, and it sits at the foundation of STARK-family proof systems (Ben-Sasson et al., 2018).

6.1 Hash-Based Cryptography Is Old and Well-Understood

Hash-based cryptography is not new. Lamport described one-time signatures in 1979 (Lamport, 1979). Merkle turned them into practical many-time schemes via tree constructions soon after (Merkle, 1988). For more than four decades, the assumption at the base of hash-based cryptography—that finding collisions in a well-designed hash function is computationally hard—has been interrogated by adversaries and conceded nothing structural. Hash functions have been broken (MD5, SHA-1) (Stevens et al., 2017; Wang and Yu, 2005), but the *paradigm* has not. Each fallen hash was replaced by a successor with larger output and a more conservative design; the structural assumption survived.

This is one of the oldest live cryptographic primitives we have. It is older than practical elliptic-curve cryptography. It is older than RSA at internet scale. That longevity is evidence, not proof, but it is the kind of evidence cryptographic hardness assumptions are evaluated on.

6.2 STARKs: Trust in a Math Stack, Not a Manufacturer

A STARK (Scalable Transparent ARgument of Knowledge) (Ben-Sasson et al., 2018) is a non-interactive proof system whose soundness rests on a public, mathematics-only assumption stack: an interactive oracle proof (IOP) with a low-degree-testing soundness argument, a Fiat–Shamir transformation analyzed in the random-oracle model, and a commitment scheme whose binding reduces to collision resistance of the instantiating hash function. Concrete parameter choices (field, query count, proof-of-work grinding, hash output length) fix the soundness error in a specific deployment. A STARK proof that a computation $y = f(x)$ was executed correctly gives a verifier a short certificate that can be checked in time polylogarithmic in the size of the computation, without trusting the prover, without trusting any setup, and without trusting any hardware.

For verifiable AI, the distinction from the hardware-attested chain is semantic as much as architectural. A TEE attestation says *trust this hardware*. A STARK proof says *verify this math*. The verifier of a TEE attestation must confirm a chain terminating at a manufacturer’s certificate authority. The verifier of a STARK proof must confirm a computation terminating at a hash digest.

6.3 Quantum Effects on Hash-Based Primitives

Quantum computers do not leave hash functions untouched, but they do not break them either. Grover’s algorithm (Grover, 1996) finds a pre-image of an n -bit hash in $O(2^{n/2})$ quantum queries, a quadratic speedup over classical brute force. The Brassard–Høyer–Tapp algorithm (Brassard et al., 1998) finds collisions in $O(2^{n/3})$ quantum queries, reducing the effective collision-resistance exponent from $n/2$ to $n/3$. Both are polynomial degradations of the classical security level, not exponential breaks of the underlying assumption. For a proof system targeting 128 bits of post-quantum collision security, one enlarges the hash output—for example from 256 to 384 bits, which under BHT preserves the 128-bit post-quantum target—and parameterizes the proof system accordingly; the performance cost is moderate, the mitigation is well-understood, and there is no Shor-equivalent hanging over the core assumption.

This is why STARKs are described in the literature as *plausibly post-quantum* (Ben-Sasson et al., 2018). The qualifier “plausibly” matters—cryptography is humble by tradition, and no one will claim certainty about tools adversaries will develop decades from now. But the structural argument is sound. A STARK proof produced today with a collision-resistant hash function of adequate output length remains verifiable, and remains sound under the best understood quantum attacks, without a built-in cryptographic expiration date tied to hardware.

6.4 What the Trust Anchor Actually Is

The trust anchor of a STARK-based verification chain is not a single assumption but a public, mathematics-only stack: conservative hash assumptions (collision and pre-image resistance of the instantiating hash), Fiat–Shamir soundness in the random-oracle model, the soundness of the underlying IOP and low-degree test, and the concrete parameter choices that fix the soundness error in a specific deployment. Implementation correctness of the verifier is an additional layer. The dominant cryptographic term over long horizons is the hash assumption, which is why we treat the stack as hash-dominated when reasoning about aging; the other terms are not zero, and an honest comparison names them. If any of these assumptions degrades—most plausibly through hash cryptanalysis or a Fiat–Shamir-specific attack—the remedy is to re-prove (where artifacts and source computations are still available), re-parameterize, or accept a downgraded soundness level (where they are not). The remedy does not require hardware migration, manufacturer cooperation, or coordination across a supply chain. It is a mathematical remedy applied to a mathematical artifact.

We state the positive analog of Threat Model 1 for structural symmetry.

Observation 1 (Graceful aging of hash-rooted proof systems). Let a STARK proof π of a statement φ be produced under the assumption stack $(H, \text{FS}, \pi_{\text{IOP}}, \text{params})$, consisting of a collision-resistant hash H , Fiat–Shamir soundness under random-oracle modeling of FS, soundness of the underlying IOP / low-degree test π_{IOP} , and the concrete parameter choices params . For any classical or quantum polynomial-time adversary \mathcal{A} , the probability that \mathcal{A} causes a verifier to accept a proof of a false statement is bounded by a soundness error ε that is a function of the four terms above. Degradation of any single term—most plausibly H under classical cryptanalysis or Grover/BHT—is a polynomial shift in ε that can be recovered by re-parameterization of (H, params) ; no term in the stack admits a Shor-style exponential break under current cryptanalytic understanding. Recovery does not require hardware migration, manufacturer cooperation, or coordination across a supply chain.

