Identity-Free Economic Attestations with Reward Commitments and Private Ownership Proofs

Alvin Tanpoco
PriceChain Labs, Inc.

Revision: January 30, 2026

alvin@pricechainlabs.com

Abstract

We present a system-level primitive for identity-free economic attestations with deferred, blinded ownership claims. The primitive separates factual attestations of real-world events from incentives and identity, enabling applications to issue wallet-claimable rewards without learning or recording wallet identifiers at attestation time. An identity-free attestation (e.g., a verified spend record) is first produced and may be publicly indexed or batched without carrying user identity. Applications may then apply arbitrary reward policies to such attestations and issue reward commitment tokens that bind rewards to wallets only via cryptographic commitments with hidden openings. Ownership is proven later by the wallet holder through commitment opening or zero-knowledge proofs, without retroactively linking identity to the original attestation.

The construction relies exclusively on known privacy-preserving cryptographic primitives, including commitments, blinders, and optional zero-knowledge proofs, and introduces no new cryptographic assumptions. Its contribution is architectural: formalizing the ordering and separation of fact, incentive, and identity as a reusable system primitive. We analyze security and privacy invariants, including unlinkability between attestations and wallets, exclusivity of reward claims, and issuer-agnostic attestation consumption (i.e., attestations may be consumed by any application given a public claim-verification interface and scope-bound replay protection, without requiring the attester to learn recipient identity). We compare the primitive to prior work in anonymous credentials, privacy-preserving cryptocurrencies, and anonymous reward schemes, and show how it generalizes these patterns to real-world economic attestations. This primitive enables a class of privacy-preserving incentive systems in which applications compete over neutral attestations without reintroducing identity-based tracking.

1. Introduction

Modern measurement and incentive systems default to identity-based architectures because many outcomes are not directly observable. In domains such as advertising, content discovery, or engagement optimization, systems rely on behavioral proxies—clicks, impressions, dwell time—that require correlation across sessions and contexts to approximate causality. Persistent identifiers are therefore used as a coordination primitive for inference.

However, this dependence on identity is not universal. In outcome-attestable domains—such as verified purchases, subscriptions, or completed actions—the outcome itself can be directly attested. When a system can produce a verifiable record that a concrete event occurred, correlation over identity is no longer a technical necessity but an architectural choice.

Applying identity-first architectures to outcome-attestable domains introduces structural failures. Persistent identifiers enable cross-context correlation beyond the scope of any single application. Incentives tied directly to identity encourage long-lived tracking and profiling, as rewards must be attributed to specific users. Most critically, identity-bearing attestations are not safely reusable: downstream applications inherit the identity assumptions of the original issuer, preventing neutral reuse and composability across independent programs.

This work proceeds from the observation that, in outcome-attestable domains, identity is not required to coordinate measurement or incentives. Instead of inferring causality through identity-linked behavioral graphs, systems can evaluate outcomes directly by comparing attested events across incentive regimes. Measurement shifts from inference over identities to observation of verified facts.

This paper addresses a narrower but foundational question: **how can verified economic outcomes be attested, reused, and incentivized without making identity the coordination primitive?**

We formalize a system-level primitive that fixes the ordering between fact, incentive, and identity. Under this ordering, economic events are attested without identity; incentives are applied as external, application-defined policy; and identity enters only at claim time through blinded ownership proofs. This constraint is enforced by construction and is independent of cryptographic novelty, deployment environment, or incentive semantics.

The result is a neutral attestation layer that supports outcome measurement and incentive coordination without persistent identifiers. In such systems, causality is evaluated by comparing attested outcomes across incentive regimes, not by tracking users across contexts. Measurement shifts from inference over identity graphs to direct observation of verified facts.

The remainder of the paper formalizes this primitive, analyzes its security and privacy properties, and shows how it enables privacy-preserving incentive and measurement systems in outcome-attestable domains—without reintroducing identity-based tracking.

2. Nature of the Contribution

This paper does not introduce new cryptographic primitives. Its contribution is the formalization of an ordering constraint—**attest before identity, bind identity only through blinded commitments, defer proof to claimant**—that produces privacy and composability properties by construction.

Ordering constraints of this kind have precedent in systems research. Lamport clocks formalized “logical time before physical time.” Two-phase commit formalized “vote before commit.” Capability-based security formalized “pass the permission, not the identity.” None introduced new mathematics; each fixed a boundary that enabled new system designs.

This primitive fixes the boundary between fact, incentive, and identity. The resulting properties—unlinkability, issuer-blindness, multi-issuer composability—are not achieved through novel cryptography but through enforced sequencing of existing tools. Systems that do not enforce this ordering cannot recover these properties retroactively.

3. Related Work

This section surveys prior work in privacy-preserving systems that inform or relate to the proposed primitive.

3.1 Anonymous Credentials

Anonymous credential systems, pioneered by Chaum [1] and formalized by Camenisch and Lysyanskaya [2], allow users to obtain credentials from issuers and later prove possession without revealing identity. These systems provide strong unlinkability across credential presentations. However, credentials are inherently bound to a user-held secret at issuance time, making the credential itself an identity-rooted object. Our primitive differs by treating attestations as identity-free facts that exist independently of any user secret, with identity introduced only at the optional reward layer.

3.2 Privacy-Preserving Cryptocurrencies

Zerocash [3] and its successors (Zcash Sapling [4]) use zero-knowledge proofs to hide transaction senders, recipients, and amounts while maintaining public verifiability of state transitions. These systems achieve strong transactional privacy but tightly couple factual state changes with value transfer within a single protocol object. Our primitive explicitly separates factual attestation from value allocation, enabling multiple independent incentive layers over shared attestations.

3.3 Anonymous Token Schemes

Privacy Pass [5] and related anonymous token protocols issue unlinkable tokens that can be redeemed without revealing which issuance corresponds to which redemption. These schemes are optimized for rate limiting and access control rather than economic incentives. Notably, Privacy Pass tokens are unlinkable at redemption, but the issuance event is linkable to the origin context. Our primitive inverts this pattern: attestations are neutral at issuance (no identity bound), and identity enters only at claim time. This inversion enables durable, reusable attestations rather than single-use access tokens. Our primitive extends similar unlinkability guarantees to reward systems where attestations persist and may be consumed by multiple independent applications.

3.4 Anonymous Reward and Airdrop Systems

Anonymity mining schemes and privacy-preserving airdrops distribute rewards without revealing recipient identity, typically using Merkle proofs over committed recipient sets [6]. These systems conflate eligibility determination, issuance, and claim within application-specific constructions. Our primitive factors these concerns: attestations are neutral and reusable, reward policies are application-defined and replaceable, and ownership claims are deferred and blinded.

