ZK Snip: Addressless Private Payments

Yield-Bearing Privacy and Selective Compliance

0x8E1X4P2O9S3E7D

Architecture Paper

Version 1.1 February 18, 2026

Abstract

ZK Snip is a privacy payment protocol designed to prevent transaction graphs from forming at all, while supporting selective compliance without surveillance. It eliminates persistent wallet addresses, suppresses entry–exit correlation, and economically reinforces uncertainty on both sides of the transaction lifecycle.

Instead of address-based transfers, ZK Snip uses short-lived, human-shareable claim codes ("snips"). Because snips are ephemeral and redeemed via zero-knowledge proofs, payments avoid address reuse, clipboard risk, and wallet exposure by construction.

A sender deposits funds into a shielded pool and distributes a snip off-chain. The recipient redeems the snip by producing a zero-knowledge proof of inclusion and non-spend (via a nullifier), while a decentralized set of privacy amplifiers injects indistinguishable cover traffic at both deposit and claim time. External observers see $k + 1$ identical events, and no stable entry–exit linkage can be established.

Unlike opt-in privacy systems whose anonymity sets stagnate or leak at their boundaries, ZK Snip couples privacy with incentive alignment. Protocol fees fund active cover traffic, increasing the veil multiplier k as usage grows. Privacy compounds with participation and becomes yield-bearing infrastructure rather than a fixed-cost feature.

Selective, proof-based compliance is supported as a first-class primitive and enforced without surveillance or trusted intermediaries. Transactions may optionally carry zero-knowledge attestations that predefined rules are satisfied—such as membership in an allowed set or non-membership in a deny list—without revealing identity, transaction history, or counterparties. Compliance state is encoded via cryptographic accumulators whose updates are proven correct and executed by a decentralized operator set shared with privacy amplification. Enforcement occurs via transaction-level pre-checks, preserving default privacy while enabling institutional participation without reintroducing transaction graph visibility. These mechanisms are enforced by a live, deployed execution environment that atomically verifies amplification correctness, liveness, and amplifier settlement.

ZK Snip is not a mixer: it suppresses correlation at graph formation rather than relying on delayed withdrawal or address reuse.

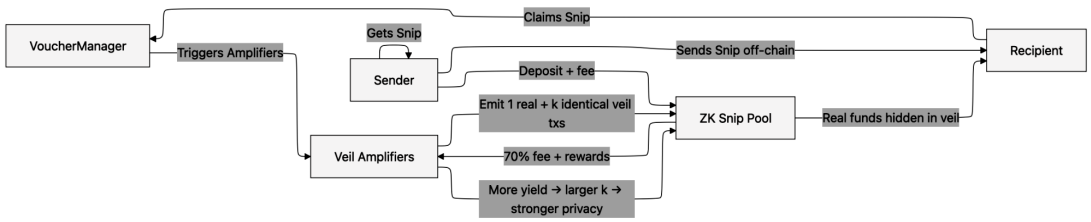


Figure 1: High-level system overview illustrating how protocol fees are transformed into privacy through veil amplification.

Contents

Executive Summary

ZK Snip is a privacy payment protocol designed around three mutually reinforcing structural moats. Each moat addresses a fundamental failure mode in existing on-chain payment systems. Together, they define the protocol's security model, economic resilience, and adoption surface.

1. Addresslessness (Mutual Address Blindness). ZK Snip eliminates persistent wallet addresses entirely. Value is transferred via short-lived claim codes ("snips") rather than identity-bearing endpoints. This enforces *mutual address blindness* by default: the sender never learns the recipient's address, the recipient never learns the sender's address, and no reusable identifier is exposed on-chain. Because no durable address exists, no transaction graph can accumulate over time. This removes the primary substrate on which blockchain surveillance operates and enables payment flows—payroll, treasury settlement, gifts, and institutional transfers—that are irrational on address-based rails.

2. Privacy as Yield-Bearing Infrastructure. Privacy in ZK Snip is not static or opt-in. Protocol fees are converted into active cover traffic via privacy amplification, increasing the veil multiplier k as usage grows. Higher transaction volume increases rewards for Privacy Amplifiers, which supports larger anonymity sets and tighter unlinkability bounds for all participants. Privacy compounds with participation rather than degrading under load, transforming anonymity from a fixed feature into economically reinforced infrastructure.

3. Selective Compliance Without Surveillance. ZK Snip separates privacy from compliance by enabling optional, proof-based enforcement of rules without identity disclosure. Compliance enforcement is decentralized and cryptographically enforced by the same operator set that provides privacy amplification, eliminating trusted intermediaries and single points of control. Transactions may carry zero-knowledge attestations that predefined conditions are satisfied—such as membership in an allowed set or non-membership in a deny list—without revealing identities, transaction histories, or counterparties. Rules are enforced via transaction-level pre-checks rather than continuous monitoring, enabling institutional participation without reverting to transparent rails.

Bidirectional Anonymity (Emergent Property). Beyond these core moats, ZK Snip achieves *bidirectional anonymity*. Uncertainty is expanded at both transaction entry and exit points: deposits are obscured through protocol-managed entropy, while claims are protected via veil amplification. An adversary must therefore succeed in correlating both deposit and withdrawal events, imposing a multiplicative cost on global linkage attacks. This property emerges naturally from the protocol's architecture and strengthens as usage increases.

Decentralized Compliance Execution. Selective compliance in ZK Snip is enforced by a decentralized execution layer operated by the same staked nodes responsible for privacy amplification. Compliance state is encoded via cryptographic accumulators whose updates are governed by deterministic rules and proven correct via zero-knowledge proofs.

Operators do not exercise discretion over compliance outcomes. They execute fixed verification logic, submit proofs of correctness on-chain, and serve inclusion or exclusion proofs to

users. Any operator can independently reproduce accumulator state from authoritative inputs and prove correctness without coordination.

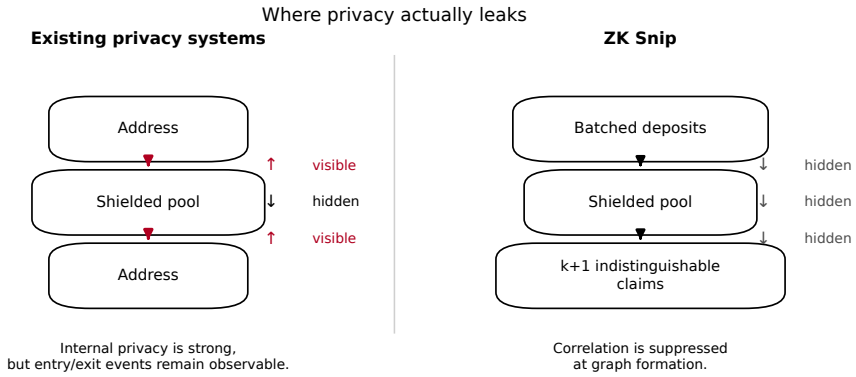
This architecture eliminates trusted compliance operators, discretionary enforcement, and centralized update authority, while preserving liveness through redundant execution.

Summary. ZK Snip removes persistent identity surfaces, converts privacy into yield-bearing infrastructure, supports selective compliance without surveillance,and compounds anonymity in both directions of the transaction lifecycle. The rational action—using the system—strengthens privacy for everyone.

1 Introduction

Public blockchains create a permanent, linkable record of economic behavior. While this property simplifies verification, it renders many real-world payment flows irrational: payroll, merchant payments, treasury operations, and bilateral settlements all expose long-lived identifiers whose transaction graphs only become easier to analyze over time.

1.1 Boundary Leakage in Existing Privacy Systems



A shared limitation across most contemporary privacy payment systems is the persistent visibility of *entry and exit points* to the anonymity set. While internal transaction details may be cryptographically concealed, the act of entering or leaving a shielded domain typically remains observable.

Deposits into privacy pools produce visible inflows from transparent contexts, and withdrawals or claims generate observable outflows. These boundary events expose timing, frequency, and value information that can be exploited by adversaries using correlation attacks, statistical disclosure, or behavioral heuristics. Over long observation windows, such edge leakage enables partial or full reconstruction of transaction graphs, even when the internal pool semantics are cryptographically sound.

This limitation is architectural rather than cryptographic. It arises from reliance on persistent addresses, explicit shielding and unshielding steps, or fixed deposit–withdrawal patterns. As a result, effective anonymity is bounded not by internal pool size, but by the distinguishability of ingress and egress events.