Observation 1 is not a theorem. Each of its four terms is the subject of active cryptanalytic work, and the soundness bound it asserts is the product of bounds each term independently provides. We state it to make the comparison with Threat Model 1 structurally symmetric: one named adversary, one stated assumption set, one quantified outcome, on each side.

This is a structurally different aging profile, and the difference is the central contribution we ask the reader to internalize.

6.5 The Honest Limits of the Mathematics-Rooted Substrate

The case for mathematics-rooted verification is not that it is free of assumptions. It is not. Five honest limitations deserve explicit statement.

Specification gap. A proof certifies that the proven computation was executed correctly. It does not certify that the proven computation was the right computation to prove. The specification gap between “what the prover claims to be computing” and “what the verifier cares about” is orthogonal to the soundness argument and must be addressed by complementary mechanisms. We have argued elsewhere (Bakhta, 2026) that this is where formal specifications and safety cases contribute.

Prover performance. Generating a STARK proof for a forward pass over a frontier-scale transformer is, at 2026 state-of-the-art, expensive in prover time and memory. Published work over the past several years has reported substantial reductions in prover cost through better commitment schemes, smaller fields, and improved arithmetizations (Haböck et al., 2024; RISC Zero, Inc., 2023; Succinct Labs, 2024), and the trajectory is encouraging; the absolute cost is not yet at per-call parity for Llama- or GPT-class workloads without specialized infrastructure. We address this in Section 9.

Hash choice. Hash-based security is only as strong as the hash function used. Proof systems that pick a hash for prover-efficiency reasons (arithmetization-friendly hashes, new constructions optimized for algebraic structure) are making a bet that the cryptanalytic interrogation those hashes receive will match the interrogation SHA-family hashes have received over the past two decades. Conservative hash choices are available; they cost prover performance.

Soundness vs. completeness. STARK verifiers accept proofs of true statements with probability one; they accept proofs of false statements with negligible probability. *Negligible* is a concrete number in the deployed system (typically 2^{-80} or 2^{-100}), and the system designer must commit to it. For evidence regimes where 2^{-80} is an acceptable soundness error, the guarantee is strong; for regimes where it is not, the parameters must be adjusted.

Trust in the verifier implementation. The soundness argument is about the protocol, not the code. A bug in a proof-system verifier implementation can accept invalid proofs regardless of what the underlying mathematics guarantees. This risk is shared with every cryptographic primitive ever deployed, and the mitigation is the same: multiple independent implementations, formal verification of critical verifier components, and adversarial testing.

These are real limits. None of them is a non-rotatable root of trust fused into physical silicon. That is the asymmetry the rest of the paper elaborates.

7. The Honest Comparison

The argument so far has been long-term. The short-term picture is more nuanced, and the honest comparison across capability and durability dimensions is what a production architect or governance author should actually work from.

7.1 A Trust-Aging Taxonomy

We propose a three-level taxonomy for how a cryptographic trust substrate ages. The taxonomy is deliberately informal: it is a conceptual scaffolding rather than a formal classification, and its value is in forcing the aging question to be named in architectural decisions.

- **Ages gracefully** **Ages gracefully.** Trust is rooted in an assumption whose cryptanalytic degradation is polynomial under known adversary models (classical and quantum), and whose parameters can be grown within existing deployed systems without physical replacement. The canonical example is hash-based cryptography under the Grover/BHT bounds.
- **Bounded aging** **Bounded aging.** Trust is rooted in an assumption whose degradation is polynomial classically, but whose parameters are tied to non-rotatable artifacts—deployed key sizes, embedded constants, physical-layer protocols—that can be migrated only through a coordinated refresh. Many TLS deployments historically sat here: the primitive was fine, but rotation required coordinated action.
- **Brittle** **Brittle.** Trust is rooted in an assumption for which an efficient adversary algorithm is known in a class of machines expected to exist in the future (i.e., Shor-broken primitives), and the substrate is bound to non-rotatable physical artifacts. Historical evidence produced on this substrate cannot be re-signed from the future.

Under this taxonomy, STARK-based verification sits firmly in the **Ages gracefully** category, contemporary TEE attestation in the **Brittle** category, and post-quantum TEE attestation on future silicon in the **Bounded aging** category (the silicon is still non-rotatable, but the assumption is expected to hold under quantum attack).

7.2 Side-by-Side

Table 1 summarizes the two substrates across the dimensions that matter for a production architect making a verifiable-AI choice in 2026.

7.3 What the Comparison Argues

The honest framing is therefore not TEEs versus STARKs as rival technologies competing for a single production slot. They are complementary substrates with different aging profiles, different failure modes, and different fit for different evidentiary horizons; the architectural question is composition, not selection. TEEs solve the verifiable-AI problem today for the majority of deployment contexts. STARK-based systems solve it post-quantum and trustlessly for the subset that demand durable evidence. Public timestamping and transparency-log anchoring (§5) link the two by extending the evidentiary lifetime of hardware-attested artifacts already being produced on silicon whose root cannot be migrated. The open engineering question is how aggressively to invest in closing the performance and tooling gap for the mathematics-rooted leg of that composed architecture.