3.5 Attestation and Credential Systems

Direct Anonymous Attestation (DAA) [7] enables platform attestation without revealing platform identity. Privacy Pass and similar systems issue proofs of condition satisfaction. These attestations are typically single-use and consumed immediately for access control. Our primitive treats attestations as durable, reusable economic facts that may be consumed across applications and time.

References

1. D. Chaum, “Security without identification: Transaction systems to make big brother obsolete,” *Communications of the ACM*, 1985.

2. J. Camenisch and A. Lysyanskaya, “A signature scheme with efficient protocols,” *SCN*, 2002.

3. E. Ben-Sasson et al., “Zerocash: Decentralized anonymous payments from Bitcoin,” *IEEE S&P*, 2014.

4. S. Bowe et al., “Zcash Protocol Specification,” 2020.

5. A. Davidson et al., “Privacy Pass: Bypassing internet challenges anonymously,” *PoPETs*, 2018.

6. A. Gabizon, “Privacy-preserving airdrops,” 2020. (Technical report)

7. Direct Anonymous Attestation (DAA) [7] enables platform attestation without revealing platform identity.

---

4. Problem Statement

The Introduction motivates why identity-based coordination is unnecessary in outcome-attestable domains. This section formalizes the resulting requirements as explicit system constraints.

Many systems that measure or incentivize behavior implicitly adopt an identity-first architecture. Actions are collected through authenticated users, stored alongside persistent identifiers, and evaluated for rewards or analysis within the same identity-bearing context. Even when privacy-preserving cryptographic tools are employed downstream, identity is often already entangled with the primary objects of the system, making unlinkability difficult to achieve retroactively.

This coupling of fact, incentive, and identity creates several structural problems:

1. Identity-bearing records enable cross-context correlation, even when individual applications intend to use them benignly.

2. Incentives tied directly to identity encourage long-lived tracking and profiling, as rewards must be attributed to specific users.

3. Systems that embed identity at the attestation layer limit composability: downstream applications cannot safely reuse attestations without inheriting the original identity assumptions.

Existing privacy-preserving approaches partially address these issues but do not resolve them at the system level. Anonymous credentials protect authentication flows but treat credentials as identity-rooted objects. Privacy-preserving cryptocurrencies hide transaction participants but conflate factual state transitions with value transfer. Anonymous reward schemes achieve unlinkability within narrowly scoped applications but are not designed as reusable primitives for general economic attestations.

We seek a system-level primitive that decouples these concerns by construction. Specifically, we consider the problem of producing and consuming attestations of real-world economic events under the following constraints:

Constraints

Identity-Free Attestation
Attestations of events must be valid and verifiable without embedding any wallet address, user identifier, or persistent pseudonym. Identity used at the ingress or application layer for operational purposes must not be cryptographically bound into the attestation object.

Policy-Separated Incentives
Incentives or rewards must not be intrinsic to the attestation. Applications should be able to apply independent, replaceable reward policies to the same attestation without coordination.

Deferred Identity Introduction
If and when identity is introduced, it must occur only at the reward layer and only through cryptographic commitments that hide the identity by default.

Claim Exclusivity
Rewards must be claimable only by the rightful owner of the committed identity, without enabling third parties to infer or recover that identity.

Composability and Reuse
Attestations must be reusable across applications and batchable without weakening privacy guarantees or introducing implicit identity coupling.

Minimal Cryptographic Assumptions
The primitive should rely only on well-established cryptographic tools and avoid introducing new hardness assumptions.

The challenge is to satisfy all of these properties simultaneously. Systems that satisfy identity-free attestation often struggle with reward attribution, while systems that support anonymous rewards frequently assume identity-rooted objects at issuance. Our goal is to formalize a primitive that resolves this tension by fixing the ordering of operations: attest first without identity, apply incentives second, and allow identity to enter only through blinded ownership claims.

---

5. System Model and Assumptions

5.1 Entities and Vocabulary

To avoid ambiguity, we define three distinct roles that are often conflated under the term “issuer”:

- **Attester:** An entity that produces verifiable attestations of real-world events. An attester validates evidence (e.g., receipts or transaction records) and issues an identity-free Spend Token. Attesters sign sig_A on Spend Tokens.

- **Reward Issuer (Settler):** An entity that issues rewards and settles claims. In batch-based deployments, the reward issuer authenticates commitments by signing system-stream commitment events (e.g., batch commitment records), from which Reward Commitment Tokens are derived. The issuer bears economic exposure for incorrect attestations. In current deployments, the attester and reward issuer are the same entity (PriceChain Labs); the primitive does not require this.

- **Program Sponsor:** An application, brand, or organization that defines incentive policy (eligibility predicates, reward amounts, budgets, campaign windows) but does not issue rewards directly. Program sponsors specify what they are willing to pay for; the reward issuer executes on their behalf.

- **Wallet:** A cryptographic identity capable of holding keys and producing signatures or zero-knowledge proofs. Wallets generate commitment openings and prove ownership of rewards. Wallets are not required to be known to the system prior to reward issuance.

- **Ledger (Optional):** A public or shared data layer used to record attestations, reward commitments, or both. The primitive does not require a ledger but is compatible with one.

- **Ingress Identity Provider (Out of Scope):** Applications may rely on identity providers for operational purposes (rate limiting, abuse prevention). These identities are explicitly excluded from protocol-level objects.

5.2 Data Objects

- **Spend Token (Attestation Object):** A signed statement asserting that a specific economic event occurred. Spend tokens are identity-minimizing by default: portable tokens SHOULD omit wallet identifiers unless a verifier explicitly requires recipient binding (e.g., for routing or a verification service that needs wallet identity). They may include canonicalized event facts, timestamps, schema identifiers, and verification metadata. Formally defined in §5.1.

- **Reward Commitment Token (RCT):** A proof bundle indicating that a recipient's aggregated reward leaf is included under a committed batch root. An RCT includes batch metadata, the recipient identifier (recipientId, which may be a blinded commitment), a Merkle inclusion proof, and the signed commitment history that authenticates the batch root. The wallet address itself is not revealed when blinded schemas are used. Formally defined in §5.2.

- **Observed GMV Token (Optional Derived Artifact):** A signed, verifiable record of measured economic throughput over a defined window, derived from verified attestations. This is an optional portability artifact for third-party verification of aggregate throughput; it is not required by the core primitive.

5.3 Threat Model

We consider the following adversarial behaviors:

- **Curious Observers:**
  Adversaries may observe all attestations, reward commitments, and ledger state. They attempt to link attestations to wallets or infer identity through correlation.

- **Malicious Applications:**
  Applications may attempt to deanonymize users, issue malformed reward commitments, or selectively censor rewards. The system does not assume applications are honest with respect to user privacy.