Boundary leakage is a primary reason why many real-world payment flows—such as payroll, treasury settlement, and institutional transfers—remain impractical on existing privacy rails. Users are exposed not within the privacy set, but at its edges.

This boundary leakage motivates the design choices described in the following sections, particularly ZK Snip's addressless transfer model and bidirectional anonymity guarantees.

ZK Snip begins from a different premise:

> *Addresses are not a feature. They are an attack surface.*

Practical corollary: faster than addresses. In address-based systems, the payment surface is the identifier itself. Users must obtain it, verify it, paste it, and permanently reveal it. ZK

Snip collapses this ceremony into a short-lived claim code that expires after redemption. This is not merely more private—it is a simpler interaction model, and the reason addressless payments are both safer and operationally faster.

Rather than masking transactions between reusable addresses, ZK Snip removes the identity surface entirely. Users exchange short-lived claim codes off-chain and prove redemption on-chain without exposing sender–recipient linkage.

A central property of ZK Snip is *mutual address blindness*. Neither sender nor recipient ever reveals a persistent on-chain identifier to the other, and no reusable address is exposed at any point in the transaction lifecycle. The sender does not learn the recipient's address; the recipient does not learn the sender's address; external observers learn neither. Crucially, ZK Snip is designed to eliminate correlation at both boundaries of the privacy set. Deposits are incorporated without durable linkage to future claims, and redemptions are accompanied by active cover traffic. The resulting anonymity set is therefore *bidirectional*: uncertainty is expanded at both entry and exit points, and an adversary must succeed in correlating both to reconstruct transaction flows. This property is structural rather than heuristic, and it suppresses transaction graph formation at its origin.

The second premise is economic. Privacy is an emergent property that survives only when reinforced by incentives. ZK Snip introduces *veil amplification*, converting protocol fees into active cover traffic. Privacy therefore compounds with participation and becomes yield-bearing infrastructure.

2 System Overview

ZK Snip consists of:

- **Shielded Pool**: Holds user funds inside a privacy set.
- **Voucher Manager**: Tracks voucher commitments, Merkle roots, nullifiers, and expiry.
- **Payload Availability Layer (PAL)**: A modular, off-chain availability layer responsible for serving encrypted, unlinkable payloads required for snip redemption. Payloads are content-addressed, client-encrypted, and reveal no identity, linkage, or transaction semantics. Compromise or censorship of PAL instances can affect liveness but does not degrade privacy or unlinkability.
- **Veil Amplifiers**: Decentralized actors emitting indistinguishable cover transactions.
- **Selective Assurance Layer (SAL)**: Decentralized predicate enforcement using ZK-verified accumulator roots. SAL execution and accumulator correctness are provided by a redundant operator set that may overlap with Veil Amplifiers, using cryptographic proofs rather than trust.

The protocol is designed so observers learn only voucher hashes and observe $k + 1$ identical events per claim (one real, k veil).

2.1 Payload Availability Layer (PAL)

ZK Snip requires an off-chain mechanism for making encrypted redemption payloads available to recipients. This function is provided by the *Payload Availability Layer (PAL)*, a deliberately

minimal and modular component.

PAL instances store only client-encrypted, content-addressed payloads. They do not possess decryption keys, observe sender or recipient identities, or learn any on-chain linkage. Payload identifiers are derived from snip secrets and are computationally unlinkable to deposits or claims.

The PAL is *not a trusted component*. Failure, censorship, or compromise of any PAL instance impacts only availability (liveness), never privacy. An adversary controlling the PAL cannot distinguish real users, correlate deposits to claims, or reconstruct transaction graphs.

The PAL abstraction may be instantiated using:

- replicated stateless relay services,
- decentralized object storage (e.g., content-addressed networks),
- or client-mediated distribution mechanisms.

The choice of PAL backend does not affect the protocol's anonymity guarantees. This modularity ensures that ZK Snip remains robust against infrastructure-level surveillance or takedown attempts.

2.2 Bidirectional Anonymity (System-Level Property)

ZK Snip enforces *bidirectional anonymity* at the protocol level. Unlinkability is guaranteed in both directions of the transaction lifecycle: observers cannot reliably associate a claim with its originating deposit, nor infer which deposit eventually resulted in a given claim.

This property emerges from two complementary mechanisms embedded directly into the system design.

Deposit-Side Ambiguity. Deposits into the shielded pool are incorporated via batched state transitions and standardized commitment formats. Multiple user deposits may be aggregated into a single observable update, and protocol-managed decoy commitments may be injected to further expand the entry-side anonymity set. Deposit amounts are snapped to predefined buckets shared with the claim phase, eliminating value-based fingerprinting across the deposit–claim boundary.

Claim-Side Amplification. During redemption, each real claim is accompanied by k computationally indistinguishable veil claims selected via verifiable randomness. All $k+1$ claims are validated in an aggregated proof without revealing execution order, timing, or origin. External observers therefore observe a symmetric set of events with no reliable distinguisher.

Resulting Effect. By expanding uncertainty at both entry and exit points, ZK Snip suppresses transaction graph formation even under long-horizon observation. Any successful linkage attack must simultaneously resolve ambiguity in the deposit set and the amplified claim set, imposing a multiplicative cost on correlation attempts.

3 Cryptographic Preliminaries

ZK Snip relies on standard primitives:

- **Commitments and Merkle Trees**: deposits create commitments accumulated into a tree; the root is stored on-chain.
- **Nullifiers**: each redemption computes a unique nullifier to prevent double-spend without linking to the deposit.
- **ZK-Friendly Hashing**: Poseidon2 (or equivalent) for efficient hashing in circuits and identifier derivation.
- **Zero-Knowledge Proofs**: claims prove inclusion, validity constraints (expiry, amount bucket), and non-spend.
- **Post-Quantum Orientation**: STARK-style proving where practical (hash-based security posture).
- **Randomness**: VRF or unbiased beacon for veil amplifier selection.

3.1 Post-Quantum Security Model

ZK Snip is designed under conservative post-quantum assumptions. All cryptographic security derives from hash-based primitives and transparent proof systems.

In particular, the protocol relies on:

- collision-resistant hash functions for commitments, nullifiers, and Merkle accumulators,
- STARK-style zero-knowledge proofs for correctness, inclusion, and unspentness,
- and algebraic hash functions (e.g., Poseidon variants) optimized for proof systems.

No trusted setup, pairing-based cryptography, or discrete logarithm assumptions are required. As a result, the protocol remains secure against adversaries equipped with both classical and quantum computational capabilities.

This positions ZK Snip to remain viable under emerging post-quantum cryptographic standards while preserving transparency, upgradability, and auditability. **Explicit post-quantum posture.** Because ZK Snip's correctness and privacy arguments rely on transparent proof systems (STARK-style) and hash-based primitives (commitments, Merkle accumulators, nullifiers), the protocol avoids pairing curves, trusted setup artifacts, and discrete-log security dependencies. This makes the privacy core *post-quantum oriented* by construction: security degrades with hash strength, not with elliptic-curve assumptions.

4 Protocol Specification

4.1 Actors and State

- **Sender**: deposits funds and generates a snip code.
- **Recipient**: redeems a snip by recovering payload P and producing a ZK proof.
- **Shielded Pool**: holds assets; processes deposits and claim transfers.
- **Voucher Manager**: maintains voucher roots, nullifier set, and expiry logic.
- **PAL (Payload Availability Layer**: stores ciphertext indexed by id; cannot decrypt without code-derived key.
- **Veil Amplifiers**: emit indistinguishable veil transactions; compensated from fees.

4.2 Deposit / CreateVoucher

The sender performs:

1. Generate secret s, payload P, and voucher hash H_v.
2. Deposit funds into the Shielded Pool and pay the settlement fee.
3. Submit `CreateVoucher(`H_v`, proof)` to the Voucher Manager.
4. Derive $id = \text{Poseidon2}(H_v)$.
5. Derive $key = \text{KDF}(\texttt{code})$ and compute $cipher = \text{Enc}(key, P)$.
6. Store $(id, cipher)$ with the Payload Availability Layer and send the snip code off-chain to the recipient.

On-chain observers learn only H_v and standard metadata.