The consequence asymmetry. What should settle the prioritization argument is the asymmetry of consequences. Getting it wrong on TEEs as the long-horizon evidentiary

substrate produces a structural credibility crisis for AI evidence in the late quantum era: historical attestations across an entire class of deployments become indistinguishable from forgeries, and there is no protocol move available to recover them. Getting it wrong on accelerating STARK-based verification produces some duplicated R&D investment and a few years of slower progress on a capability the field will pursue anyway. Current funding and research-attention patterns in the commercial verifiable-AI space implicitly invert that asymmetry: the substrate whose trust decays is where the capital and the engineering effort are concentrated.

The asymmetry, stated plainly: *for deployment contexts with long-horizon evidentiary requirements, the cost of underinvesting in mathematics-rooted verifiable AI is bounded by research and engineering effort; the cost of overrelying on hardware-rooted verifiable AI is bounded only by the value of the historical AI evidence that becomes repudiable the day a capable quantum adversary exists. These two bounds are not of the same order.*

We state this as a judgment about cost structure, not a theorem. It depends on priors over timelines, adversary models, and the value of the evidence at stake. Reasonable readers may disagree. The point of naming it explicitly is to force the disagreement to happen at the level of those priors, rather than at the level of which substrate is “faster” or “cheaper” today—which is the level at which the conversation currently happens.

8. Beyond the Quantum Question

The quantum threat is the sharpest technical argument for migrating long-horizon verifiable AI to mathematics-rooted foundations, but the case does not rest on it alone. Two further considerations hold independently of whether, and when, capable fault-tolerant quantum computation arrives: the *topology* of trust, and the ongoing *side-channel* arms race. A reader who assigns low prior to the quantum scenario should still take the argument in this section seriously on its own terms.

8.1 Trust Topology

A TEE attestation is a statement signed by a hardware vendor. Verifying it means verifying a chain that terminates at Intel, AMD, or NVIDIA. For many purposes, this is acceptable. For some, it is not.

The deployments where vendor-rooted trust becomes a structural compromise share a common property: they are precisely the deployments whose entire point is to remove trusted intermediaries. Sovereign AI deployments where a nation does not want to trust the chip manufacturer of a potentially adversarial power. Cross-jurisdictional model audits where neither jurisdiction accepts the other’s regulatory reach over the silicon vendor. Evidentiary processes where the opposing party is the same entity that controls the attestation root. Scientific reproducibility claims that must hold across the bankruptcy, acquisition, or end-of-life of the vendor ecosystem.

In each of these, rooting trust in any single vendor is a compromise, and the compromise is structural rather than goodwill-based. This is true regardless of whether the vendor has any present intention of misusing the trust. Architectures should be evaluated on their

assumptions, not on the character of their custodians.

A mathematics-rooted verification protocol moves the trust assumption from *this vendor’s silicon is honest and uncompromised* to *this public mathematical stack remains sound*. The second is publicly stateable, vendor-neutral, testable by anyone in the world with the requisite training, and falsifiable by counterexample rather than by trust elicitation. It is the kind of assumption that scales across borders, across jurisdictions, and across generations.

8.2 Side Channels

The last decade of TEE history is, in part, a history of a disclosed arms race over speculative execution leaks, power analysis, voltage glitching, and architectural-state attacks (Borrello et al., 2022; Kocher et al., 2019; Moghimi, 2023; Murdock et al., 2020; Van Bulck et al., 2018; Van Schaik et al., 2020). Each published disclosure has been, in its context, responsibly reported and patched. Each has also served as a reminder that the abstraction *this code ran inside an enclave and was not observed* is more fragile than the attestation surface implies.

The honest position here is not that TEE vendors have failed to address side channels. They have addressed each specific channel as it has been disclosed; the process has worked in the sense that public disclosures have been followed by mitigations. The less comfortable position is that the underlying architectural surface—shared speculative hardware, shared caches, shared power delivery, shared microarchitectural state—is a continuously expanding research frontier, and that the closed period between vulnerability discovery and public disclosure is not obviously short from the perspective of an attestation-evidence regime trying to argue that historical enclaves were never observed.

A mathematics-rooted proof system is, by construction, indifferent to most of this. A STARK does not assert that a computation was hidden from observation; it asserts that a computation was performed, with the integrity property targeted, and that the claim is independently verifiable from the proof alone. The side-channel arms race is primarily a concern for *confidentiality* claims about computations. Its relevance to *integrity* claims is more bounded: a STARK verifier’s guarantee rests on the mathematics alone, but a prover executing on compromised hardware remains vulnerable to fault-injection attacks of the Plundervolt family. The concern here is *pipeline integrity*, not proof soundness. Generic arithmetic corruption inside the proof system typically causes prover failure or an invalid proof; to obtain a valid proof of a false statement, the attack has to reach the surrounding pipeline—trace capture, witness generation, or the code that binds the public statement to the proof. A hybrid architecture must account for this: the integrity of the pipeline that constructs the statement being proved is a separate surface from verifier integrity after it, and from proof soundness as a cryptographic property.