- **Dishonest Claimants:**
  Wallets may attempt to claim rewards they do not own or claim the same reward multiple times.

- **Malicious Attesters:**
  Attesters may issue incorrect or fraudulent attestations. The primitive does not cryptographically guarantee the correctness of attestations. Instead, correctness is enforced indirectly through economic incentives at the application layer: incorrect attestations become costly only when an issuer commits rewards against them (via batch commitments) in anticipation of

downstream monetization; this exposure creates incentives to invest in verification proportional to reward value. This design relies on incentive alignment rather than cryptographic enforcement to bound attestation error rates.

We do **not** consider:

- Network-layer anonymity
- Side-channel leakage
- Traffic analysis

We also do not attempt to prevent inference from auxiliary data sources outside the protocol.

5.4 Trust Assumptions

The primitive makes the following minimal assumptions:

- **Cryptographic Assumptions:**
  Commitment schemes used for reward binding are assumed to be binding and hiding. Optional zero-knowledge proofs are assumed to be sound and zero-knowledge under standard assumptions.

- **Economic Correctness Assumption:**
  Applications that issue rewards are assumed to act under rational economic incentives. Because rewards represent real economic cost, issuers are incentivized to verify attestations to a degree commensurate with the value of the reward. Misverification leads to direct economic loss, providing endogenous pressure toward attestation correctness without requiring protocol-level enforcement.

- **Issuer Non-Collusion (Optional):**
  Privacy guarantees are strongest when attesters and applications do not collude with external identity providers. However, even under collusion, protocol objects do not directly reveal wallet identity.

- **Ledger Integrity (If Used):**
  If a ledger is used, it is assumed to provide integrity and availability guarantees but not confidentiality.

No trusted setup, trusted third party, or honest majority is required for the core primitive.

5.5 Design Boundary

A central design boundary of this work is the separation between **operational identity** and **cryptographic identity**.

Identity may be used transiently at the application or ingress layer but **must not** be embedded in attestation or reward objects. All privacy guarantees discussed in this paper apply strictly to protocol-level objects and flows.

5.6 Design Principles

The primitive rests on two orthogonal principles:

1. **Privacy by Ordering**
   Unlinkability between attestations and wallets arises from the fixed sequence:

   attest first without identity → commit rewards blindly → defer ownership proof

   This ordering is enforced by construction and holds regardless of application-layer behavior.

2. **Correctness by Economics**
   Attestation correctness is not cryptographically enforced. Instead, the primitive is designed such that when applications expose issuers to economic cost for rewards, verification effort self-aligns with reward value. The protocol enables this; applications choose whether to leverage it.

3. **Neutrality Enables Competition**
   Because attestations carry no intrinsic incentive and no embedded identity, multiple independent programs may apply reward policies to the same attestation without coordination. This is structurally difficult in systems where credentials are identity-rooted or where factual state changes are coupled with value transfer.

Neither principle subsumes the other. Privacy by ordering would hold even for worthless attestations; correctness by economics would work even in identity-bearing systems. The primitive’s contribution is their composition into a single reusable abstraction.

5.7 Compatibility Boundary

Many incentive and measurement systems require identity persistence to attribute outcomes. Cross-context identifiers—whether deterministic or probabilistic—serve as the coordination mechanism through which actions are linked to users, users are linked to rewards, and program effectiveness is evaluated.

This primitive allows outcome attribution **without identity persistence**, by binding incentives to attestations rather than users. Attribution is performed over identity-free attestations and program-scoped reward commitments, not over cross-context user graphs.

As a result, outcome measurement and incentive delivery can be implemented without identity persistence, while still supporting exclusivity of claims and issuer-agnostic reward issuance.

Systems that depend on cross-context identifiers for attribution or measurement are structurally incompatible with the invariants defined in Section 5 in outcome-attestable domains where these invariants are enforced. This incompatibility is by design: the primitive trades identity-based coordination for attestation-based coordination, enabling a class of applications that cannot be built on identity-first architectures.

Identity persistence is increasingly regulated and operationally costly. Systems that can compute outcomes without identity reduce compliance surface area and operational fragility. When identity persistence becomes unavailable or unreliable, systems that depend on it fail discretely; systems built on attestation-based coordination continue to function without modification.

5.8 Operational and Deployment Boundaries

This paper defines protocol objects and their ordering. Execution infrastructure, verification policy, and reward economics are explicitly out of scope.

- **Current Deployment:**
  In current deployments, PriceChain Labs operates both the attester and reward issuer roles. This centralization reflects operational realities—receipt

parsing, fraud detection, merchant normalization, and dispute handling require coordinated infrastructure and economic accountability that are difficult to decentralize prematurely.

- **No Claim of Current Decentralization:**
  This paper does not claim that verification is decentralized today. It claims that the primitive does not require centralized verification, and that the properties described hold regardless of how verification authority is distributed.

- **Centralization Is an Execution Choice, Not a Design Dependency:**
  The Spend Token and Reward Commitment abstractions intentionally separate who verifies facts, who applies incentive policy, and who claims rewards. A Spend Token is valid if and only if its attester signature is trusted by the consuming program. Whether that signature comes from a single operator, a federation, or competing attesters does not alter the ordering, privacy, or composability properties of the system.

- **Decentralization Path:**
  The primitive supports multiple independent attesters issuing Spend Tokens under distinct keys, program-level allowlists of trusted attesters, and competition between verifiers. Decentralizing verification would change who issues Spend Tokens and how trust is established; it would not change the ordering of fact → incentive → identity, the identity-free nature of attestations, or the reward commitment mechanism.

Summary:
Verification is centralized today for operational correctness, but the primitive is verifier-agnostic by design. Decentralization is a deployment choice, not a prerequisite for the properties described here.

---

6. Primitive Definition

This section formally defines the system-level primitive for identity-free attestations with deferred, blinded ownership claims. The primitive consists of two object types and three operations: **attestation**, **reward commitment issuance**, and **ownership proof**. No assumptions are made about deployment environment, ledger choice, or reward semantics.

6.1 Spend Token (Identity-Free Attestation)

A Spend Token is a verifiable attestation that a specific economic event occurred. It is the primary factual object of the system.

