



Correction to the paper: all nine axioms were individually inconsistent

The companion paper (*A Machine-Checked Necessity Result for Accountable Delegated Authority*, 28 Aug 2026) states nine `axiom` declarations and describes eight theorems derived from them as proofs from "standard," "documented," or "justified" assumptions. On 2026-09-07, independent review (`Consistency.lean`) found that each of the nine axioms, as literally stated over the file's own section variables, is individually **inconsistent** — not merely imprecise, not merely stronger than needed. Each axiom is generalized by Lean over every predicate of the matching type, not fixed to the one predicate the file actually reasons about. Instantiating each axiom at a deliberately pathological predicate of the right type (e.g. `C_functional` at the constantly-true signature predicate on `Bool`) derives `False` from that axiom alone, with no other axiom or `sorry` involved.

This is a correction, not a softening. An axiom from which `False` follows makes every theorem depending on it vacuously true regardless of its proof — the eight comparison/uniqueness theorems built on these axioms were not wrong in their mathematical content, but the axioms underneath them, as stated, could prove anything at all. The paper's prose describing these as reasonable, narrow, independently-justified assumptions ("Justified on standard unforgeability grounds", "standard content of the calculus", "the documented bearer property", etc.) was accurate about *what the axiom was meant to say*, and wrong about *what it actually asserted in Lean* — the two came apart because of universal generalization over section variables, not because the informal reasoning was flawed.

The sealed development (the paper as it stands) states 31 theorems: 8 depended on one of the nine axioms and are listed below; the other 23 never depended on any axiom (checked individually, not assumed) and are untouched by this correction — they are not listed below because nothing about them changes.

The repair, `A50.lean` v4 (SHA-256 `0e5ebe3f5113727ae0cd41ec4399661a352fcc4fb522fcfe7ef7be3318f6b452`), restates the 8 with explicit hypotheses, leaves the 23 unchanged, and adds two new theorems that exist nowhere in the sealed development: `quorum_two_governors` and `quorum_not_CFunctional`, documenting the boundary `CFunctional` now names on `O_unique`(\$Uniqueness, below, cites both). `#print axioms` on all 33 theorems in v4 — the sealed 31 plus these 2 additions — now reports zero axiom dependency; `Consistency.lean` fails to compile against it, since there is no axiom left to instantiate.

This correction proposes the following section-by-section amendments. It does not edit the sealed `.tex/.pdf`; the paper as sealed 29 August 2026 (tag `paper-sealed-2026-08-29`) is unrevised.

§Axioms — opening paragraph

Current:

Nine irreducible axioms remain in the development. Two are this work's own; seven instantiate a named prior system's documented mechanism, solely to derive a comparison result in §Distinguishing prior systems.

Proposed:

Nine axioms were stated in the sealed 28 August development. Each was generalized by Lean over every predicate of its declared type, not fixed to the one predicate this work reasons about — so each, as literally stated, was individually inconsistent. `Consistency.lean`, an independent review, found this on 2026-09-07 by instantiating each axiom at a pathological predicate and deriving `False`; see the correction record. Zero axioms remain: each is now an explicit hypothesis on the theorem(s) that consume it. Two were this work's own (`C_functional`, restated as `CFunctional`; `P2`); seven instantiate a named prior system's documented mechanism, solely to derive a comparison result in §Distinguishing prior systems.

§This work's two — `C_functional`

Current axiom:

None

```
axiom C_functional :
  forall (P P' : Principal) (A : Agent) (X : Action) (t : Time),
    C P A X t -> C P' A X t -> P = P'
```

Prose: "A valid signature verifies against exactly one principal's key. Justified on standard unforgeability grounds. ... `C_functional` is the narrower replacement from which uniqueness is now derived."

Proposed:

None

```
def CFunctional (C : Principal -> Agent -> Action -> Time -> Prop)
: Prop :=
  forall (P P' : Principal) (A : Agent) (X : Action) (t : Time),
    C P A X t -> C P' A X t -> P = P'
```

"A valid signature verifies against exactly one principal's key, for the signature predicate `C` this development actually reasons about — justified on standard unforgeability grounds. Stated as an axiom, this was individually inconsistent: Lean generalized it over every predicate of `C`'s type, including ones with no unforgeability property at all; instantiating it at the constantly-true predicate on `Bool` derives `False`. `CFunctional` is the named, non-axiomatic restatement, taken as a hypothesis by the one theorem (`O_unique`) that uses it."

