Who Is Signing?

Daniel Hardman

2026-02-07

Abstract

Explores the reasons why a simple answer to the question of which entity is in control of cryptographic keys may limit our insight.

In cryptographic systems, one of the most important questions we can ask is deceptively simple:

> Who is signing?

Asking the question is insightful. Expecting a single, clean answer is usually a mistake.

In modern identity systems, signatures sit at the intersection of keys, delegation, intent, governance, and accountability. Understanding that intersection is essential if we want trust to be based on evidence rather than assumption.

This article offers a mental model for thinking clearly about "who is signing"—one that applies across decentralized identity, PKI, verifiable credentials, automation, and agentic systems.

A human analogy

Suppose a pilot named Alice is assigned to captain a flight. The plane belongs to Swiss Air, a subsidiary of Lufthansa. The flight is marketed by Air Dolomiti, another Lufthansa subsidiary. Alice brings along a trainee pilot and, at one point, hands over the controls for thirty seconds while she speaks with a flight attendant.

If I ask, during that interval, *who is flying the plane*, which answer is correct?

- Alice?
- The trainee?
- Swiss Air?
- Air Dolomiti?
- Lufthansa?
- All of the above?

The answer depends on context. A lawyer, a regulator, a passenger, and an insurer may all give different—and reasonable—answers. The complexity arises because *control, authority, responsibility, and attribution overlap but do not coincide*.

Cryptographic signing embodies exactly this same dynamic.

The naive model—and its limits

The simplest model of signing goes like this:

- A signature proves that someone controls a private key.
- Therefore, the signer is whoever controls the key.

This is a good first step. It already distinguishes self-managed keys from custodial wallets, and direct control from outsourced infrastructure.

But it breaks down quickly.

Organizations do not act directly. They act through employees, contractors, service providers, automation, and APIs. Keys are held in HSMs, protected by policy engines, triggered by workflows, and sometimes unlocked by people who never see the data being signed.

So "who controls the key?" is necessary—but not sufficient.

Signing can have several meanings

One reason the question is hard is that signing proves someone processed content, but *it does not inherently clarify their intent with respect to that content*.

A signature may mean:

- "I created this."
- "I witness that I saw this content."
- "I assert this is true."
- "I endorse this."
- "I authorize this action."
- "I am acting on behalf of someone else."

Consider signing a petition. You did not write it. You may not agree with every sentence. But your signature expresses endorsement. The meaning of the signature is *not authorship*—it is *stance*.

Cryptographic systems inherited a bias to assume signing was an assertion of authoriship from early message-signing use cases. In credential contexts, specifications often do make the intent of a signature clear. However, occasional naive gaps surface. Identity systems cannot afford to neglect this nuance. [1]

A lifecycle view of signing

In robust identity architectures, signatures typically appear at **multiple points in an evidence lifecycle**, and each plays a different role [2].

1. **Issuers sign credentials**
   These signatures assert facts about the world: legal identity, control of a resource, rights to use a brand. The signer is an authoritative source, not the subject.
2. **Subjects sign collections of evidence**
   When an entity assembles credentials into a dossier or similar structure, its signature is not

claiming authorship of the underlying facts. It is saying: *this is the set of evidence I stand behind, together, for this purpose*.

3. **Delegated signers sign live artifacts**
   Operational signatures—on transactions, messages, or real-time protocols—are often produced by infrastructure acting under delegated authority. These signatures are proximate and time-sensitive, but they derive meaning from earlier evidence and delegation.

At each step, asking "who is signing?" yields a different answer. That is not a flaw; it is the point.

Delegation is normal—opacity is the risk

Delegation is unavoidable. No serious organization can function without it.

The real question is not whether delegation exists, but *whether it is inspectable*.

Some systems externalize governance. For example, multisignature identifiers can make signing thresholds, role separation, and recovery mechanisms visible and verifiable. Observers can see not just *that* something was signed, but *how authority is distributed* [3].

Unfortunately, most tech in common use today hides governance behind a single key. A sophisticated policy engine may exist internally—requiring approvals, enforcing limits, or coordinating humans—but externally, the result is indistinguishable from a lone script with a leaked secret [4, 5, 6].

From the outside, these two situations look the same.

This is not a critique of any specific tool. It is an observation about evidence. *Opaque control collapses meaningful distinctions*, and collapsed distinctions force trust decisions to rely on reputation and hope rather than inspection [7].

The same concern applies to agentic AI. The danger is not that an AI signs something. The danger is that we cannot tell whether a signature represents human intent, delegated authority, automated policy, or accidental execution.

Intent and its boundaries

Every signature claims to represent intent—but intent is internal. Only the intender knows it fully.

That means signatures inevitably sit at *intent boundaries*: places where outsiders must infer purpose from action. [8] Good systems acknowledge these boundaries and make them narrow. Bad systems pretend they do not exist [9].

Delegation, endorsement, and automation all stretch intent across distance and time. The question is whether a system preserves enough structure to let others reason responsibly about what a signature does—and does not—mean.

A better question

So when we ask "who is signing?", we should really be asking a richer set of questions:

- Who controlled the key?
- In what role was that key acting?
- Whose intent is being represented?
- What authority links this signer to an accountable principal?
- Which of these relationships are inspectable, and which are assumed?

Systems that surface these answers do not eliminate ambiguity—but they make ambiguity *explicit and bounded*.

That, in the end, is what trustworthy identity infrastructure should do.

References

[1] Hardman, D. *Signing Doesn't Always Claim Authorship* Codecraft Papers, 2026. https://dhh1128.github.io/papers/sign-author.html

[2] Hardman, D. *The Evidence Lifecycle in Telco*. Codecraft Papers, 2025. https://dhh1128.github.io/papers/ev-life.html

[3] Smith, S. M. 2024. KERI Specification (v2.7.0). Trust Over IP Foundation. Retrieved December 18, 2024 from https://trustoverip.github.io/kswg-keri-specification/

[4] HashiCorp. n.d. *Transit secrets engine / Vault*. HashiCorp Developer Documentation. Retrieved February 7, 2026 from https://developer.hashicorp.com/vault/docs/secrets/transit

[5] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and Polk, W. 2008. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. RFC 5280. IETF. DOI: https://doi.org/10.17487/RFC5280

[6] Barnes, R., Hoffman-Andrews, J., McCarney, D., and Kasten, J. 2019. Automatic Certificate Management Environment (ACME). RFC 8555. IETF. DOI: https://doi.org/10.17487/RFC8555.

[7] Hardman, D. *Why Anchored Signatures?* Codecraft Papers, 2025. https://dhh1128.github.io/papers/was.html

[8] Hardman, D. 2025. *Intent and Boundaries: A Framework for Digital Agency*. SSRN Electronic Journal. DOI: https://doi.org/10.2139/ssrn.5909382

[9] Hardman, D. *Intent and Boundaries* Codecraft Papers, 2025. https://dhh1128.github.io/papers/intent-boundaries.html