4.3 Claim / Redemption with Veil Amplification

The recipient:

1. Derives *key* from the snip code, requests and decrypts *cipher* to recover P.
2. Computes nullifier $N = \text{Poseidon2}(s \| context)$.
3. Builds a ZK proof attesting voucher inclusion under the latest root, unspent status (nullifier unused), expiry, and bucket constraints.
4. Calls `ClaimVoucher(`H_v`, `N`, proof, recipient_addr)`.

Upon successful verification:

- The voucher is marked spent (nullifier inserted).
- Funds are released to `recipient_addr`.
- k veil amplifiers are selected via VRF and emit k indistinguishable veil transactions.

4.4 Why Addressless Payments Matter (Practical Impact)

Permanent wallet addresses were an expedient abstraction in early on-chain systems, but they introduce long-lived identity surfaces that are incompatible with real economic use.

Once an address is shared, it becomes a durable anchor for behavioral inference: balances, counterparties, timing patterns, and strategic intent accumulate over time. Mitigations such as address rotation reduce but do not eliminate this exposure.

ZK Snip removes this identity surface entirely. Value is transferred via short-lived claim codes that:

- do not persist on-chain as reusable identifiers,
- are unlinkable across transactions,
- and expire once claimed.

From a user perspective, this results in a simpler and safer interaction model: sending value no longer requires revealing a permanent identifier, and receiving value does not expose prior activity.

This property enables on-chain payments in contexts where address-based transfers are currently avoided, including gifts, payroll, treasury operations, and institutional settlements.

4.5 Use Cases (Why This Gets Used)

ZK Snip is designed for contexts where the rational actor avoids reusable addresses. The core product primitive is simple: *share a short-lived claim code, not an identity surface.* This unlocks payment flows that are currently avoided on public chains:

- **Gifts and peer transfers.** Send value like sending a link. No address exchange, no "paste the wrong wallet," no durable graph formation. A snip expires once claimed.
- **Payroll and contractor payouts.** Employers should not learn employees' on-chain histories; employees should not have to reveal a permanent address to get paid. Snips enforce *mutual address blindness*: neither side learns a reusable on-chain identifier.
- **Treasury operations and bilateral settlement.** Treasury moves are strategic signals. Address-based rails turn strategy into a public dataset. Snips allow settlement without exposing a long-lived endpoint.
- **Institutional distribution and compliance zones.** For high-stakes flows, integrators can require *optional proof-based assurance* (Section 7) while keeping default privacy intact. This enables "private by default, provable when required."
- **Activists, journalists, and high-risk recipients.** Receiving funds via an address is often equivalent to publishing an identity anchor. Snips minimize exposure: the recipient receives without revealing a durable identifier, and observers cannot build a persistent relationship graph.

These are not niche edge cases; they are the set of payments that do *not* happen today because sharing an address is itself the risk.

5 Sequence Diagrams (Protocol Graphs)

5.1 Step 0 — Private Deposit (Sender Shields Funds)

Step 0: Private Deposit (Shielding Funds) The sender shields funds in the pool to enable private voucher creation.mermaid

```
sequenceDiagram
    title Step 0 — Private Deposit (Sender Shields Funds)
    participant S as Sender Wallet
    participant P as Shielded Pool
    participant O as Observer

    S->>P: deposit(amount) via relayer
    P-->>S: issue shielded notes (STARK-proofed)
    Note over O: Sees deposit tx, but not note ownership (shielded)
```

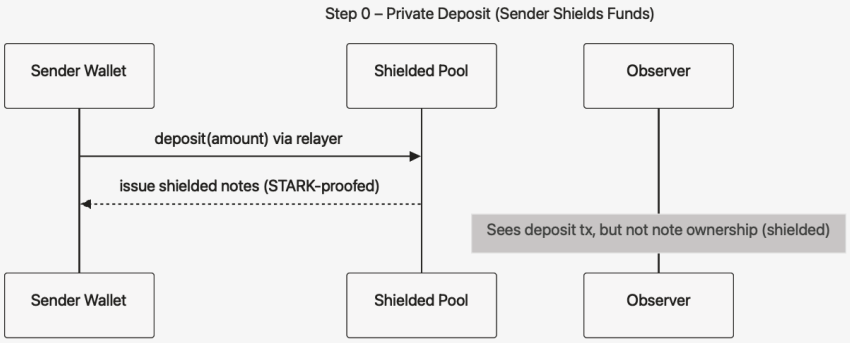


Figure 1: Step 0 — Private Deposit: sender shields funds into the pool.

5.2 End-to-End: Deposit → Code Distribution → Claim → Amplification

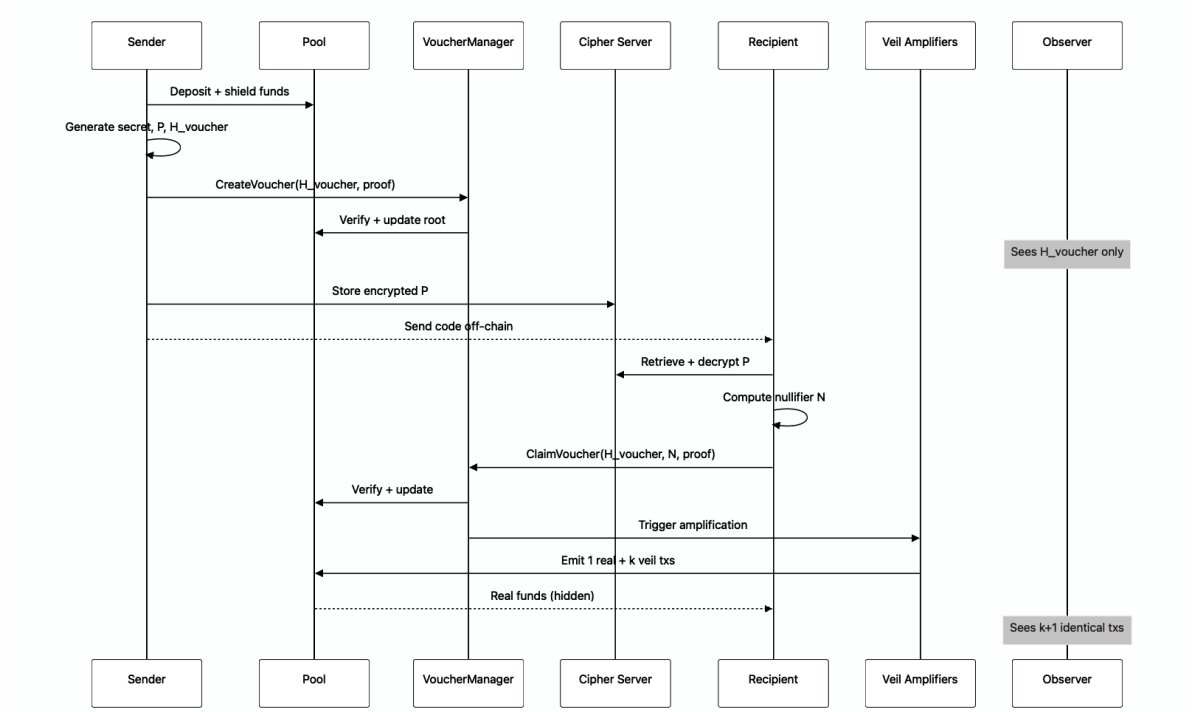


Figure 2: End-to-end flow: deposit, voucher creation, off-chain code distribution, claim, and veil amplification.