Definition 1 (Spend Token).
A Spend Token STSTST is a tuple:

```
ST = (spendId, schemaId, canonicalData, attestationMeta, sig_A)
```

Where:

- `spendId` is a unique identifier for the attested event
- `schemaId` identifies the attestation schema
- `canonicalData` encodes normalized event facts, which may include:
  - Merchant or activity classifications
  - Transaction amount buckets
  - Temporal buckets (e.g., coarse time windows)
  - Coarse geographic indicators
- `attestationMeta` includes verification tier, evidence references, or other non-identifying metadata
- `sig_A` is a signature by an attester over the above fields

Invariant ST-1 (Identity-Minimizing by Default):
Spend Tokens SHOULD omit wallet addresses and other stable identifiers in portable contexts. A wallet MAY be included only when explicitly required for recipient binding (e.g., reward routing, custodial verification services, or operator-internal audit flows). Operational identity used during ingestion (e.g., authentication or rate limiting) MUST NOT be cryptographically bound into the Spend Token unless the deployment explicitly opts in to recipient-bound issuance.

Invariant ST-2 (Neutrality):
A Spend Token does not encode incentives, rewards, or economic obligations. It represents a claim about an event, not a right to value.

6.2 Reward Commitment Token (Batch-Based, Blinded Recipient)

A Reward Commitment Token (RCT) records that a recipient’s aggregated reward leaf is included under a committed batch root, without revealing the wallet when blinded schemas are used.

Definition 2 (Reward Commitment Token).
A Reward Commitment Token RCTRCTRCT is a tuple:

```
RCT = (chainId, batch, recipientId, leaf, proof, systemEvents,
rewardInclusionProof?)
```

Where:

- chainId identifies the commitment ledger or anchoring context
- batch = (batchId, root, schemaVersion, committedAt, txRef) is the committed batch metadata
- recipientId is a recipient reference: either a transparent WalletRef or a blinded per-batch commitment
- leaf is the recipient’s aggregated reward leaf (e.g., total points for this batch; optionally includes rewardEventsRoot in linkable schemas)
- proof is a Merkle inclusion proof from leaf to batch.root
- systemEvents is a signed, prevHash-linked system-stream segment authenticating the commitment event
- rewardInclusionProof (optional) links a specific spendId to rewardEventsRoot when using a linkable leaf schema

RCTs are proof bundles: authenticity derives from the signed system-stream commitment history (and optional on-chain anchoring), not from a token-level signature.

Invariant RCT-1 (Wallet Hiding):
Under blinded schemas, recipientId is a per-batch commitment. The wallet WWW MUST NOT be recoverable from recipientId without knowledge of the opening (W,batchId,blinder)(W, batchId, blinder)(W,batchId,blinder).

Invariant RCT-2 (Claim Exclusivity):
Only an entity possessing a valid opening of the blinded recipientId can prove that the leaf corresponds to their wallet and claim the reward.

Invariant RCT-3 (Scope Binding):
Recipient commitments are batch-scoped. Cross-program correlation SHOULD be avoided by isolating programs into distinct batches or encoding program scope into batchId. Blinders MUST be unique per (wallet,batchId)(wallet, batchId)(wallet,batchId) to prevent linkability across batches.

6.3 Deployment Profiles (Blinder Generation)

The privacy properties of Reward Commitment Tokens depend on who generates the per-batch blinder. Protocol deployments MUST specify which profile is used.

Profile A (Issuer-Blind):
Wallets generate the blinder locally and transmit only the derived recipientId (or (wallet, batchId, blinder)) to the issuer/batcher before the batch is committed. The issuer cannot recover the wallet from recipientId alone.

This is the **normative default** for privacy-preserving deployments and is **REQUIRED** to achieve issuer blindness (Property P-1 in §7.4).

Profile B (Issuer-Visible):
Issuers generate the blinder or compute recipientId from (wallet, batchId). External observers cannot link wallets to attestations, but the issuer can deanonymize.

This profile is acceptable only when issuer trust is explicitly assumed.

The invariants in this section hold under both profiles; however, the strongest privacy guarantees require Profile A.

6.4 Issuance and Commitment (Attestation → Batch → RCT)

Reward issuance is an application-layer decision and is not intrinsic to the Spend Token. Rewards are committed in batches, and RCTs are derived from those batch commitments.

Definition 3 (Reward Issuance and Commitment).

Let:

- Policy be an application-defined function
- Eligible(Policy, ST) → {true, false}

- `Reward(Policy, ST) → rewardAmount`

Then the flow is:

```
IssueReward(Policy, ST, W) → RewardEvent
CommitBatch(RewardEvents, RecipientRegistrations) → CommitmentBatch
DeriveRCT(CommitmentBatch, recipientId) → RCT
```

Such that:

- `Eligible(Policy, ST) = true` before a reward event is emitted
- `RecipientRegistrations` supply a per-batch `recipientId` (blinded or transparent) for each recipient
- A batch commitment is published and authenticated via the system stream (and optionally on-chain)
- If `Eligible(Policy, ST) = false`, no reward event is issued and no batch inclusion occurs

Invariant IR-1 (Policy Separation):
Reward eligibility and amount are determined solely by application policy and not by the Spend Token itself.

Invariant IR-2 (Issuer Exposure):
Reward issuance incurs economic cost to the issuer, incentivizing verification effort and bounding incorrect issuance.

Invariant IR-3 (Batch Commitment):
Once a batch is committed, inclusion proofs are publicly verifiable and non-repudiable.

6.5 Batching Semantics

Rewards are committed in batches with one leaf per recipient per batch (aggregation preserved).

Definition 4 (Batch Commitment).
A batch groups reward events and commits a Merkle root over recipient leaves.

Linkable schemas MAY also commit a per-recipient `rewardEventsRoot` to enable compact spend ↔ reward proofs.

Batching does not alter privacy or identity guarantees.

Invariant B-1 (Non-Amplification):
Batching does not introduce new identity linkage beyond what exists for individual recipients.

Invariant B-2 (Freshness):
Blinders MUST be unique per `(wallet, batchId)` to prevent cross-batch correlation.

6.6 Ownership Proof (Claim)

Ownership of a reward is proven after batch commitment.

Definition 5 (Ownership Proof).
A claimant proves ownership by demonstrating knowledge of `(W, batchId, blinder)` such that:

```
Commit(W, batchId, blinder) = RCT.recipientId
```

This may be done via:

- Explicit opening, or
- Zero-knowledge proof of commitment opening

Invariant OP-1 (Deferred Identity):
Wallet identity is introduced only at proof time and only by the claimant.

Invariant OP-2 (No Identity Propagation):
The Spend Token remains identity-free and does not become identity-bearing as a result of reward commitment or claim. Any wallet disclosure at claim time is voluntary and program-scoped and does not modify or contaminate the attestation object.

6.7 Primitive Summary

The primitive enforces the ordering described in §1.1:

1. Economic event
2. Identity-free attestation (ST)
3. Application-defined incentive
4. Batch-committed reward (RCT)
5. Deferred ownership proof

This ordering is fixed by construction and is independent of application domain, attester identity, or ledger implementation.