We emphasize that this is not an indictment of TEE engineering. The people working on SGX, TDX, SEV-SNP, and H100 confidential computing are not making mistakes a better team would avoid. They are solving a genuinely hard problem against a continuously expanding adversary model. Our point is that *hardware-rooted trust* and *mathematics-rooted trust* are different categories of guarantee, with different failure modes and different aging behavior. A mature verifiable-AI infrastructure will use both, deliberately, with explicit reasoning about which guarantee each component of the system actually provides.

9. A Path Forward

The practical path forward is not a flag day on which TEEs are abandoned and STARKs are switched on. It has three layers, each with a different time horizon and a different set of actors.

9.1 Layer 1: Hybrid Architectures

The immediate step, feasible now, is hybrid. TEE-attested inference for live serving, where latency and cost dominate and evidentiary horizons are short. Asynchronous STARK proofs produced in parallel or post-hoc for the subset of computations whose evidence must endure: regulated decisions, model-provenance snapshots, training-run attestation, audit samples, evidentiary captures for disputes. The STARK proof does not need to block the inference path. It needs to exist at the moment future verification requires it.

Architecturally, this looks like a serving stack that emits a TEE attestation per request and, for a configurable subset of requests, also emits a STARK proof of the same computation into a long-term evidence archive. The configurability is the design surface. An enterprise deploying a model in a low-stakes analytics setting can set the STARK-proof rate to zero without losing anything. An agency deploying a model in a setting subject to FOIA, judicial review, or independent audit can set the rate to one. A regulator can mandate the rate by deployment context.

This hybridization captures the latency and cost advantages of TEEs for live workloads while building a durable mathematical evidence base for the computations that future verification will actually probe. The design does not require a new proof system, a new hardware architecture, or a new standard. It requires deliberate system-level choices about what evidence to produce at what frequency for what retention horizon. The prover’s own execution integrity must be protected on the same footing as the TEE enclave (§8): a fault attack on trace capture, witness generation, or statement-to-proof binding can yield a valid proof of a false statement even though the proof system itself remains sound, so a serious hybrid deployment runs proof generation inside an attested environment even when the proof itself is the long-horizon artifact.

By the same aging-behavior argument, and as the **Scope** paragraph in §1 flagged, the hybrid pattern extends to training provenance: a STARK proof over a training-step trace, sampled rather than exhaustively, gives the same aging profile for training-time evidence that per-call proofs give for inference. We note this as an inherited extension rather than a prescription; the training threat model is adjacent to, but not identical to, the inference threat model this paper formalizes.

A third operational layer can be turned on today, before any STARK infrastructure is in place: anchoring TEE attestation reports into a broadly witnessed public timestamping log, as developed in §5. OpenTimestamps over Bitcoin (Nakamoto, 2008; Todd, 2016) gives a concrete, near-zero-cost-per-anchor path: the serving stack batches attestation digests and anchors the batch root in Bitcoin at whatever cadence the evidentiary context requires. The anchor does not substitute for a mathematics-rooted soundness argument, but it closes the retroactive-forgery window for the archival record already being produced on classical-ECC silicon, because a forgery manufactured after quantum capability arrives cannot retroac-

tively appear in a block already mined. Hardware-rooted attestation, mathematics-rooted proof, and public timestamp anchoring compose into a defense-in-depth architecture whose failure modes are *partially independent*: no single cryptanalytic advance against classical ECC or against any one proof-system primitive collapses all three, and each layer carries the evidentiary load where the others are weakest. We stress that independence here is partial rather than total—the STARK layer and the timestamping layer both rest on long-lived hash assumptions, and the anchoring layer additionally assumes the continued public accessibility of the chosen log or chain. A systemic break in hash cryptanalysis, or the disappearance of a log whose anchors were the only durable evidence, would degrade the mathematics-rooted and timestamping legs simultaneously.

9.2 Layer 2: Accelerated Investment in STARK-Friendly ML

The medium-term step is to close the performance gap between TEE inference and STARK-proved inference. This is a technical research program, currently concentrated in a small number of academic groups and crypto-native companies, with several load-bearing open problems.

Arithmetization of common ML operations. Matrix multiplication, softmax, layer normalization, GELU, attention, and their fixed-point variants need arithmetizations that are efficient inside the specific algebraic structure of the proof system, not merely encodable. Recent work on custom gates, lookup arguments, and arithmetization-friendly activations (Gabizon and Williamson, 2020; Haböck, 2022; Haböck et al., 2024) has produced substantial prover speedups; more is available.

Hash function choice. Prover efficiency depends heavily on the hash function used for commitments. Hashes optimized for proof-system performance (e.g., Poseidon2, Rescue, Vision, GKR-friendly hashes) (Aly et al., 2019; Grassi et al., 2021) are efficient but young; conservative hashes (SHA-3, BLAKE3) are well-studied but expensive inside proof systems. The trade-off has not been fully characterized for post-quantum settings.

Field choice. Work on small-field arithmetizations (Mersenne-31, BabyBear, Goldilocks) and the recent Circle-STARK family (Haböck et al., 2024) has substantially improved prover throughput on modern CPUs and GPUs. Further work integrating these with ML-specific bit-widths (typically 8-bit or 16-bit fixed-point or float) would further close the gap.

