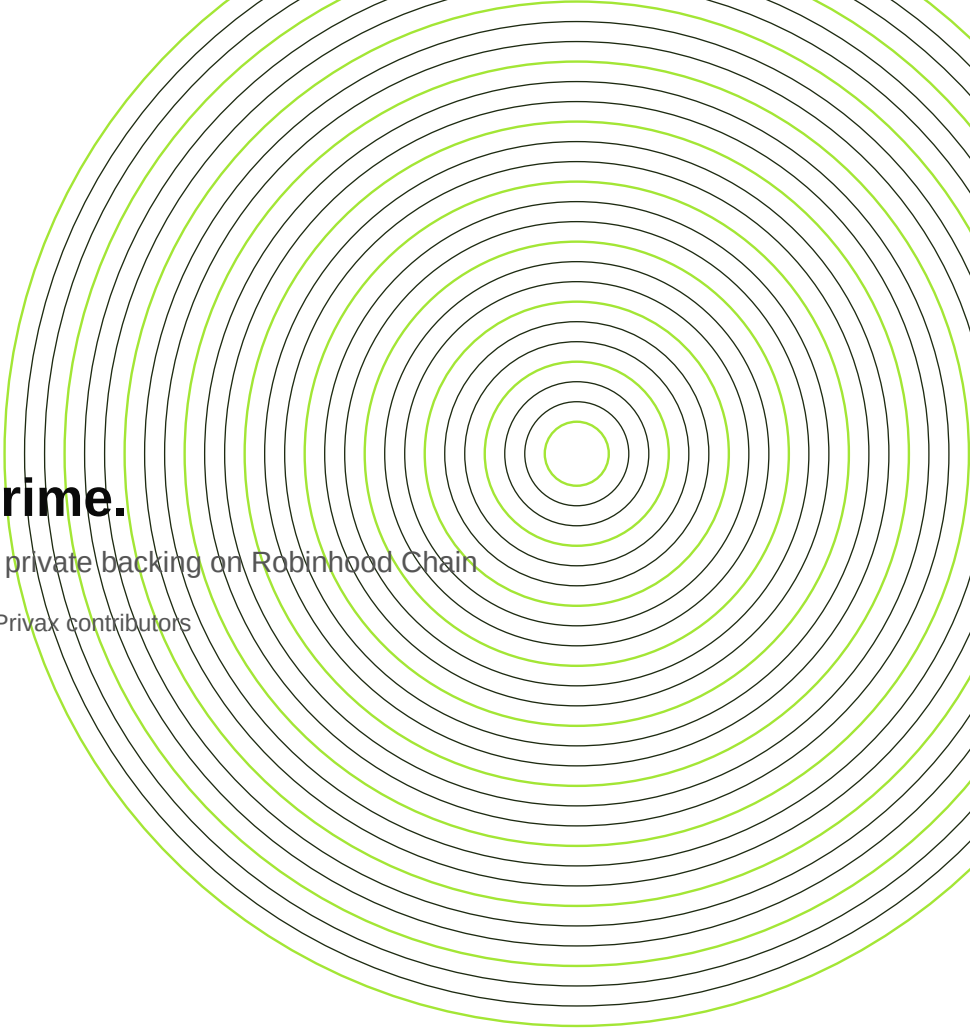




PRIVAX PROTOCOL PAPER

Privacy is not a crime.

Stealth payments and fee-funded private backing on Robinhood Chain

Protocol paper v1.1 — October 2026 — Privax contributors

Abstract

Privax is a recipient-privacy protocol for PRIVAX on Robinhood Chain. It separates a conventional public ERC-20 market from a share-accounting program in which payments can be assigned to one-time stealth addresses. A recipient publishes two compressed secp256k1 public keys. A sender combines those keys with fresh randomness to derive a destination that only the recipient can recognise and control. The chain records the sender, amount, one-time destination and announcement data, but does not publish the relationship between that destination and the recipient's ordinary wallet.

The economic layer routes a 3% Pons creator fee to the Privax Fee Circuit. 100% of newly claimed fee value is directed to the protocol, acquiring PRIVAX for the Privax Math. Donations increase backing without minting shares. The protocol therefore couples private custody with a transparent, market-funded increase in assets per share, while making no claim to hide senders, amounts, timing or network metadata.

1 Introduction

On a public blockchain every balance and every payment is visible forever. Paying someone reveals your balance and history to them and to everyone else; visible wallets become targets, and counterparties can rebuild a person's finances.

PRIVAX focuses on the part of that problem that can be solved without new cryptography or trusted parties: **receiving**. With stealth addresses, the person who is paid stays unlinkable. PRIVAX then adds an economic reason to hold privately: only the Privax Math earns from trading.

1. **Trades anywhere.** PRIVAX is a plain Pons ERC-20.
2. **Unlinkable receiving.** Every payment goes to a fresh stealth address.
3. **Privacy is where holding pays.** Creator fees are harvested into the Privax Math; public PRIVAX receives nothing.