5.3 Step 2 — Code & PAL Distribution (Off-Chain)

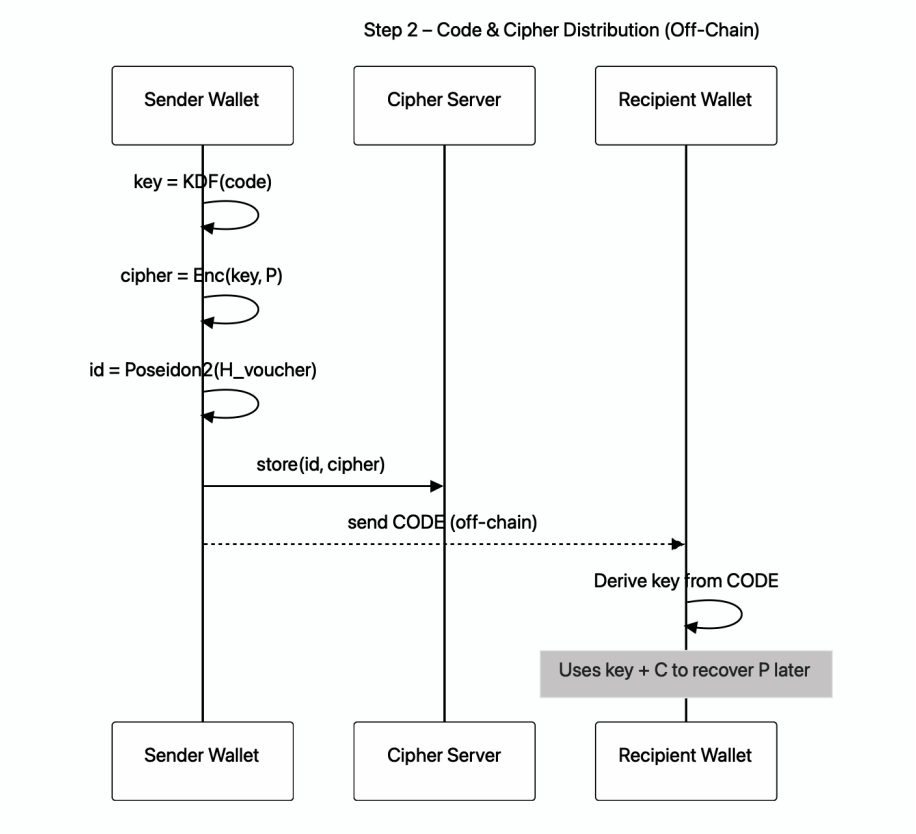


Figure 3: Step 2 — Off-chain code & PAL distribution: sender stores encrypted payload; recipient derives key from the snip code.

5.4 Step 3 — Claim (With Veil Amplification)

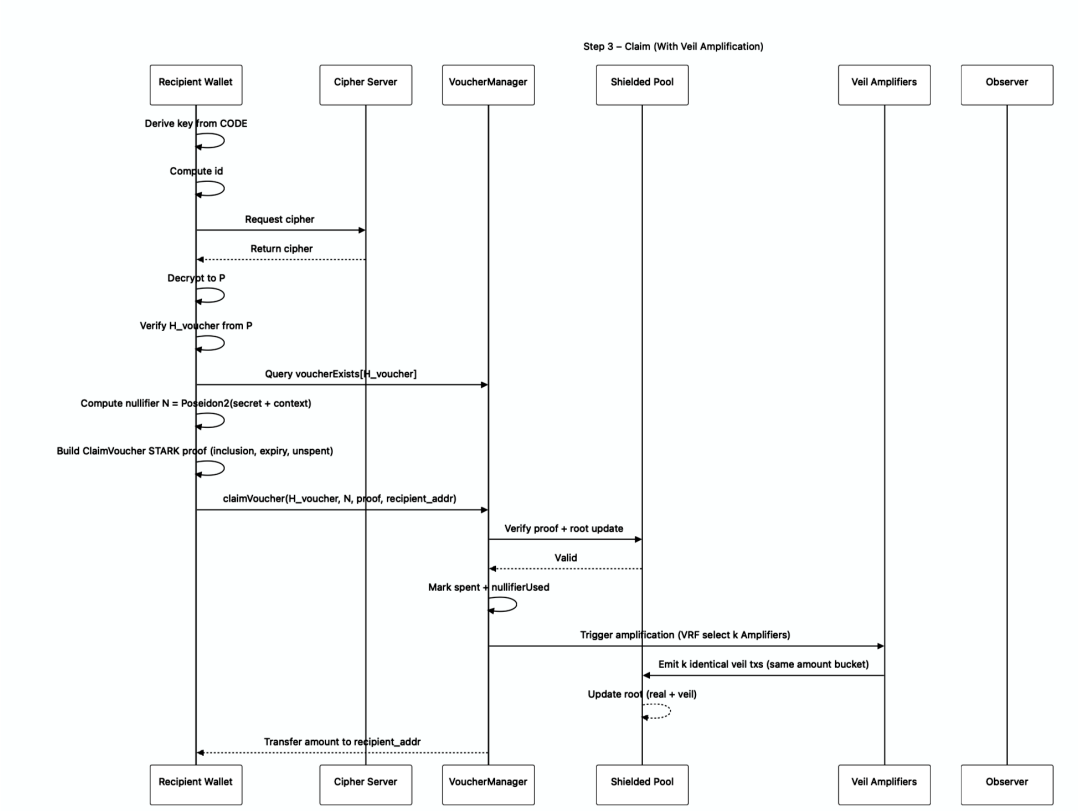


Figure 4: Step 3 — Claim with veil amplification: recipient proves validity; k veil transactions are emitted to suppress linkage.

6 Veil Amplifiers: Privacy as Yield-Bearing Infrastructure

Veil amplifiers convert protocol fees into privacy by providing cover traffic that is computationally indistinguishable from real claims. Veil amplification introduces a structural shift in how privacy is sustained. Rather than treating anonymity as a fixed-cost feature or a user opt-in, the protocol makes privacy responsive to economic activity.

As usage increases, protocol fees fund additional cover traffic, increasing the effective anonymity set for all users. Privacy therefore strengthens with participation, rather than degrading over time. In this sense, privacy becomes yield-bearing: economic activity directly finances the uncertainty that protects it.

6.1 Deposit-Side Entropy Expansion

In addition to claim-time veil amplification, ZK Snip optionally introduces *deposit-side entropy* to further suppress correlation between fund entry and redemption events. Rather than emitting explicit decoy deposits per user action, the protocol expands uncertainty at the pool level through protocol-managed ambiguity mechanisms.

Deposits are accepted continuously but committed to the shielded pool in batched and partially randomized intervals. Multiple user deposits may be incorporated into a single observable state transition, preventing external observers from reliably associating a specific deposit transaction with a subsequent voucher commitment. To further widen the entry-side anonymity set, the protocol may inject protocol-owned decoy commitments that are computationally indistinguishable from user deposits.

Low-volume protection (minimum effective batch). Critically, protocol-injected decoy commitments are not merely an optimization they are a *bootstrap safety mechanism*. Even when organic deposit flow is sparse, the protocol can maintain a minimum effective batch size by funding additional indistinguishable commitments at the pool level. This prevents the entry-side anonymity set from collapsing to a single observed deposit and reduces the usefulness of "100 in, 100 out" heuristics that rely on low-liquidity periods. These decoys are treated as a protocol-level privacy budget item, not an optional enhancement, and may be sustained through treasury reserves during low-activity regimes.

Crucially, deposit-side entropy is *protocol-funded* rather than user-funded. The cost of maintaining deposit ambiguity is amortized across protocol fees and treasury-controlled reserves, preserving a frictionless user experience while treating privacy as a shared public good.

Deposit amounts are snapped to discrete buckets aligned with claim-side bucketing, eliminating value-based fingerprinting across the deposit–claim boundary. As a result, observers cannot reliably infer whether a given commitment corresponds to a user deposit, a batched aggregation, or a protocol-injected decoy.

By expanding uncertainty at both entry and exit points, ZK Snip grows anonymity sets bidirectionally. Deposit-side entropy compounds with claim-time veil amplification, increasing the cost of global correlation attacks without introducing additional trust assumptions or user complexity. In effect, ZK Snip does not treat deposits as "bare" entry events: ingress ambiguity is a protocol budget item, maintained even when organic flow is temporarily thin.

6.2 Incentives and Economics

ZK Snip targets a flat **0.20% settlement fee** on transferred value (governance-configurable). A configurable fraction α of fee revenue is distributed to veil amplifiers. Higher volume increases amplifier rewards; larger rewards support a higher veil multiplier k; higher k strengthens privacy for all participants.

APR intuition (volume-driven, not speculative). Amplifier returns are funded by protocol activity (fees) and optional emissions, not by rehypothecation. For a fixed veil cost-per-event and stable participation, amplifier APR is approximately proportional to settlement volume and the fee share α; the protocol can therefore target yield bands (e.g., 15–40% in high-activity regimes) while treating k as a governance-controlled privacy budget rather than an abstract parameter.