6.8 Protocol Boundary Interface (Inclusion Verification and Redemption)

For the primitive to support issuer-agnostic verification, a minimal interface is required to validate batch inclusion proofs without depending on the original issuer.

Definition 6 (Inclusion Verification Interface).

```
VerifyInclusion(RCT) → {accept, reject}
```

Where:

- `RCT` is the Reward Commitment Token being verified
- `systemEvents` authenticate the batch commitment
- `proof` proves inclusion of `leaf` under `batch.root`

Verification Logic:

1. Verify `systemEvents` integrity and authority validity for the commitment event
2. Verify `proof` against `batch.root`

3. If recipientId is blinded and the claimant provides an opening, recompute recipientId and compare

4. If all checks pass: return accept; else return reject

Redemption (optional, out-of-protocol):
Applications MAY add a claim/anti-replay layer (e.g., a scope-bound nullifier set) to prevent double-claims. This redemption layer is separate from batch commitment verification and does not change the commitment proof semantics.

---

7. Security and Privacy Properties

This section analyzes the security and privacy properties of the primitive under the system model and assumptions defined in Section 5. We focus on properties that arise from construction and ordering rather than from new cryptographic assumptions.

7.1 Unlinkability Between Attestations and Wallets

Property U-1 (Attestation–Wallet Unlinkability).
Given a Spend Token ST and a Reward Commitment Token RCT, an observer cannot determine whether a specific wallet W is associated with ST except through a valid ownership proof produced by W.

Argument.
Spend Tokens are identity-minimizing by default (Invariant ST-1). Reward Commitment Tokens bind rewards to recipients via a per-batch recipientId commitment (Invariant RCT-1). Without knowledge of the opening (W, batchId, blinder), the commitment reveals no information about W. RCTs are batch proofs and are not inherently linked to specific spends; optional spend ↔ reward linkage proofs reveal a spendId but not the wallet. Observers therefore cannot infer wallet association through protocol objects alone in the default case.

7.2 Deferred Identity and Voluntary Disclosure

Property U-2 (Deferred Identity Introduction).
Wallet identity is introduced into the system only at the time of ownership proof and only by the claimant.

Argument.
The primitive does not require wallets to be known at attestation time or reward issuance time. Ownership proofs are initiated by wallets and may be implemented via explicit commitment opening or zero-knowledge proofs (Invariant OP-1). Prior to this step, no protocol object reveals wallet identity, preventing involuntary or premature disclosure.

7.3 Claim Exclusivity and Double-Claim Resistance

Property S-1 (Claim Exclusivity).
Only an entity possessing a valid opening of the blinded `recipientId` can claim the associated reward.

Argument.
By the binding property of the commitment scheme, no two distinct wallets can open the same per-batch commitment. An ownership proof requires knowledge of `(W, batchId, blinder)` such that `Commit(W, batchId, blinder)` matches the `recipientId` in the RCT.

Property S-2 (Double-Claim Resistance).
A reward cannot be claimed more than once without detection.

Argument.
Double-claim prevention can be enforced at the redemption layer by associating a nullifier or claim identifier with each RCT at claim time. Once a claim is accepted, the nullifier is recorded, preventing replay. This mechanism is orthogonal to identity hiding and does not require revealing `W`.

7.4 Issuer Blindness

Property P-1 (Issuer Blindness).
Under Profile A (wallet-generated blinders), reward issuers cannot learn the wallet identity associated with a reward commitment. Under Profile B (issuer-generated blinders), third-party observer blindness holds, but the issuer can deanonymize.

Argument.
Under Profile A, wallets generate the blinder locally and transmit only the derived `recipientId`. The hiding property of the commitment scheme ensures the issuer cannot recover `W` without knowledge of the opening. Under Profile B, the issuer computes `recipientId` and can deanonymize, but external observers still cannot link wallets to attestations.

7.5 Resistance to Correlation and Batch Privacy

Property P-2 (Non-Amplification Under Batching).
Batch commitments do not amplify identity leakage beyond what exists for individual recipients.

Argument.
Each batch has one leaf per recipient, and blinded schemas use a per-batch recipientId (Invariant B-2). Observers learn only that a recipient has some aggregated rewards in a batch, not the underlying wallet. Optional spend ↔ reward linkage proofs (when used) reveal spendId membership but still do not reveal wallet identity. Batching therefore does not introduce additional correlatable identity material beyond what the deployment already exposes.

7.6 Incentive-Aligned Attestation Correctness

Property E-1 (Economic Correctness Enforcement).
The primitive encourages attestation correctness through economic exposure at the reward layer rather than cryptographic guarantees.

Argument.
Reward issuance incurs real cost to the issuer. Issuing rewards against incorrect or fraudulent attestations results in direct economic loss. Rational issuers are therefore incentivized to invest in verification proportional to reward value. This mechanism parallels incentive-based correctness assumptions in oracle systems and does not require protocol-level enforcement of real-world truth.

7.7 Adversarial Limitations

The primitive does not protect against:

- Side-channel leakage at the network or application layer
- Collusion between issuers and external identity providers
- Inference from auxiliary data sources outside protocol objects

These limitations are inherent to the problem domain and are addressed in Section 9.

7.8 Summary of Properties

The primitive achieves:

- **Unlinkability between attestations and wallets** — Property U-1
- **Deferred and voluntary identity disclosure** — Property U-2
- **Exclusive and non-replayable reward claims** — Properties S-1, S-2
- **Issuer blindness under Profile A; third-party observer blindness under both profiles** — Property P-1
- **Privacy preservation under batching** — Property P-2
- **Incentive-aligned correctness** — Property E-1

These properties arise from the ordering constraint defined in §1.1 and the invariants in §6.

8. Comparative Analysis

This section compares the proposed primitive against prior systems that address privacy, incentives, or attestations. The goal is not to claim uniqueness of individual techniques, but to clarify differences in **ordering**, **separation of concerns**, and **reuse**.

8.1 Comparison Dimensions

We compare systems along the following axes:

- **Attestation Identity:** Whether identity is embedded in the primary object
- **Incentive Binding:** Whether rewards are intrinsic or external
- **Wallet Introduction:** When and how wallet identity enters
- **Reuse Across Applications:** Whether attestations can be reused independently
- **Correctness Model:** Cryptographic vs. economic enforcement

8.2 Anonymous Credential Systems

Anonymous credential systems issue credentials bound to a user-held secret at issuance time and support later selective disclosure. While they provide strong unlinkability across presentations, the credential itself is identity-rooted.

Difference.
In the proposed primitive, the primary object (Spend Token) is not a credential and is not bound to any user secret. Identity may exist at the application ingress layer but is explicitly excluded from the attestation object. Rewards are optional and external, whereas credentials are intrinsic rights.