§This work's two — P2

Current axiom:

None

```
axiom P2 :
  forall (P : Principal) (A : Agent) (X : Action) (t t' : Time),
    t < t' -> Appointed P A X t -> Revoke P A X t' -> not (R P A X
t')
```

Prose: "The binding must be revocable by the principal alone. ... The present form asserts who can trigger revocation."

Proposed: (unchanged Lean shape, no longer declared as `axiom`) "The binding must be revocable by the principal alone. Stated as an axiom, this was individually inconsistent for the same reason as `C_functional`: generalized over every `R`, `Appointed`, and `Revoke` of the right type, including ones for which the stated implication is simply false. Instantiated at constantly-true `R/Appointed/Revoke` with `Time := Nat`, it derives `False`. It is now an explicit hypothesis, `hP2`, on the two theorems that consume it, `revoked_defeats_0` and `revocation_blocks_enforcement`."

§The prior systems' seven

Current: lists the seven axiom names and what each instantiates, ending "Each is stated in §Distinguishing prior systems at the point of use."

Proposed: append: "Each of these seven was, like the two above, individually inconsistent as an axiom — generalized over every predicate of the relevant type rather than fixed to the one prior system's own documented relation. Each is now an explicit hypothesis on the one comparison theorem that uses it, stated at the point of use in §Distinguishing prior systems, below."

§Distinguishing prior systems — ABLP, D3

Current:

None

```
axiom conj_speaks_for_left  : forall (A B : Prin), SpeaksFor A
  (Conj A B)
axiom conj_speaks_for_right : forall (A B : Prin), SpeaksFor B
  (Conj A B)

theorem D3_compound_principals_violate_single_governor
  (A B : Prin) (hAB : A <> B) :
  SpeaksFor A (Conj A B) /\ SpeaksFor B (Conj A B) /\ A <> B :=
  ...
```

Proposed:

None

```
theorem D3_compound_principals_violate_single_governor
  (hLeft  : forall (A B : Prin), SpeaksFor A (Conj A B))
  (hRight : forall (A B : Prin), SpeaksFor B (Conj A B))
  (A B : Prin) (hAB : A <> B) :
  SpeaksFor A (Conj A B) /\ SpeaksFor B (Conj A B) /\ A <> B :=
  ...
```

Append: "ABLP's compound-principal rule is taken as a pair of hypotheses on this theorem, not axioms — as axioms, each was individually inconsistent for the reason given in §Axioms."

§SPKI/SDSI

Current: no axiom shown in this subsection's listed excerpt, but the paper's full development declares `spki_threshold_empowers_left/_right` as axioms consumed by

`SPKI_threshold_violates_single_governor` (mentioned in prose, "a \$1\$-of-\$2\$ threshold subject empowers either member alone").

Proposed: append after the existing prose: "`SPKI_threshold_violates_single_governor` takes RFC 2693's threshold-subject rule as two hypotheses, not axioms, for the same reason as ABLP's compound-principal rule above."

§Macaroons

Current:

None

```
axiom copy_preserves_authorization :
  forall (P Q : Prin), Authorized P -> Copy P Q -> Authorized Q

theorem bearer_tokens_admit_multiple_governors
  (A B : Prin) (hA : Authorized A) (hCopy : Copy A B) (hAB : A <>
B) :
  Authorized A /\ Authorized B /\ A <> B := ...
```

Proposed:

None

```
theorem bearer_tokens_admit_multiple_governors
  (hCopyPreserves : forall (P Q : Prin), Authorized P -> Copy P Q
-> Authorized Q)
  (A B : Prin) (hA : Authorized A) (hCopy : Copy A B) (hAB : A <>
B) :
  Authorized A /\ Authorized B /\ A <> B := ...
```

Append: "The documented bearer property is taken as a hypothesis on this theorem, not an axiom — as an axiom it was individually inconsistent for the reason given in §Axioms."