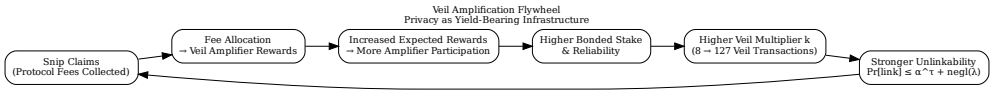


Figure X: Veil amplification flywheel. Protocol fees fund veil amplifiers, increasing the veil multiplier k, which tightens the α^{τ} unlinkability bound and reinforces privacy through participation.

6.3 Coprocessor-Enforced Settlement

Execution correctness and economic settlement are enforced by a dedicated privacy amplification coprocessor.

For each amplified claim, the coprocessor verifies: (i) correct veil amplifier selection, (ii) timely emission of all required veil transactions, and (iii) cryptographic validity of the aggregated amplification proof.

Fee allocation and amplifier rewards are computed deterministically as part of the same execution flow. Only amplifiers whose veil contributions are verified are eligible for compensation. If amplification is incomplete, delayed, or malformed, the claim is rejected and no rewards are paid.

By coupling veil execution and reward settlement atomically, ZK Snip eliminates reliance on off-chain accounting, discretionary payouts, or trusted coordinators. Privacy amplification and economic compensation are enforced at the same layer.

Implementation Status and Enforcement Guarantees. The privacy amplification logic described above is not a theoretical construct. ZK Snip's execution environment for veil amplification is implemented today as a production-grade coprocessor that is live, deployed, and actively enforcing protocol rules.

A reference implementation of SAL execution using Herodotus Sentinel is live and operational, and can be made available to reviewers under controlled disclosure.

This coprocessor deterministically verifies veil participation, aggregates amplification proofs, and enforces amplifier compensation and slashing conditions as part of settlement. Veil trans-

actions are validated atomically with the real claim, and amplifier rewards are computed and distributed only if correctness and liveness conditions are satisfied. As a result, privacy amplification is not a best-effort overlay or off-chain coordination mechanism, but a *hard execution constraint* of the protocol.

All guarantees in the following sections therefore rely on an operational system, not on assumed future components.

6.4 Selection, Fairness, and Liveness

Amplifier selection must be unbiased and verifiable. We implement VRF-based selection weighted by bonded stake and reliability scores (uptime, latency, correctness), with liveness enforcement and slashing for non-performance. Transaction templates are standardized to minimize distinguishers (gas patterns, calldata shape, timing). These mechanisms ensure unbiased participation, but the privacy guarantee of veil amplification follows from a stronger, probabilistic unlinkability argument formalized below.

6.5 Latency, Synchronization, and User Experience

Veil amplification requires coordinating the emission of one real claim transaction alongside k indistinguishable veil transactions. To suppress timing-based correlation, these transactions must be broadcast within a narrow execution window.

Bounded coordination window (UX envelope). Let Δ denote the protocol-defined coordination window for a claim at veil tier k. Amplifiers must broadcast their veil transactions within Δ of the real claim broadcast. In practice, Δ is chosen to be short (e.g., on the order of tens of seconds) to prevent timing separation from becoming a distinguisher while keeping user-perceived latency predictable.

Enforcement and failover. To prevent UX degradation from amplifier underperformance, ZK Snip treats liveness as a first-class correctness condition. Selected amplifiers that miss the window are slashable, and repeated misses reduce future selection probability. If insufficient veils arrive within Δ, the protocol can (i) re-sample replacement amplifiers, and/or (ii) temporarily top up missing veils using protocol-operated capacity (treasury-funded veils) to maintain the target anonymity tier. This ensures that latency remains bounded while preserving the conditions required for timing unlinkability.

This introduces a bounded latency tradeoff. Claim finality is gated on amplifier participation rather than immediate inclusion, and user-perceived latency increases with higher veil tiers k.

ZK Snip addresses this through three mechanisms.

First, amplifier participation is economically enforced. Amplifiers are selected via stake-weighted VRFs and are subject to strict slashing for missed or delayed execution. Failure to broadcast within the synchronization window results in partial or full stake penalties, making non-participation economically irrational.

Second, veil tiers are user-selectable. Users may choose lower tiers for latency-sensitive payments or higher tiers for high-value transfers where privacy guarantees dominate UX concerns. This exposes the tradeoff transparently rather than hiding it behind protocol defaults.

Third, synchronization windows are short and deterministic. Amplifier execution occurs within bounded delays (e.g., tens of seconds), preventing unbounded waiting while eliminating exploitable timing variance.

As a result, latency is predictable, economically enforced, and tunable, rather than adversarially exploitable.

6.6 Formal Unlinkability Guarantee (Veil Amplification)

Veil amplification provides a probabilistic, post-quantum unlinkability guarantee by emitting one real claim transaction alongside $k = \tau$ computationally indistinguishable veil transactions. Observers cannot distinguish the real claim from cover traffic without controlling the entire veil set.

We formalize this property below.

6.7 Formal Unlinkability Guarantee (Bidirectional Veil Amplification)

ZK Snip provides a probabilistic, post-quantum unlinkability guarantee by expanding uncertainty at *both* transaction entry and exit points. Rather than relying solely on claim-side cover traffic, the protocol compounds anonymity through *bidirectional entropy*: deposit-side ambiguity and claim-time veil amplification.

At claim time, one real withdrawal is emitted alongside $k = \tau$ computationally indistinguishable veil transactions. At deposit time, user commitments are obscured via batched pool updates and protocol-injected decoy commitments. An external observer must therefore succeed in *both* deposit identification and claim attribution to deanonymize a payment.

We formalize this compounded property below.

6.8 Deposit-Side Correlation, Protocol-Injected Decoys, and Low-Volume Regimes

While claim-side unlinkability is actively reinforced through veil amplification, deposit-side privacy is constrained by observable inflows. This limitation is shared by most privacy payment systems: deposits originate from transparent contexts and therefore expose timing and coarse value information at entry.

ZK Snip mitigates ingress correlation through a combination of (i) batching, (ii) amount bucketing, and (iii) *protocol-injected decoy notes*. Deposits are mapped into discrete buckets and aggregated over time windows to reduce the precision of amount- and timing-based heuristics. In addition, the protocol may inject decoy deposits (cover notes) that are indistinguishable from user deposits at the pool level, increasing ambiguity even in low-volume regimes.

As a result, observers cannot reliably distinguish whether an observed inflow corresponds to a real user deposit or protocol-generated cover traffic, nor can they determine which specific inflow is relevant to a later claim.

Residual leakage is bounded and actively correctable. In extremely low-liquidity regimes, an observer may attempt bucket-level statistical inference (e.g., correlating a rare inflow in a bucket with a later claim in the same bucket). ZK Snip treats this as a *managed regime*,

not a steady state: batching parameters, decoy ratios, and (when enabled) treasury-funded cover notes are adjusted to preserve a minimum entry-side ambiguity. In other words, entry-side uncertainty is designed to be *maintained* rather than passively hoped for, and it improves monotonically with participation.

Theorem 1 (Bidirectional Unlinkability Against Majority Adversary). Let:

- τ denote the user-selected veil tier, where $k = \tau \in \{8, 16, 32, 64, 127\}$,
- $\alpha \in [0, 1]$ denote the fraction of total bonded privacy-amplifier stake controlled by a probabilistic polynomial-time (PPT) adversary $\mathcal{A}$,
- $\beta \in [0, 1]$ denote the adversary's best achievable advantage in identifying the *originating deposit batch element* (i.e., distinguishing the true user deposit from deposit-side entropy).

ZK Snip satisfies *bidirectional τ-unlinkability*: for any PPT adversary $\mathcal{A}$ with $\alpha \leq \frac{2}{3}$, the probability that $\mathcal{A}$ correctly links a real claim to its originating deposit is bounded by

$$\Pr[\text{de-anonymisation}] \leq \beta \cdot \alpha^{\tau} + (\lambda),$$

where $\lambda = 256$ is the security parameter and (λ) is a negligible function.

Bounding Deposit-Side Advantage. Let deposits be committed to the shielded pool in batches of total size $B = U + D$, where each batch contains U user deposits and D protocol-injected decoy commitments, with all commitments computationally indistinguishable at the transaction and calldata level. Assume deposits are snapped to discrete value buckets aligned with claim-side bucketing.

Under these conditions, the adversary's optimal success probability of selecting the originating user deposit from the batch is at most $1/(U + D)$, and thus

$$\beta \leq \frac{1}{U + D}.$$

For example, with conservative parameters $U = 8$ and $D = 8$, we obtain $\beta \leq \frac{1}{16}$. Larger batch sizes or higher protocol-funded decoy ratios reduce β proportionally.