Implication.
Anonymous credentials are well-suited for authentication and authorization but do not generalize cleanly to neutral, reusable economic attestations consumed by multiple applications.

8.3 Privacy-Preserving Cryptocurrencies

Privacy-focused cryptocurrencies conceal transaction participants and amounts during value transfer. These systems tightly couple factual state transitions and economic value within a single protocol object.

Difference.
The proposed primitive separates factual attestation from value transfer entirely. Spend Tokens record facts without transferring value, while Reward Commitment Tokens allocate value without revealing identity. This separation enables multiple independent incentives to be layered on the same attestation.

Implication.
Privacy coins protect transactional privacy but do not address multi-issuer incentives over shared factual records.

8.4 Anonymous Reward and Incentive Schemes

Systems such as privacy-preserving airdrops and anonymity mining schemes distribute rewards without revealing recipient identity. These systems often rely on commitments, Merkle proofs, or shielded accounting.

Difference.
Such schemes are typically application-specific and conflate eligibility determination, reward issuance, and claim within a single construction. In contrast, the proposed primitive explicitly decouples attestation from incentives and treats reward issuance as a replaceable policy decision.

Implication.
The primitive generalizes anonymous reward techniques into a reusable abstraction that supports competing incentives and heterogeneous policies.

8.5 Attestation Systems

Attestation systems such as Direct Anonymous Attestation and Privacy Pass issue proofs that an entity satisfied certain conditions, often without revealing identity.

Difference.
Attestations in these systems are typically single-purpose and consumed immediately for access control. The proposed primitive treats attestations as durable, reusable objects that may be consumed by multiple applications at different times.

Implication.
This durability enables economic reuse but introduces new incentive and privacy considerations addressed by the primitive’s ordering.

8.6 Summary Table

System	Attestation Identity	Incentive Binding	Wallet Introduction	Reuse Across Apps	Correctness Model
Anonymous Credentials	Yes (hidden)	Yes	At issuance	Limited	Cryptographic
Privacy Coins	Hidden	Yes	During transaction	No	Cryptographic
Anonymous Airdrops	No	Yes	At claim	No	Cryptographic

Attestation Systems	No	No	Immediate	No	Cryptographic
This Primitive	No	No	Deferred, blinded	Yes	Economic

8.7 Positioning

The proposed primitive occupies a distinct design point: identity-free primary attestations, externalized incentives as program policy, support for multiple independent program sponsors over shared attestations, and economic rather than cryptographic correctness enforcement. These properties make it suitable as a substrate for privacy-preserving incentive systems in outcome-attestable domains.

8.8 What This Replaces

In outcome-attestable domains, systems built on persistent identifiers treat identity as the coordination layer for measurement and incentives. This primitive replaces identity-based coordination with attestation-based coordination, enabling equivalent or superior outcome attribution without cross-context linkage.

9. Applications

This section outlines representative applications enabled by the proposed primitive. These examples are illustrative rather than exhaustive and are intended to demonstrate how the separation of attestation, incentive, and identity enables system designs that are difficult to realize in identity-first architectures.

9.1 Privacy-Preserving Incentives for Real-World Economic Activity

The most direct application is the issuance of incentives for verified real-world economic events without requiring identity disclosure. Applications may observe identity-free attestations of spend, apply reward policies, and issue rewards that are claimable only through blinded, per-batch recipient commitments.

This enables incentive systems in which:

- Users are rewarded for verified behavior without persistent tracking
- Multiple programs can independently define incentives over the same attestation
- Rewards do not require custodial accounts or identity-linked balances

Such systems generalize loyalty programs while avoiding centralized user profiles and long-lived identifiers.

9.2 Multi-Program Incentives Over a Single Attestation Layer

This work supports multiple independent incentive programs operating over the same set of verified attestations.

Deployment note.
In current deployments, PriceChain Labs is the sole attester and reward issuer. Program sponsors define incentive policies (eligibility predicates, budgets, reward schedules); PriceChain executes verification, reward issuance, and settlement on their behalf. Programs do not share user identifiers, wallets, or reward state. Programs interact only with identity-free Spend Tokens and program-scoped outcome reports.

This mirrors established infrastructure patterns (e.g., payment processors executing for many merchants) without exposing identity.

9.3 Predicate-Based Incentive Eligibility

Programs may define eligibility predicates over Spend Token fields:

```
Predicate(ST) → {eligible, ineligible}
```

Examples:

- `merchantCategory == "athletic_footwear" AND brand != "nike"`
- `transactionAmount > $100 AND daysSinceLast > 60`

- `COUNT(category == "organic_grocery") ≥ 3`

Users may prove satisfaction of predicates via zero-knowledge proofs without revealing which specific attestations satisfy the condition. Predicate satisfaction proofs are unlinkable across interactions.

This enables incentive allocation based on verified behavior rather than inferred identity, without creating persistent cross-context identifiers.

9.4 Worked Example: Outcome Measurement Without Identity

The point of this example is not cashback. It is that **interest** → **engagement** → **conversion** → **redemption** can be measured using verifiable artifacts without a persistent user identifier.

Consider a brand promotion:

> *"Purchase any product from Category X, receive 500 points, upload receipt for verification."*

Step 1: Eligibility Check (Interest / Intent)

A user queries whether their wallet satisfies the promotion predicate (e.g., geographic eligibility, wallet age, prior activity threshold). This check is performed via a zero-knowledge proof.

- The brand observes that an eligibility check occurred
- No identity is revealed

This is **verifiable interest**—an intentful protocol interaction, not a passive impression.

Step 2: Promo Engagement

The user opens the promotion details and reviews the offer terms.

- This interaction is logged against a blinded session identifier

This is **verifiable engagement**.

Step 3: Purchase and Attestation (Conversion)

The user purchases a qualifying product and uploads the receipt.

- The receipt is verified
- A Spend Token is issued—identity-free, containing only canonical spend facts

This is **verified conversion**.

Step 4: Reward Claim (Redemption)

The user claims the reward by proving inclusion in a committed batch (opening their blinded recipient commitment if required).

- The Reward Commitment Token (proof bundle) is issued or retrieved

The brand knows:

- A conversion occurred
- A reward was claimed

The brand does **not** know who.

What the Brand Observes

- Count of eligibility checks (interest / intent)
- Count of promo opens (engagement)
- Count of verified Spend Tokens matching the predicate (conversion)
- Count of Reward Commitment Tokens claimed (redemption)

Outcome measurement and redemption accounting occur **without persistent identifiers**. Each metric is derived from a cryptographically verifiable artifact.