2 Problem statement

Account-based ledgers make repeated addresses convenient, but that convenience produces a durable social and financial graph. A public receiving address connects counterparties, balances and future activity. Address rotation alone does not solve the problem when a sender cannot derive a recipient-controlled destination without coordination.

Privax narrows the objective: provide non-interactive, unlinkable receiving while preserving standard token trading and auditable solvency. The protocol deliberately avoids claiming transaction invisibility. Its privacy set depends on uniform denominations, delayed spending, independent destinations and the number of similar payments on-chain.

3 Design goals

When two goals conflict, the earlier one wins.

1. **Honest claims.** The interface and docs state exactly what is and is not hidden.
2. **Recoverable.** Keys derive from one wallet signature; all funds are recoverable from chain data.
3. **Compatible.** Any exchange interface, wallet or scanner treats PRIVAX as a normal token.
4. **Few dependencies.** No oracle, no committee, no dedicated relayer. Anyone can relay or index.
5. **Invisible to the user.** One signature on first use; no extra seed phrase; no ETH on stealth addresses.
6. **Explicit trust boundaries.** Administrative powers, market dependencies and privacy leakage are documented rather than hidden.

4 System architecture

```
App ■■■buy/sell■■■> PrivaxProtocol ■■■shield■■■> PrivaxMath <■■■donate■■■ PrivaxFeeCircuit
<■■■claim■■■ Pons FeeEscrow
■■■
```

■ registerKeys ■ trade ■■ Announcement events ■■> recipients scan

▼ ▼

CryptographicPrivax Pons curve → Uniswap v4 pool

Component	Role
PRIVAX	Pons ERC-20, 3% creator fee
PrivaxMath	Shares, stealth transfers, signed unshields, price average, network fee
PrivaxProtocol	ETH ↔ program shares in one transaction
CryptographicPrivax	Stealth meta-addresses
PrivaxFeeCircuit	Claims creator fees, buys PRIVAX, donates to the Privax Math

5 Stealth-address construction

5.1 Key derivation. The app asks the wallet to sign a fixed domain-separated message. The signature is deterministically expanded into spending scalar k and viewing scalar v ; their public keys are $K = kG$ and $V = vG$ on secp256k1. The 66-byte concatenation of both compressed keys is registered in the ownerless CryptographicPrivax under scheme identifier 1.

5.2 Destination derivation. The sender samples cryptographically secure random r , publishes $R = rG$, computes shared point rV and scalar $s = \text{keccak256}(rV)$. The one-time public key is $P = K + sG$; its Ethereum address is the payment destination.

5.3 Discovery and control. A recipient computes $vR = rV$, derives the same s , and checks whether $\text{addr}(K + sG)$ equals the announced destination. The corresponding private scalar is $p = (k + s) \bmod n$. A one-byte view tag lets scanners reject most unrelated announcements before full curve arithmetic.

6 Registration and announcements

The CryptographicPrivax stores one current meta-address per registrant and rejects values not exactly 66 bytes. It has no owner and no privileged mutation path: only the caller can replace the value mapped to that caller.

The Privax Math emits the ERC-5564-shaped **Announcement** event with scheme identifier, stealth address, caller, ephemeral public key and metadata. Events provide discovery, not secrecy: all fields are public and permanent. The protocol's privacy property follows from the hardness of deriving the recipient relationship without viewing scalar v .

7 Transaction lifecycle

- 1. Acquire.** ETH is swapped for PRIVAX on the active Pons market and shielded in one transaction.
- 2. Shield.** Public PRIVAX is transferred into the program and converted into shares.
- 3. Send.** Shares move to a derived one-time address and an announcement is emitted.
- 4. Discover.** The recipient scans announcements locally using the viewing key.
- 5. Spend.** The one-time key signs a typed transfer or unshield request; any account may relay it.
- 6. Exit.** PRIVAX may be unshielded directly or sold through PrivaxProtocol for ETH.

Each signed request commits to the owner, destination, amount, fee, current nonce and deadline. The announcement hash is included for transfers; sale destination and minimum output are bound through the unshield **extra** field.

8 Share accounting

Let B be **totalBacking**, S be **totalShares**, and $V = 10^6$ virtual shares. Depositing a assets mints $\text{floor}[a(S+V)/(B+1)]$ shares. Converting s shares reports $\text{floor}[s(B+1)/(S+V)]$ assets. Withdrawals add one

share when the rounded conversion would underfund the requested asset amount, placing rounding cost on the withdrawer.

A donation increases **B** without changing **S**. Consequently, the represented asset value of each existing share rises. The virtual offset makes first-deposit manipulation more expensive and defines behavior at initialization; it does not eliminate market, implementation or administrative risk.

Accounting scope. Program shares are internal balances, not transferable ERC-20 tokens. Solvency can be checked from the PRIVAX token balance, totalBacking, totalShares and emitted VaultUpdated events.