Proof Sketch. Fix a claim c of veil tier τ.

Claim-side analysis. Veil amplifiers are selected using an on-chain verifiable random function (VRF) modeled as a random oracle. The adversary $\mathcal{A}$ learns which veil slots it controls only after c is broadcast. To identify the real claim transaction, $\mathcal{A}$ must occupy all τ veil slots in the amplification set S_c, which occurs with probability at most α^{τ}.

Deposit-side analysis. Deposits are incorporated into the shielded pool via batched state transitions and protocol-injected decoy commitments. Deposit amounts are bucketed to eliminate value-based fingerprinting. As a result, $\mathcal{A}$ can distinguish a real user deposit from protocol-managed entropy with probability at most β, where β captures residual structural leakage under conservative assumptions.

Composition. To successfully deanonymize a payment, $\mathcal{A}$ must correctly identify both the originating deposit and the corresponding real claim. These events are independent under the

model assumptions, yielding a multiplicative bound:

$$\Pr[\mathcal{A} \text{ identifies real deposit and claim}] \leq \beta \cdot \alpha^{\tau}.$$

Timing and ordering side-channels are further mitigated via randomized execution delays $\delta_i \sim \text{Uniform}[0, 30]$ seconds and recursive STARK aggregation, which validates all $k + 1$ claim transactions in a single proof without revealing execution order.

Concrete Bounds (High-Tier Snip). For a "Sovereign" snip with $\tau = 127$ and conservative $\beta \leq 0.1$:

Adversary Stake α	**De-anonymisation Probability**
$1/3$ ($\sim 33\%$)	$\leq 1.8 \times 10^{-41}$
$1/2$ (50%)	$\leq 5.4 \times 10^{-40}$
$2/3$ ($\sim 66\%$)	$\leq 8.4 \times 10^{-24}$
0.99 (99%)	$\leq 2.8 \times 10^{-7}$

Even under extreme adversarial control, bidirectional anonymity imposes a multiplicative penalty on deanonymisation, rendering linkage attacks economically and statistically impractical.

Comparison to Prior Systems. Unlike passive anonymity systems with fixed set sizes, veil amplification is *active* and economically reinforced. Tornado-style mixers and Railgun rely on static pools whose anonymity degrades under surveillance pressure. Monero employs a fixed ring size (currently 16), offering bounded probabilistic privacy.

By contrast, ZK Snip's veil multiplier k scales with protocol revenue, making privacy a function of usage. As volume increases, the anonymity set expands and the cost of attack grows superlinearly.

Privacy therefore does not merely persist — it compounds.

6.9 Sybil Resistance (Design Outline)

Sybil resistance is achieved via: (i) minimum bonded stake per amplifier identity, (ii) stake-weighted sampling, (iii) slashing for missed veil duties, and (iv) optional reputation scoring. The selection probability is expensive to manipulate and economically punishable.

7 Selective Proof-Based Compliance (Transaction-Level Pre-Checks)

Privacy and assurance are not opposites. Many payment flows require *assurance*—that a transaction satisfies constraints—without requiring continuous identity disclosure. ZK Snip implements assurance through **transaction-level pre-checks**: lightweight on-chain gates whose predicates are satisfied by succinct ZK proofs generated against published accumulator roots. If a proof fails, the transaction reverts; if it succeeds, the payment proceeds with default privacy intact. The following design is supported by a live execution environment.

7.1 Decentralized SAL Execution via ZK-Verified Accumulators

ZK Snip's selective compliance mechanism is implemented through accumulator-based predicates enforced by the Selective Assurance Layer (SAL). Accumulator roots encode allow-lists, deny-lists, jurisdictional constraints, or other rule sets relevant to a transaction. Correctness of these accumulators is not assumed — it is proven.

SAL execution is backed by a decentralized proving infrastructure based on Herodotus Sentinel. Each accumulator is governed by a dedicated proving module that defines: (i) authoritative data sources (on-chain registries, attestation contracts, or other verifiable inputs), (ii) update logic encoded in Cairo, and (iii) a zero-knowledge proof that every accumulator root transition was computed correctly from verified chain data.

No operator can arbitrarily modify accumulator state. All updates are accepted on-chain only if accompanied by a valid proof of correctness. As a result, compliance rules are enforced cryptographically rather than discretionarily.

Operators executing SAL do not decide what enters an accumulator. They: (i) run the proving infrastructure, (ii) submit accumulator update proofs on-chain, and (iii) serve inclusion or exclusion proofs to users at claim time.

Liveness is ensured through redundancy. Accumulator updates are deterministic given the same authoritative inputs; therefore, any operator can independently reproduce the correct state and submit a valid proof. If one operator withholds updates or goes offline, others continue without coordination. Safety is cryptographic; liveness is economic.

7.2 Decentralized Compliance via Shared Operator Set

ZK Snip aligns SAL execution with its existing privacy infrastructure by allowing the same staked operator set to provide both veil amplification and compliance proving. This creates a unified infrastructure layer in which privacy amplification and compliance enforcement share incentives, execution guarantees, and fault tolerance.

7.3 Architecture: Roots + Proofs + Pre-Check Hook

The assurance stack has three components:

- **Off-chain accumulator**: Merkle / Sparse Merkle / MMR encoding large datasets (allow-lists, deny-lists, risk tiers, attestation registries, sanctions sets, sybil/humanhood registries). Only a **root** is published on-chain.
- **ZK coprocessor / prover**: heavy computation (set membership/non-membership, multi-criteria scoring, cross-domain checks) is performed off-chain and proven succinctly.
- **On-chain pre-check hook**: a minimal verification step checks the proof against the latest root and reverts on failure.

This pattern permits contracts to enforce rules over large datasets cheaply and composably.

7.3.1 Concrete Pre-Check Flow (Accumulator-Based Enforcement)

Optional compliance pre-checks in ZK Snip are implemented via accumulator-based rule enforcement combined with zero-knowledge coprocessing.

Externally defined association sets (e.g., allowed or disallowed classes of participants or funds) are committed to using public accumulator roots. These roots are updated independently of transaction execution and are verifiable on-chain.

Example predicates. A pre-check proof can assert statements of the form:

1. **Allow-set membership:** "the depositor holds a valid attestation from an approved issuer under root ρ_{allow},"
2. **Deny-set non-membership:** "the depositor (or funds source tag) is *not* contained in a published deny list under root ρ_{deny},"
3. **Risk-tier bound:** "a private risk score computed off-chain is below threshold T under policy hash pid,"

without revealing the attestation, the identity, the underlying score, or the feature inputs used to compute it. Only predicate satisfaction against the current roots is proven, and enforcement occurs *before* state transition.

Prior to value transfer, a participant may optionally generate a zero-knowledge proof asserting that the transaction satisfies a given rule with respect to the current accumulator state. The proof attests only to predicate satisfaction; no identity, transaction history, or counterparty information is revealed.

The protocol verifies this proof against the accumulator root before allowing state transition. Because enforcement occurs as a pre-check, non-compliant transactions are rejected without revealing additional information or weakening anonymity guarantees.

This model enables selective rule enforcement without introducing persistent identity, mandatory disclosure, or surveillance mechanisms.

7.4 Privacy of the Rules

Proofs can attest that "rule-set R is satisfied under root ρ" without disclosing full criteria, intermediate signals, weights, or proprietary sources. This reduces adversarial gaming while preserving verifiable assurance.

7.5 Deployment Modes

The same pre-check pattern supports:

- **Contract-level enforcement**: gate specific operations (deposit, claim, sponsorship) behind pre-checks when required by integrators.
- **System-level enforcement**: enforce shared rule-sets at gateways/regulated zones without changing application logic.

Default mode is **no pre-check** (max privacy). Assurance is **opt-in**.

7.6 Proof Types Supported

- **Membership proofs**: inclusion in allowed association sets (e.g., "KYC'd by an approved attester").
- **Non-membership proofs**: exclusion from deny lists (e.g., sanctions) using sparse trees.
- **Multi-criteria proofs**: composite conditions (wallet age, reputation score thresholds, risk constraints) without revealing individual signals.
- **Sybil/bot filtering**: uniqueness or anti-sybil criteria for distributions and sponsorship programs.