> **Note:** “Interest” here refers to protocol-level interactions (eligibility queries), not passive impressions. This primitive does not measure attention; it measures outcomes and outcome-proximate interactions.

The brand can run A/B tests by varying reward amounts across promotion instances and comparing conversion differentials—**regime comparison over attested facts**, not inference over tracked users.

9.5 A Note on Interpretation

The example above is easy to underestimate.

A reader may see “receipt upload” and conclude this paper describes a privacy-preserving loyalty system—a useful but narrow application. That interpretation misses the structural claim.

Consider what the primitive actually provides:

Metric	Identity-Based Systems	This Primitive
Impression tracking	Identity	Not measured
Interest / intent signal	Identity	Eligibility ZK proof
Click tracking	Identity	Blinded session
Conversion attribution	Identity	Spend Token
Outcome measurement	Persistent identity graph	Outcome measurement only

Outcome-proximate metrics are available **without identity at any step**.

Passive impressions are explicitly out of scope. This primitive measures **intentful interactions and verified outcomes**, not attention.

This is not loyalty infrastructure with a privacy feature.
It is **outcome measurement where identity never enters the system.**

If you reached Section 9 and concluded the opportunity is “privacy-preserving cashback,” return to Section 1 and ask what problem is actually being solved.

The paper does not claim to replace inference-based attribution. It describes a primitive that makes identity-based inference unnecessary for outcome measurement in domains where outcomes are attestable. The distinction is left to the reader.

Scope constraint:
This primitive applies to outcome-attestable domains (receipts, invoices, subscriptions, verified events). It does not apply to pure attention metrics (impressions, views) unless a verifiable outcome proxy is defined. The “replace clicks” implication is surgical, not universal.

9.6 Measurement Without Identity-Based Tracking

The primitive enables measurement and aggregation of economic activity without identity-based analytics.

Attestations can be indexed, batched, and analyzed statistically while remaining unlinkable to individuals.

Applications include:

- Aggregate spend measurement across a shared pool of neutral attestations
- Program effectiveness analysis using only post-fact attestations and program-scoped reward commitments
- Market-level insights derived from attestations rather than users
- Causal program evaluation without cross-program user linkage

Rewards can be issued to encourage participation or disclosure without collapsing measurement into tracking.

Downstream consumers value verifiable, reusable economic attestations because they reduce reliance on probabilistic inference and identity persistence. Because attestations are machine-verifiable and reusable, they may serve directly as inputs to automated decision systems that allocate resources, budgets, or incentives without requiring user-level tracking.

9.7 Public-Good and Collective Incentive Systems

Public-good funding mechanisms often struggle to balance accountability and privacy.

Using identity-free attestations, contributions or qualifying actions can be recorded without identifying participants. Applications can then issue rewards or funding allocations through blinded commitments.

Examples include:

- Subsidies for qualifying economic behavior
- Retroactive public-good rewards
- Grant allocation mechanisms that avoid identity disclosure

Economic exposure at the reward layer incentivizes issuers to enforce attestation quality without requiring centralized identity registries.

9.8 Cross-Application Composability

Because attestations are neutral and reusable, they can serve as shared inputs across applications.

A single attestation format may be consumed by:

- Wallets
- Analytics platforms
- Incentive programs
- Research systems

without requiring shared identity assumptions.

This composability supports:

- Interoperable incentive systems

- Modular application architectures
- Reduced duplication of verification effort

9.9 Non-Financial Rewards and Credentials

Although described primarily in economic terms, Reward Commitment Tokens need not represent monetary value.

Applications may issue:

- Access rights
- Credentials
- Non-transferable acknowledgments

using the same blinded commitment mechanism.

This allows:

- Privacy-preserving access control
- Proof-of-participation without identity linkage
- Deferred disclosure of entitlement

9.10 Summary

The applications above share a common enabler: the ordering constraint from §1.1 allows incentive and measurement systems to preserve privacy by construction while remaining economically rational and composable.

---

10. Limitations and Open Questions

While the proposed primitive enables a clean separation of attestation, incentive, and identity, it does not solve all problems associated with privacy-preserving economic systems.

10.1 Attestation Quality and Verification Costs

The primitive does not cryptographically guarantee attestation correctness. Instead, it relies on economic exposure at the reward layer.

When an issuer commits rewards via batch commitments in anticipation of downstream monetization, incorrect attestations represent direct financial liability.

This creates a natural quality gradient:

- High-value rewards → stronger verification incentives
- Low-value rewards → weaker verification incentives
- Monetization risk → verification investment floor

Issuers operating under this model function analogously to insurance underwriters: they assess evidence, price risk, and absorb losses from verification errors. This parallels oracle systems in decentralized finance, where economic exposure substitutes for cryptographic proof of real-world state.

10.2 Conditions for Economic Correctness

The “correctness by economics” property (§5.6) is not automatic. It requires the following conditions to hold:

1. **Issuer bears downside risk:**
   The issuer must have economic exposure to verification errors—through bonds, reserves, clawback mechanisms, or chargeback liability. Issuers that are insolvent, judgment-proof, or able to externalize losses do not face the incentives assumed by this model.

2. **Rewards are not paid before verification finality:**
   If rewards are disbursed before attestation correctness is established, issuers can profit from issuing bad attestations and exiting. Dispute windows or deferred settlement are required to align timing.

3. **Repeated-game incentives or contractual penalties exist:**
   One-shot interactions do not induce reputation costs. Issuers must operate in contexts where future revenue depends on current correctness or where explicit contractual penalties apply.

4. **Adversarial collusion is bounded:**
   If issuers collude with claimants to fabricate attestations and split rewards, economic correctness breaks down. Mitigations include audit rights, selective verification by sponsors, and slashing mechanisms.

These conditions are **deployment requirements**, not protocol guarantees. The primitive enables economic correctness; applications must structure incentives to realize it.

10.3 Sybil Resistance and Participation Constraints

The primitive does not address Sybil resistance or participant uniqueness.

Identity-free attestations may be produced at scale by adversarial actors unless constrained by external mechanisms such as:

- Rate limits
- Economic costs
- Attester-side controls

This limitation is deliberate. Sybil resistance is orthogonal to identity separation and may be layered independently. Designing incentive-compatible Sybil mitigation that does not reintroduce identity remains an open area of research.

10.4 Inference from Auxiliary Data

Although protocol objects are identity-free, adversaries may infer associations through auxiliary data such as:

- Timing correlations
- Network metadata

- Off-protocol information

The primitive does not attempt to mitigate these inference channels.

Addressing such leakage may require complementary techniques—network-layer anonymity, batching delays, or application-level noise injection—each introducing trade-offs outside the scope of this work.