Compiler and tooling integration. Proof-system toolchains that integrate with standard ML compilers (XLA, TVM, MLIR, StableHLO) and that can emit proof-provable programs from unmodified PyTorch or JAX source—with correctness and performance guarantees—do not yet exist at the level of maturity that ONNX-style interchange formats have reached for plain inference. The zkVM ecosystem (RISC Zero, Inc., 2023; Succinct Labs, 2024) has made important progress on the general-purpose side.

Prover-friendly model architectures. Models designed with proof-system friendliness as an explicit co-design objective (Sun et al., 2024)—activation choice, quantization schedule, attention structure—may offer substantial prover speedups at modest task-performance cost. The research frontier here is wide open.

Each of these is a multi-year research track. Taken together, they are the technical program

that would bring mathematics-rooted verifiable AI from “research” to “deployable” on a timescale relevant to the quantum transition.

9.3 Layer 3: Cryptographic Agility in AI Evidence Regimes

The governance step is to treat cryptographic agility as a first-order design requirement in AI evidence regimes being drafted now. The EU AI Act (European Parliament and Council of the European Union, 2024), the NIST AI Risk Management Framework (National Institute of Standards and Technology, 2023), and emerging sector-specific compliance standards are being written at a moment when the cryptographic assumptions underlying the evidence they reference are known to have finite lifetimes. A framework that treats a TEE attestation and a mathematical proof as interchangeable “technical measures” is encoding an equivalence that is false along the temporal dimension.

Concretely, we suggest three principles.

Substrate-aware evidentiary regimes. Rules about what constitutes acceptable AI evidence should distinguish between evidence whose validity is contingent on a non-rotatable cryptographic root and evidence whose validity is contingent on a rotatable mathematical assumption. The distinction is technical; the governance implication is that the retention period and dispute-resolution procedures should differ.

Horizon-matched substrate selection. For a given regulatory or evidentiary context, the chosen verifiable-AI substrate should be matched to the verification horizon. Short-horizon contexts (operational monitoring, live adjudication) are well served by TEEs. Long-horizon contexts (records retention, judicial admissibility, cross-border disclosure) benefit from mathematics-rooted or hybridized evidence.

Counterparty diversity of verification. In contexts where the verifier and the prover may be the same organization (provider-signed attestations verified by the provider’s own SDK), the governance regime should require a second verification path that does not rely on the same trust root. A mathematics-rooted proof admits an arbitrary third-party verifier at low cost; a hardware-rooted attestation requires the verifier to trust the hardware vendor. Where the deployment is adversarial to the vendor or to the provider, this asymmetry is load-bearing.

These are not policy prescriptions at the level of specific rule text. They are principles that should inform rule-drafters and rule-interpreters as the next generation of AI accountability infrastructure is written. The window during which these principles can be written into foundational frameworks is finite, and it coincides with the drafting windows of 2025–2028.

10. Why This Matters Beyond Cryptography

The AI systems we are deploying now mediate access to information, healthcare, credit, judicial processes, and economic opportunity. They produce evidence that future generations will need to interrogate. The decisions those systems make today will be adjudicated for decades. The models trained today will be subject to provenance claims long after the hardware they trained on has been recycled.

Whether that evidence remains meaningful is a question about the kind of accountability we are able to encode into the substrate of the digital world. It is not a question that can be deferred to a later revision, because the evidence regimes being written now will govern the retroactive examinability of everything those regimes cover.

If we build long-horizon verifiable AI on foundations whose trust expires when a hardware generation ages out, or when a particular cryptanalytic capability matures, we are quietly handing future actors the ability to deny, forge, or rewrite the record. That is a power no institutional design—regulatory, judicial, journalistic—has ever managed to constrain after the fact, and the architectural default once set is very hard to undo.

If we build long-horizon verifiable AI on foundations whose trust is rooted in mathematical assumptions that age gracefully and are publicly verifiable across borders and generations, we preserve a different possibility: the record of what AI systems did, on what inputs, with what outputs, remains examinable by anyone, anywhere, for as long as the mathematics holds. This is not a utopian claim. It is a claim about which cryptographic assumption carries the evidentiary weight.

This is the deeper reason the half-life question is not merely a technical one. It is a choice about the time horizon of accountability. About whether the evidentiary substrate of the AI era is built to outlive the corporations and governments that operate it. About whether the truth of what machines did in our time will remain available to the people who come after.

The pragmatic answer for today is TEEs. The right answer for the durability of the record is mathematics-rooted, post-quantum verification. Both should be said plainly. And the field should be moving on the second faster than current investment patterns suggest it is.

11. Conclusion

We have argued that the *aging behavior* of a verifiable-AI trust substrate is a first-class engineering property, and that current discourse has not treated it as such. Hardware-rooted trusted execution environments, the pragmatic production substrate of 2026, embed classical public-key cryptography into non-rotatable silicon, and therefore age brittly: a capable quantum adversary at any point in the future renders archival attestations retroactively forgeable, without a protocol move available to recover them. Mathematics-rooted verifiable computation, exemplified by STARK-family proof systems whose soundness rests on a hash-dominated mathematical assumption stack, ages gracefully under the best understood quantum attacks: parameters can be grown, the math carries forward, and no hardware refresh is on the critical path for preserving existing evidence.