7.7 Integration with Snips (Optional Badge)

A snip may optionally carry an **assurance badge** attached at deposit time: a ZK attestation that the deposit satisfies an association-set predicate under a published root. At claim time, recipients may inherit the badge for downstream assurance or drop it to retain only default privacy.

7.8 Implementation Status

The selective compliance mechanisms described in this section are not purely theoretical. A production-grade execution environment implementing transaction-level, proof-based compliance has been developed and deployed by core contributors to the ZK Snip project.

This environment verifies compliance predicates via zero-knowledge proofs at execution time, without revealing identities, transaction histories, or counterparties, and without introducing persistent identifiers or transaction graph observability.

The implementation enforces correctness, liveness, and settlement of compliance-related execution as part of protocol operation, demonstrating that selective compliance can be cryptographically enforced without weakening default privacy guarantees.

Details of the reference implementation are available to reviewers under controlled disclosure.

8 Fee Model and Sponsorship

ZK Snip charges a settlement fee on transferred value. Default payer is the sender; the protocol supports **fee sponsorship** (third-party payment of fees) for payroll, enterprise payouts, exchanges, and consumer apps.

Sponsorship is represented as a separate commitment verified at claim time without linking sponsor identity to recipient identity. This allows applications to abstract fees away from end users, aligning with the principle that real users should not need to understand gas or fee topology.

9 Anonymity Set Growth Under Veil Amplification

ZK Snip aims to suppress linkage by ensuring that each successful claim event is accompanied by k additional *indistinguishable* veil events, such that an external observer sees $k + 1$ events

that are identical with respect to all observable features the protocol can standardize (amount bucket, calldata shape, gas pattern, timing window, etc.).

9.1 Basic Model

Consider a time window W and an amount bucket b. Let:

- $R_{b,W}$ be the number of *real* claims in bucket b during window W,
- $k_{b,W}$ be the veil multiplier applied in (b, W),
- $E_{b,W}$ be the total number of observed claim-like events (real + veil) in (b, W).

Assuming each real claim triggers exactly $k_{b,W}$ veil events,

$$E_{b,W} \;=\; (k_{b,W} + 1)\, R_{b,W}. \tag{1}$$

9.2 Observer Posterior and Local Anonymity

If the observer cannot distinguish which event is real within (b, W), then for any observed event e,

$$\Pr[e \text{ is real} \mid b, W] \;=\; \frac{R_{b,W}}{E_{b,W}} \;=\; \frac{1}{k_{b,W} + 1}. \tag{2}$$

The immediate local anonymity set size is thus:

$$A_{b,W}^{(\text{local})} \;=\; k_{b,W} + 1. \tag{3}$$

9.3 Effective Anonymity at Bucket/Window Level

A proxy for effective anonymity is the number of plausible indistinguishable candidates in (b, W):

$$A_{b,W}^{(\text{eff})} \;\approx\; E_{b,W} \;=\; (k_{b,W} + 1)\, R_{b,W}. \tag{4}$$

This shows that anonymity grows multiplicatively in $(k + 1)$ and linearly in organic usage $R_{b,W}$.

9.4 Entropy View

When all $k + 1$ events are equally likely to be real, the per-claim Shannon entropy is:

$$H \;=\; \log_2(k + 1). \tag{5}$$

9.5 Volume-Driven Compounding

Let $V_b(t)$ be cumulative transfer volume in bucket b up to time t, f the settlement fee rate (e.g., 0.20%), and α the fee share allocated to amplification rewards. The cumulative budget available for amplification is:

$$B_b(t) \;\approx\; \alpha\, f\, V_b(t). \tag{6}$$

If the expected cost per veil event in (b, W) is $c_{b,W}$, then achievable k satisfies:

$$k_{b,W} \lesssim \frac{B_b(W)}{c_{b,W}\, R_{b,W}}. \tag{7}$$

Equations (4)–(7) formalize the flywheel: increased volume grows budget; budget increases k; k increases anonymity; improved privacy attracts more usage.

9.6 Limitations

ZK Snip does not claim unconditional anonymity. Privacy guarantees depend on sufficient participation, adherence to standardized transaction templates, and the absence of adversaries exceeding the assumed stake and network control thresholds.

Deposit-side entropy and claim-time veil amplification together define an upper bound on unlinkability under the stated assumptions. If participation temporarily decreases, anonymity sets may contract; however, protocol-level controls and economic incentives are designed to restore equilibrium. The system is therefore resilient but not assumption-free.

This analysis assumes veil and real events are indistinguishable with respect to observed features. Deviations (gas or calldata differences, insufficient bucketing, amplifier non-performance, timing leakage) reduce effective anonymity. ZK Snip therefore standardizes templates and enforces liveness (with slashing) to preserve conditions under which Eqs. (2)–(4) are meaningful.

10 Competitive Landscape (High-Level)

Most existing privacy payment systems focus on concealing transaction details within a shielded domain. However, they differ significantly in how entry and exit points to the anonymity set are exposed.

Table 1: Boundary Visibility in Privacy Payment Systems (Entry and Exit Surfaces)

System	**Visible Entry**	**Visible Exit**	**Persistent Addresses**	**Boundary Correlation**
Tornado-style Mixers	Yes	Yes	Yes	High
Railgun	Yes	Yes	Yes (shielded)	High
Zcash (Shielded Pools)	Yes (t→z)	Yes (z→t)	Optional	Medium–High
Monero	Yes (ring entry)	Yes (ring exit)	Yes	Medium
ZK Snip	**No**	**No**	**No**	**Structurally Suppres**

Comparison reflects publicly documented models and design choices; parameters may change.

Table 2: Comparison of Leading Privacy Payment Systems (2025–2026)

Feature	**Railgun**	**Solana Confidential Transfers**	**Canton Network**	**Zama FHEVM**	**ZK Snip (This Work)**
Privacy Mechanism	zk-SNARK shielded balances	ZK-ElGamal encrypted balances/transfers	Permissioned sub-ledgers (no global view)	FHE encrypted computation	**STARK shielded pool + active veil amplification (8–127 decoys)**
Address Handling	Shielded addresses required	Standard addresses required	Persistent party IDs	Addresses for encrypted states	**No address ever — expirable snip codes**
UX Flow	Shield → interact → unshield	Send to address (privacy toggle)	Enterprise SDK/nodes	Encrypted txs (slow proving)	**Paste snip in any wallet → instant claim**
Wallet Compatibility	Requires Railgun SDK/wallet	Any Solana wallet (Token-2022)	Custom enterprise tools	Custom FHE wallets	**Any standard wallet (MetaMask, Phantom, etc.)**
Quantum Resistance	Vulnerable (Groth16)	Vulnerable (ElGamal)	Mixed (depends on impl.)	Mostly resistant (FHE)	**Full STARKs + hash-only**
Anonymity Set Growth	Passive (shielded pool size)	Passive (adoption)	Fixed (permissioned)	Passive	**Active + revenue-driven (veil k grows with fees)**
Yield/Economic Incentives	Partial governance fees	None	None (licensing)	None	**70% fees + emissions to Privacy Amplifiers**
Compliance Tools	Proofs of innocence	Optional view keys	Built-in permissions	Encrypted but auditable	**Optional privacy-pool style proofs**
Primary Focus	Private DeFi	Native token privacy	Institutional finance	Encrypted AI/DeFi	**Addressless private payments**
2025 Traction	$150M+ monthly volume	High launch visibility	Institutional pilots ($185M funded)	$150M+ funding	**Emerging: UX + yield moat**

==========

11 Conclusion

ZK Snip removes persistent identity surfaces, converts privacy into yield-bearing infrastructure, and reconciles default privacy with optional assurance. Privacy compounds with usage. Assurance is surgical. The rational action strengthens the system for everyone. Crucially, these guarantees are enforced by an operational system today, not deferred to future coordination assumptions.

A Attack Model and Assumptions

This appendix outlines adversarial capabilities and assumptions under which the anonymity arguments in Section 9 hold.

A.1 Adversary Capabilities

We consider a global passive adversary with:

- **Full blockchain visibility**: observes all on-chain transactions, logs, calldata, gas usage, and timestamps.
- **Protocol knowledge**: knows ZK Snip's public specification including veil amplification rules.
- **Arbitrary offline analysis**: can analyze full history at unlimited compute.
- **Limited network observation**: may observe coarse timing/propagation signals but does not control all network paths.