§Biscuit

Current:

None

```
theorem biscuit_chain_integrity_matches_C_functional
  (P Q R : Prin) (h1 : SignedBlock P R) (h2 : SignedBlock Q R) :
  P = Q := ...
```

(the axiom `biscuit_signature_functional` it depends on is not shown in this excerpt but is declared elsewhere in the development)

Proposed:

None

```
theorem biscuit_chain_integrity_matches_C_functional
  (hFunctional : forall (P Q R : Prin), SignedBlock P R ->
  SignedBlock Q R -> P = Q)
  (P Q R : Prin) (h1 : SignedBlock P R) (h2 : SignedBlock Q R) :
  P = Q := ...
```

Append: "Ed25519 chain-signature functionality is taken as a hypothesis on this theorem, not an axiom, for the same reason as `CFunctional` above — the same shape as this development's own signature-functionality assumption, correctly named `CFunctional` in v4."

§UCAN

Current:

None

```
axiom ucan_revocation_dominance :
  forall (Iss Sub : Prin), Appointed' Iss Sub -> UCANRevoke Iss Sub
  -> not (R' Iss Sub)

theorem ucan_matches_P2_shape
  (Iss Sub : Prin) (hApp : Appointed' Iss Sub) (hRev : UCANRevoke
  Iss Sub) :
  not (R' Iss Sub) := ...
```

Proposed:

None

```

theorem ucan_matches_P2_shape
  (hRevDominance : forall (Iss Sub : Prin), Appointed' Iss Sub ->
UCANRevoke Iss Sub -> not (R' Iss Sub))
  (Iss Sub : Prin) (hApp : Appointed' Iss Sub) (hRev : UCANRevoke
Iss Sub) :
  not (R' Iss Sub) := ...

```

Append: "UCAN's own issuer-revocation mechanism is taken as a hypothesis on this theorem, not an axiom, for the reason given in §Axioms."

§Uniqueness — **O_unique**

(Supersedes the narrower correction of the same date.)

Current:

None

```

theorem O_unique :
  forall (P P' : Principal) (A : Agent) (X : Action) (t : Time),
    O C I M S R Appointed P  A X t ->
    O C I M S R Appointed P' A X t ->
    P = P' := by
  intro P P' A X t h0 h0'
  exact C_functional C P P' A X t h0.1 h0'.1

```

"At most one principal governs a delegated act. Proved, not assumed, from Axiom [C_functional] alone. `#print axioms O_unique` reports exactly that one dependency."

Proposed:

None

```

theorem O_unique (hC : CFunctional C) :
  forall (P P' : Principal) (A : Agent) (X : Action) (t : Time),
    O C I M S R Appointed P  A X t ->
    O C I M S R Appointed P' A X t ->
    P = P' := by

```

```
intro P P' A X t h0 h0'
exact hC P P' A X t h0.1 h0'.1
```

"If the signature predicate is functional, at most one principal governs a delegated act. The original axiom this was derived from, `C_functional`, was individually inconsistent as stated (\$Axioms); `CFunctional` is its non-axiomatic, correctly-scoped restatement, taken here as an explicit hypothesis, `hC`. `#print axioms 0_unique` reports zero dependencies. The boundary this hypothesis carves out is documented, not merely asserted: `quorum_two_governors` exhibits two distinct principals both satisfying \$O\$ on a board-quorum instance, and `quorum_not_CFunctional` proves that instance fails exactly this hypothesis."

What this does not do

It does not edit the sealed `a50-paper.tex/.pdf` or any signed instrument. It does not describe the sealed development's 23 axiom-free theorems as changed — they are not listed above because nothing about them changes. It does not describe `quorum_two_governors` or `quorum_not_CFunctional` as changed either, since neither existed in the sealed development to change — they are declared as additions above, not restatements. It does not retract the paper's mathematical content: every comparison and uniqueness result the paper describes is still true and still proved in v4, now from an explicit, non-collapsing hypothesis instead of an axiom that happened to collapse to `False`.

Signature: _____

Fitzgerald J. Heslop

Founder & CEO, Agent OBO Inc.

Date: _____