The frame is not TEEs versus STARKs. They are complementary substrates with different aging profiles, different failure modes, and different fit for different evidentiary horizons, and the architectural question is composition, not selection. The frame is *which trust ages with the evidence we want to carry forward*, and under that frame the answer is composed: TEEs for ephemeral, cost-sensitive, latency-bound verification; STARK-family proofs for durable, cross-jurisdictional, long-horizon evidence; and public timestamp anchoring as the bridge that extends the evidentiary lifetime of hardware-attested artifacts already being produced

on silicon whose root cannot be migrated. Getting this architectural distinction embedded into the AI evidence regimes drafted over the next three years is the governance challenge. Closing the performance gap between the two substrates for frontier-scale inference is the research challenge. Building the hybrid deployment tooling—TEE, STARK, and public-timestamp layers jointly—is the engineering challenge.

None of these is solved. Each is tractable. The cost of working on them now is a few years of concentrated research and engineering effort. The cost of deferring them is structural repudiability of the historical record of AI decisions, for the period between the emergence of capable quantum computation and whenever post-quantum hardware has reached installed-base saturation. That interval is precisely the interval during which the first generation of serious AI governance evidence will accumulate.

We close with the observation that motivates the title. Trust has a half-life. Some substrates decay with their hardware. Others decay with our understanding of mathematics. When we choose which substrate to carry our evidence forward, we are choosing which decay we are willing to live with. Math is patient. Hardware is not. We get to choose—now, while the choice is still available—which one we trust to carry the record forward.

Acknowledgments. This paper has benefited from long conversations with colleagues across the cryptographic verification and AI safety communities. The framing of *trust aging* as a first-class property emerged from a sequence of discussions over 2024–2026 that it would be impossible to enumerate fairly; the responsibility for any remaining confusions is mine alone.

References

-
- Advanced Micro Devices. AMD SEV-SNP: Strengthening VM isolation with integrity protection and more. White paper, Advanced Micro Devices, Inc., January 2020. URL <https://www.amd.com/content/dam/amd/en/documents/epyc-business-docs/white-papers/SEV-SNP-strengthening-vm-isolation-with-integrity-protection-and-more.pdf>.
- Abdelrahman Aly, Tomer Ashur, Eli Ben-Sasson, Siemen Dhooghe, and Alan Szepieniec. Design of symmetric-key primitives for advanced cryptographic protocols. Cryptology ePrint Archive, Paper 2019/426, 2019. URL <https://eprint.iacr.org/2019/426>.
- Anthropic and Irregular. Confidential inference systems: Design principles and security risks. White paper, Anthropic, PBC, June 2025. URL https://assets.anthropic.com/m/c52125297b85a42/original/Confidential_Inference_Paper.pdf.
- Abdelhamid Bakhta. Toward high-assurance AI: Safety by design for autonomous systems. Preprint, April 2026. Companion paper on the composed assurance stack.
- William Barker, William Polk, and Murugiah Souppaya. Getting ready for post-quantum cryptography: Exploring challenges associated with adopting and using post-quantum cryptographic algorithms. Cybersecurity White Paper NIST CSWP 15, National Institute

- of Standards and Technology, April 2021. URL <https://csrc.nist.gov/pubs/cswp/15/getting-ready-for-postquantum-cryptography/final>.
- Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable, transparent, and post-quantum secure computational integrity. Cryptology ePrint Archive, Paper 2018/046, 2018. URL <https://eprint.iacr.org/2018/046>.
- Pietro Borrello, Andreas Kogler, Martin Schwarzl, Moritz Lipp, Daniel Gruss, and Michael Schwarz. ÆPIC leak: Architecturally leaking uninitialized data from the microarchitecture. In *Proceedings of the 31st USENIX Security Symposium*, pages 3917–3934. USENIX Association, 2022. URL <https://www.usenix.org/conference/usenixsecurity22/presentation/borrello>.
- Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum cryptanalysis of hash and claw-free functions. In *LATIN’98: Theoretical Informatics, 3rd Latin American Symposium*, volume 1380 of *Lecture Notes in Computer Science*, pages 163–169. Springer, 1998. doi: 10.1007/BFb0054319.
- David A. Cooper, Daniel C. Apon, Quynh H. Dang, Michael S. Davidson, Morris J. Dworkin, and Carl A. Miller. Recommendation for stateful hash-based signature schemes. Special Publication SP 800-208, National Institute of Standards and Technology, October 2020. URL <https://csrc.nist.gov/pubs/sp/800/208/final>.
- Victor Costan and Srinivas Devadas. Intel SGX explained. Cryptology ePrint Archive, Paper 2016/086, 2016. URL <https://eprint.iacr.org/2016/086>.
- European Parliament and Council of the European Union. Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). Official Journal of the European Union, July 2024. URL <https://eur-lex.europa.eu/eli/reg/2024/1689/oj>.
- European Telecommunications Standards Institute. Electronic signatures and infrastructures (ESI); time-stamping protocol and time-stamp token profiles. Standard EN 319 422, ETSI, 2016. URL https://www.etsi.org/deliver/etsi_en/319400_319499/319422/.
- Ariel Gabizon and Zachary J. Williamson. plookup: A simplified polynomial protocol for lookup tables. Cryptology ePrint Archive, Paper 2020/315, 2020. URL <https://eprint.iacr.org/2020/315>.
- Craig Gidney. How to factor 2048 bit RSA integers with less than a million noisy qubits. arXiv preprint arXiv:2505.15917, May 2025. URL <https://arxiv.org/abs/2505.15917>. Google Quantum AI.
- Craig Gidney and Martin Ekerå. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. *Quantum*, 5:433, 2021. doi: 10.22331/q-2021-04-15-433. URL <https://quantum-journal.org/papers/q-2021-04-15-433/>.
- Tobias Gondrom, Ralf Brandner, and Ulrich Pordesch. Evidence record syntax (ERS). RFC 4998, Internet Engineering Task Force, August 2007. URL <https://www.rfc-editor.org/rfc/rfc4998>.
- Google Quantum AI. Quantum error correction below the surface code threshold. *Nature*,