10.5 Collusion and Trust Boundaries

The strongest privacy guarantees assume that reward issuers do not collude with ingress-layer identity providers or external data brokers.

While the primitive prevents direct cryptographic linkage, collusion can reintroduce correlation through side channels. Exploring governance, auditing, or cryptographic enforcement mechanisms that limit collusion without requiring trusted intermediaries remains an open challenge.

10.6 Reward Semantics and Transferability

This work treats reward semantics abstractly.

Whether rewards are:

- Transferable
- Non-transferable
- Fungible
- Expiring

is left to application policy.

Different choices may affect privacy guarantees, economic behavior, and regulatory interpretation. Formalizing reward semantics and understanding their interaction with privacy and incentives is an area for future study.

10.7 Standardization and Interoperability

While the primitive is designed to be reusable, practical interoperability requires shared schemas, commitment formats, and claim interfaces.

Premature standardization may limit flexibility, while fragmentation may hinder adoption. Determining appropriate abstraction boundaries for standardization is an open systems question.

10.8 Summary

These limitations reflect deliberate design choices rather than oversights. The primitive aims to establish a minimal, composable foundation for privacy-preserving incentives, leaving orthogonal concerns to be addressed by higher-level systems or future research.

---

11. Conclusion

This paper formalized a system-level primitive for identity-free economic attestations with deferred, blinded ownership claims.

The contribution is the ordering constraint described in §1.1:

> **attest before identity → bind identity only through blinded commitments → defer proof to claimant**

This ordering produces privacy and composability properties by construction.

We analyzed how unlinkability, issuer blindness, and claim exclusivity arise from this ordering rather than from novel cryptography, and how attestation correctness is incentivized economically rather than enforced cryptographically.

Systems that currently require persistent identifiers to attribute outcomes—and the compliance, operational, and privacy costs that accompany them—may find this primitive offers a structurally superior alternative in outcome-attestable domains.

Appendix A: Concrete Instantiation

This appendix provides a concrete instantiation of the primitive to demonstrate implementability. The choices below are illustrative; other valid instantiations exist.

A.1 Commitment Scheme

Choice: Domain-separated SHA-256 hash commitment (per-batch recipientId).

```
Commit(W, batchId, blinder) =
  SHA-256("crinkl.recipient.v1:" || W || ":" || batchId || ":" || blinder)
```

Where:

- W is the wallet address
  - EVM: lowercase hex with 0x prefix
  - Solana: base58
- batchId identifies the commitment batch
- blinder is a 32-byte random value

Output: 64-character lowercase hex (recipientId)

Encoding requirements:

- All string components MUST be UTF-8 encoded
- Concatenation uses : as delimiter (ASCII 0x3A)
- The blinder MUST be generated from a cryptographically secure random source

Properties:

- **Binding:** SHA-256 collision resistance prevents distinct tuples from colliding
- **Hiding:** 256-bit blinder provides computational hiding

- **Scope separation:** `batchId` scopes commitments per batch or program

Alternative: Pedersen commitments provide information-theoretic hiding but require group operations.

A.2 Spend Token Structure

```
{
  "spendId": "spend_01HXYZ...",
  "schemaId": "crinkl.spend.v1",
  "canonicalData": {
    "storeId": "store_ABCD1234",
    "totalCents": 4250,
    "timestampBucket": "2025-01-15"
  },
  "attestationMeta": {
    "verificationTier": "HARD_VERIFIED",
    "evidenceRef": "sha256:9f86d08..."
  },
  "signatures": {
    "attester": "ed25519:BASE64_SIG",
    "attesterPubKey": "ed25519:BASE64_PUBKEY"
  }
}
```

Encoding: RFC 8785 (JSON Canonicalization Scheme) for deterministic hashing.

A.3 Reward Commitment Token Structure

```
{
  "tokenType": "REWARD_COMMITMENT",
  "schemaVersion": 1,
  "chainId": "solana-devnet",
  "batch": {
    "batchId": "batch_01HDEF...",
    "root": "b3f1...64hex",
    "schemaVersion": "2b",
    "txRef": "slot:12345:tx:6ab...",
    "committedAt": "2025-01-15T14:30:00Z"
  },
  "recipientId": "a3b8c9d0e1f2...64hex",
  "leaf": {
    "recipientId": "a3b8c9d0e1f2...64hex",
    "totalPoints": "425",
    "batchId": "batch_01HDEF...",
    "rewardEventsRoot": "9c7e...64hex"
  },
  "proof": {
    "leafHash": "abc123...64hex",
    "siblings": ["def456...64hex", "..."],
    "leafIndex": 7
  },
  "systemEvents": [
    /* signed, prevHash-linked commitment history */
  ],
  "rewardInclusionProof": {
    "batchId": "batch_01HDEF...",
    "recipientId": "a3b8c9d0e1f2...64hex",
    "rewardEventsRoot": "9c7e...64hex",
    "leaf": {
      "spendId": "spend_01HXYZ...",
      "rewardEventHash": "f1e2...64hex"
    },
    "leafHash": "aa55...64hex",
    "siblings": ["bb66...64hex", "..."]
  }
}
```

A.4 Batching and Merkle Commitment

- Binary Merkle tree with SHA-256
- Leaf hash: `SHA-256(0x00 || canonicalize(leaf))`
- Internal hash: `SHA-256(0x01 || min(L,R) || max(L,R))`
- Leaves sorted by `recipientId` (lexicographic UTF-8)
- Padded to power of 2 with empty leaf hash

On-chain anchor:
Merkle root published to Solana (or other L1) with batch metadata.

A.5 Ownership Proof

Explicit opening:

```
{
  "proofType": "opening",
  "wallet": "7xKXtg2C...",
  "batchId": "batch_01HDEF...",
  "blinder": "BASE64_32_BYTES",
  "merkleProof": {
    "leafHash": "abc123...",
    "siblings": ["def456...", "..."]
  }
}
```

Verifier recomputes `Commit(wallet, batchId, blinder)` and checks it equals the `recipientId`, then verifies the Merkle proof.

Zero-knowledge alternative:
SNARK proof demonstrating knowledge of `(W, batchId, blinder)` such that:

```
Commit(W, batchId, blinder) == recipientId
```

Suitable circuits may be constructed using Groth16 or PLONK over BN254 or BLS12-381.

A.6 Ledger Integration

Solana example:

- Batch commitment stored in a PDA (Program Derived Address)
- Fields: `batchId`, `merkleRoot`, `leafCount`, `totalPoints`, `schemaVersion`, `committedAt`

- Authority signature verified on-chain
- Transaction hash serves as immutable `txRef`

Ethereum alternative:

- Emit `BatchCommitted(batchId, root, leafCount, totalPoints)` event
- Event log serves as commitment anchor