We do not assume cryptographic breaks of hashes, encryption, or ZK systems.

A.2 Non-Capabilities

The adversary is assumed not to have:

- the snip code or derived keys for ciphertext decryption,
- user secrets needed to derive nullifiers,
- control over the randomness beacon/VRF used for amplifier selection,
- guaranteed collusion with all amplifiers simultaneously.

A.3 Attack Surfaces Considered

The attack model considers adversaries attempting to correlate deposits with claims via transaction timing, value patterns, pool state transitions, and amplifier participation. In addition to claim-side observation, the model explicitly considers deposit-side correlation attempts, including inference based on deposit cadence, commitment frequency, and pool update structure.

- **Graph analysis**: linking deposits to claims via timing, amount, or structural patterns.

- **Amount correlation**: exploiting unique or rare amounts.
- **Correlation of deposits and claims across time, amount buckets, and batch boundaries**
- **Timing correlation**: exploiting deterministic claim/veil scheduling.

  Latency probing: attempting to infer the real claim via amplifier delays, partial participation, or coordination-window variance
- **Amplifier manipulation**: withholding veil events or emitting distinguishable templates.
- **Sybil capture**: biasing selection by spawning identities.

A.4 Deposit-Side Mitigations

ZK Snip mitigates deposit-to-claim correlation through protocol-managed deposit-side entropy. Deposits are committed to the shielded pool via batched state transitions, reducing the granularity of observable deposit events. Protocol-injected decoy commitments further obscure the true number and timing of user deposits. In low-volume regimes, these decoys also serve as a yield-funded bootstrap control that maintains a minimum effective batch size, bounding entry-side correlation power.

Amount bucketing ensures that deposit values do not provide distinguishing signals. Because deposit commitments are indistinguishable from protocol-generated commitments, an adversary cannot reliably determine whether a given pool update corresponds to a user deposit, a batch aggregation, or protocol-controlled entropy expansion.

These mechanisms eliminate direct temporal and structural linkage between deposit events and subsequent claims, forcing adversaries to rely on probabilistic inference across both entry and exit layers.

A.5 Mitigations

ZK Snip mitigates these via:

- **Amount bucketing**
- **Template standardization**
- **Bounded timing jitter**
- **Coordination-window enforcement with slashing and replacement/treasury failover to prevent adversarially induced timing distinguishers**
- **Bidirectional noise injection at both deposit and claim phases, combined with randomized batching and aggregated proof verification**
- **VRF-based selection**
- **Bonding + slashing**

A.6 Economic Resilience and Bootstrap Dynamics

The veil amplification mechanism is revenue-responsive by design: higher claim volume generates greater fee income, which increases rewards for Privacy Amplifiers and supports larger veil multipliers k. This directly expands anonymity sets and tightens unlinkability guarantees.

As with any activity-funded system, early-stage operation may exhibit lower real yield prior to reaching steady-state throughput. ZK Snip therefore incorporates explicit bootstrap and stabilization mechanisms to ensure continuity of privacy guarantees and incentive alignment during low-volume regimes.

A.7 Adaptive Fee Control

Claim fees are dynamically adjustable within governance-defined bounds. The protocol targets a baseline settlement fee (e.g., 0.20%), with an adaptive premium applied if the observed average veil multiplier k falls below a target threshold (e.g., $k = 64$). During bootstrap or demand contraction phases, higher effective fees increase amplifier yield, attracting additional participation until equilibrium is restored.

This mechanism treats privacy strength as a priced resource and prevents underfunded veil operation during transient demand troughs.

A.8 Veil Treasury Reserve

A protocol-controlled reserve is established to smooth early-stage incentives and maintain baseline privacy guarantees. The reserve serves two functions:

- **Yield floor provisioning.** A minimum target return range for Privacy Amplifiers (e.g., 15–20% annualized) can be supported during the initial network maturation period.
- **Protocol-operated veils.** Treasury-funded veil traffic may be deployed to enforce a minimum effective k when organic activity is temporarily insufficient.

Control of the reserve transitions to decentralized governance after predefined operational milestones (e.g., sustained throughput or maturity thresholds), ensuring that subsidy mechanisms are explicitly temporary.

A.9 Emission Tapering

Early-stage emissions may supplement activity-funded rewards to accelerate amplifier participation. These emissions are front-loaded and linearly tapered, with the system designed to transition toward fee-dominant compensation as transaction volume increases.

A.10 Steady-State Behavior

Beyond a threshold throughput regime (e.g., annualized settlement volume exceeding several hundred million units of value), protocol fees alone are sufficient to sustain competitive amplifier returns. In this regime, veil intensity is primarily constrained by governance policy rather than economic scarcity, and anonymity sets become robust to short-term fluctuations.

Overall, ZK Snip's incentive architecture is intentionally anti-fragile: early-stage risks are bounded, parameterized, and self-correcting, while mature operation converges toward subsidy-free, economically enforced privacy.

B Proof of Concept and Implementation Status

ZK Snip is accompanied by a working proof of concept (PoC) that validates the feasibility of the protocol's core architectural claims. The PoC is not intended as a full production deployment of all protocol components, but as an executable specification demonstrating correctness, composability, and incentive alignment of ZK Snip's core privacy and incentive mechanisms.

B.1 Scope

The PoC implements the full end-to-end payment flow, including shielded deposits, snip generation and redemption, nullifier-based non-spend enforcement, veil amplification, and selective proof-based compliance checks. Both deposit-side and claim-side entropy injection mechanisms are exercised. The PoC exercises decentralized compliance execution, including accumulator updates, proof verification, and liveness under redundant operator execution.

B.2 Architecture Overview

The implementation consists of: (i) on-chain contracts managing the shielded pool, voucher commitments, and nullifiers; (ii) off-chain components responsible for encrypted payload handling and amplifier coordination;

B.3 Validated Properties

The PoC demonstrates that addressless payments can be executed without revealing persistent identifiers, that veil amplification produces computationally indistinguishable transaction events, and that bidirectional anonymity is preserved under simulated adversarial observation. Selective compliance predicates are verified without exposing identity, transaction history, or counterparties.

The PoC includes a live, decentralized compliance execution environment that verifies accumulator updates via zero-knowledge proofs, serves inclusion and exclusion proofs to users, and enforces correctness without trusted operators. Compliance execution is performed by the same staked infrastructure used for privacy amplification.

The PoC additionally validates decentralized SAL execution. Accumulator roots are updated via ZK-verified proofs produced by redundant operators, and transaction-level compliance predicates are enforced without identity disclosure. Correctness does not rely on trusted execution or centralized operators.

B.4 Selective Compliance Engine Status

While the PoC validates selective compliance semantics end-to-end, the compliance execution environment itself is not experimental.

A production-grade selective compliance engine implementing transaction-level, proof-based rule enforcement has been developed and deployed by core contributors to the ZK Snip project.

This engine verifies zero-knowledge compliance attestations, enforces correctness and liveness, and settles execution outcomes without revealing identities, transaction history, or counterparties, and without introducing persistent identifiers or transaction graph leakage.

The PoC integrates with this execution environment to demonstrate composability between privacy amplification and compliance enforcement.

A reference implementation is available to reviewers under controlled disclosure.

B.5 Privacy Amplification Coprocessor Status

The privacy amplification mechanisms described throughout this paper are enforced in practice by a dedicated execution coprocessor that is already implemented and deployed.

This coprocessor is responsible for verifying veil amplification correctness, coordinating amplifier participation, enforcing liveness constraints, and settling amplifier rewards as part of protocol execution.

By offloading amplification verification and coordination into a specialized execution environment, ZK Snip ensures that privacy amplification is not advisory or best-effort, but a cryptographically and economically enforced protocol invariant.

The coprocessor operates independently of user wallets and does not introduce additional trust assumptions, persistent identifiers, or transaction graph observability.

The PoC integrates with this coprocessor to validate end-to-end composability between addressless payments, bidirectional anonymity, privacy amplification, and selective compliance.

Details of the reference implementation are available to reviewers under controlled disclosure.

B.6 Limitations

The PoC prioritizes correctness over optimization. Latency, throughput, and economic parameters related to veil amplification are intentionally conservative, and several production hardening steps—such as amplifier decentralization, MEV resistance, and network-level obfuscation—are out of scope.

An executable reference implementation exists and can be made available to reviewers under controlled disclosure.