- 638:920–926, 2025. doi: 10.1038/s41586-024-08449-y. Willow processor; published online December 2024.
- Lorenzo Grassi, Dmitry Khovratovich, Christian Rechberger, Arnab Roy, and Markus Schofnegger. Poseidon: A new hash function for zero-knowledge proof systems. In *Proceedings of the 30th USENIX Security Symposium*, pages 519–535. USENIX Association, 2021. URL <https://www.usenix.org/conference/usenixsecurity21/presentation/grassi>.
- Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing (STOC)*, pages 212–219. ACM, 1996. doi: 10.1145/237814.237866.
- Ulrich Haböck. Multivariate lookups based on logarithmic derivatives. Cryptology ePrint Archive, Paper 2022/1530, 2022. URL <https://eprint.iacr.org/2022/1530>.
- Ulrich Haböck, David Levit, and Shahar Papini. Circle STARKs. Cryptology ePrint Archive, Paper 2024/278, 2024. URL <https://eprint.iacr.org/2024/278>.
- Intel Corporation. Intel software guard extensions data center attestation primitives (Intel SGX DCAP). Technical documentation, Intel Corporation, 2020. URL <https://www.intel.com/content/www/us/en/developer/tools/software-guard-extensions/attestation-services.html>.
- Intel Corporation. Intel trust domain extensions (Intel TDX): Architecture specification. Architecture specification, Intel Corporation, 2023. URL <https://www.intel.com/content/www/us/en/developer/tools/trust-domain-extensions/documentation.html>.
- Paul Kocher, Jann Horn, Anders Fogh, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, Michael Schwarz, and Yuval Yarom. Spectre attacks: Exploiting speculative execution. In *Proceedings of the 40th IEEE Symposium on Security and Privacy (S&P)*, pages 1–19. IEEE, 2019. doi: 10.1109/SP.2019.00002.
- Leslie Lamport. Constructing digital signatures from a one-way function. Technical Report CSL-98, SRI International, Menlo Park, CA, October 1979. URL <https://www.microsoft.com/en-us/research/publication/constructing-digital-signatures-one-way-function/>.
- Ben Laurie, Adam Langley, and Emilia Käsper. Certificate transparency. RFC 6962, Internet Engineering Task Force, 2013. URL <https://www.rfc-editor.org/rfc/rfc6962>.
- Tianyi Liu, Xiang Xie, and Yupeng Zhang. zkCNN: Zero knowledge proofs for convolutional neural network predictions and accuracy. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 2968–2985. ACM, 2021. doi: 10.1145/3460120.3485379.
- Ralph C. Merkle. A digital signature based on a conventional encryption function. In *Advances in Cryptology — CRYPTO ’87*, volume 293 of *Lecture Notes in Computer Science*, pages 369–378. Springer, 1988. doi: 10.1007/3-540-48184-2_32.

- Daniel Moghimi. Downfall: Exploiting speculative data gathering. In *Proceedings of the 32nd USENIX Security Symposium*. USENIX Association, 2023. URL <https://www.usenix.org/conference/usenixsecurity23/presentation/moghimi>.
- Dustin Moody, Ray Perlner, Andrew Regenscheid, Angela Robinson, and David Cooper. Transition to post-quantum cryptography standards. Initial Public Draft NIST IR 8547 ipd, National Institute of Standards and Technology, November 2024. URL <https://csrc.nist.gov/pubs/ir/8547/ipd>.
- Michele Mosca. Cybersecurity in an era with quantum computers: Will we be ready? *IEEE Security & Privacy*, 16(5):38–41, 2018. doi: 10.1109/MSP.2018.3761723.
- Kit Murdock, David Oswald, Flavio D. Garcia, Jo Van Bulck, Daniel Gruss, and Frank Piessens. Plundervolt: Software-based fault injection attacks against Intel SGX. In *Proceedings of the 41st IEEE Symposium on Security and Privacy (S&P)*, pages 1466–1482. IEEE, 2020. doi: 10.1109/SP40000.2020.00057.
- Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Self-published white paper, 2008. URL <https://bitcoin.org/bitcoin.pdf>.
- National Institute of Standards and Technology. Artificial intelligence risk management framework (AI RMF 1.0). Technical Report NIST AI 100-1, U.S. Department of Commerce, January 2023. URL <https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-1.pdf>.
- National Institute of Standards and Technology. Module-lattice-based digital signature standard. Federal Information Processing Standards Publication (FIPS) 204, U.S. Department of Commerce, August 2024a. URL <https://csrc.nist.gov/pubs/fips/204/final>.
- National Institute of Standards and Technology. Module-lattice-based key-encapsulation mechanism standard. Federal Information Processing Standards Publication (FIPS) 203, U.S. Department of Commerce, August 2024b.
- National Institute of Standards and Technology. Stateless hash-based digital signature standard. Federal Information Processing Standards Publication (FIPS) 205, U.S. Department of Commerce, August 2024c.
- National Security Agency. Commercial national security algorithm suite 2.0. Cybersecurity advisory, National Security Agency / Central Security Service, September 2022. URL <https://www.nsa.gov/Press-Room/Press-Releases-Statements/Press-Release-View/Article/3148990/>.
- NVIDIA Corporation. NVIDIA H100 tensor core GPU architecture: Confidential computing on NVIDIA Hopper H100. White Paper WP-11459-001, NVIDIA Corporation, 2023. URL <https://images.nvidia.com/aem-dam/en-zz/Solutions/data-center/HCC-Whitepaper-v1.0.pdf>.
- RISC Zero, Inc. RISC Zero zkVM: Scalable, transparent arguments of RISC-V integrity. Technical White Paper (Draft), 2023. URL <https://dev.risczero.com/proof-system-in-detail.pdf>.
- Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms

- on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997. doi: 10.1137/S0097539795293172.
- Marc Stevens, Elie Bursztein, Pierre Karpman, Ange Albertini, and Yarik Markov. The first collision for full SHA-1. In *Advances in Cryptology — CRYPTO 2017*, volume 10401 of *Lecture Notes in Computer Science*, pages 570–596. Springer, 2017. doi: 10.1007/978-3-319-63688-7_19.
- Succinct Labs. SP1: A performant, open-source zero-knowledge virtual machine. Succinct Labs technical documentation, 2024. URL <https://blog.succinct.xyz/introducing-sp1/>.
- Haochen Sun, Jason Li, and Hongyang Zhang. zkLLM: Zero knowledge proofs for large language models. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2024. doi: 10.1145/3658644.3670334.
- The White House. National security memorandum on promoting united states leadership in quantum computing while mitigating risks to vulnerable cryptographic systems (NSM-10). Presidential Memorandum, May 2022. URL <https://bidenwhitehouse.archives.gov/briefing-room/statements-releases/2022/05/04/national-security-memorandum-on-promoting-united-states-leadership-in-quantum-computing-v>
- Peter Todd. OpenTimestamps: Scalable, trust-minimized, distributed timestamping with Bitcoin. Open-source protocol and reference implementation, 2016. URL <https://opentimestamps.org/>.
- UK National Cyber Security Centre. Timelines for migration to post-quantum cryptography. Technical report, UK National Cyber Security Centre, November 2024. URL <https://www.ncsc.gov.uk/guidance/pqc-migration-timelines>.
- Jo Van Bulck, Marina Minkin, Ofir Weisse, Daniel Genkin, Baris Kasikci, Frank Piessens, Mark Silberstein, Thomas F. Wenisch, Yuval Yarom, and Raoul Strackx. Foreshadow: Extracting the keys to the Intel SGX kingdom with transient out-of-order execution. In *Proceedings of the 27th USENIX Security Symposium*, pages 991–1008. USENIX Association, 2018. URL <https://www.usenix.org/conference/usenixsecurity18/presentation/bulck>.
- Stephan Van Schaik, Andrew Kwong, Daniel Genkin, and Yuval Yarom. SGAXe: How SGX fails in practice. Technical report, University of Michigan and University of Adelaide, 2020. URL <https://sgaxe.com/files/SGAxe.pdf>.
- Xiaoyun Wang and Hongbo Yu. How to break MD5 and other hash functions. In *Advances in Cryptology — EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 19–35. Springer, 2005. doi: 10.1007/11426639_2.

Dimension	Hardware-rooted (TEE)	Math-rooted (STARK)
Pragmatic production readiness (2026)	Mature; deployable at frontier scale	Research / early production; frontier-scale readiness remains open
Verifier cost	Light (certificate chain check)	Light (polylog in computation)
Prover / execution cost	Near-native (single-digit % overhead)	Heavy but falling fast; specialized infrastructure still needed
Trust anchor	Manufacturer + non-rotatable fused key	Public math stack: conservative hash assumptions, Fiat-Shamir / RO modeling, IOP soundness, chosen parameters
Trust topology	Vendor-rooted; chain terminates at a CA	Vendor-neutral; anyone can verify
Side-channel exposure	Ongoing arms race (Fore-shadow, Plundervolt, ÆPIC, Downfall, ...)	Verifier-side: none. Prover-side: fault injection on proof generation is a separate, localizable surface
Aging under quantum adversary	Brittle (classical ECC; retroactive forgery)	Ages gracefully (polynomial degradation)
Rotatability of root	Hardware refresh cycle (years)	Software parameter change
Suitable for ephemeral evidence	Yes (production baseline)	Yes, but cost not yet justified
Suitable for long-horizon evidence	Fragile without strong countersignature regime	Native fit

Table 1: Hardware-rooted and mathematics-rooted verifiable AI across the dimensions that matter for an architect making a substrate choice today. The table is meant to be read as descriptive, not prescriptive: no dimension alone dictates the choice, and for most real deployments the right architecture is hybrid (§9).