9 Market integration

PRIVAX begins on a Pons bonding curve paired with native ETH and can graduate to a Uniswap v4 pool. Shared market logic selects the active venue. PrivaxProtocol holds no funds between successful operations: **buy** acquires PRIVAX and shields it to the specified address; **buyToStealth** shields one fixed denomination to a fresh destination and returns excess acquired value as shielded shares to the caller.

For exits, **sell** verifies an EIP-712 unshield whose **extra** commits to the ETH recipient and minimum output. The router sells received PRIVAX, forwards any relayer fee shares, updates the price observation, and transfers ETH to the bound recipient.

10 Fee conversion and accrual

The Privax Fee Circuit must be configured as the Pons creator-fee recipient. A harvest attempts market settlement, sweeps available venue fees, claims from FeeEscrow and updates the program price observation. If no newly available ETH exists, it returns without changing backing.

100% of newly claimed ETH is directed to the protocol and used to acquire PRIVAX. Purchased PRIVAX is approved and donated, increasing backing without issuing shares.

11 Price bounds and execution

The program records tokens per ETH as a capped exponential moving average. Once initialized, at most one observation is folded per block. The prior spot observation is clipped to $\pm 10\%$ around the prior average, then combined as $EMA' = (7 \cdot EMA + \text{clippedSpot})/8$.

The Privax Fee Circuit requires output within 3% of this average. When a buy fails its bound, the candidate ETH and minimum output are halved, up to eight attempts. Unspent ETH remains available for a later execution. These controls constrain execution; they do not guarantee a fair price under thin liquidity, stale observations or adversarial transaction ordering.

12 Authorization and relaying

Stealth addresses need not hold ETH. The owner signs an EIP-712 request and a relayer submits it.

Signature validation supports externally owned accounts and compatible contract wallets. Per-owner nonces prevent replay; deadlines limit validity; exact destinations and amounts prevent substitution.

The program derives a suggested PRIVAX-denominated network fee from $\text{maxOpCostWei} \times \text{tokensPerEthEma} / 10^{18}$, refreshed no more than once per hour. This is a quote, not an enforced tariff: users sign the exact fee, including zero, and relayers independently decide whether to submit.

13 Denominations and anonymity sets

Announced payments accept 1,000, 10,000, 100,000 or 1,000,000 PRIVAX. Equal sizes reduce amount uniqueness and allow a payment to blend with other payments in the same class. Larger values can be composed from multiple outputs.

Fixed denominations do not create privacy by themselves. A thin class, unique timing, rapid consolidation, common funding source or immediate withdrawal can isolate a payment. The effective anonymity set is

behavioral and empirical, not a constant promised by the contracts.

14 Privacy analysis

Information	On-chain visibility
Recipient identity behind a fresh stealth address	Not published; discoverable with the viewing key or behavioral correlation
Sender / relayer	Visible
Payment amount	Visible fixed denomination
Ephemeral public key and view tag	Visible in Announcement
Shield and unshield endpoints	Visible
Program totals and market trades	Visible

Privax does not use zero-knowledge proofs, mixers or hidden amounts. RPC providers may observe queried addresses and IP metadata. A compromised viewing key reveals incoming-payment history but cannot spend; a compromised spending key can authorize movement. Reusing destinations or withdrawing to a previously linked wallet defeats recipient unlinkability.

15 Security and trust model

Party	Can	Cannot
Privax Math owner	Transfer ownership; emergency-drain program PRIVAX and ETH	Forge a user signature
Privax Fee Circuit owner	Release protocol balances; drain its ETH and PRIVAX; transfer ownership	Change the fixed fee recipient; move program shares; forge user signatures
Relayer	Refuse or delay	Alter a signed operation
Sequencer	Censor or delay	Forge transactions

Critical trust assumption. The Privax Math owner can emergency-drain all PRIVAX and ETH held by the program, setting backing to zero. The contracts are unaudited. Users should treat this as custodial-risk exposure and limit deposits accordingly.

- Replay resistance: per-owner nonces and deadlines.
- Sale integrity: destination and minimum output are signed; only the bound router may submit nonzero-extra unshields.
- Accounting defense: virtual shares and conservative withdrawal rounding.
- Execution defense: capped EMA movement and 3% minimum-output bound.
- Availability limitation: relayers, RPC providers, the sequencer and market venues may delay or censor operations.

16 Failure modes and recovery

Condition	Result	Recovery
No creator fees available	Harvest returns zero	Wait for trading fees
Buy violates bound	Candidate is halved up to eight times	Unspent ETH remains for a later harvest

Expired or replayed signature	Operation reverts	Sign again with a fresh nonce/deadline
Lost local browser data	Cached discoveries disappear	Reconnect the same wallet, derive keys again and rescan events
Owner emergency drain	Backing becomes zero	Requires owner restoration; no automatic recovery

17 Chain and availability assumptions

PRIVAX targets Robinhood Chain Mainnet, chain identifier 4663, with ETH for gas. Correctness inherits the chain's execution and consensus assumptions. The sequencer can order, delay or censor transactions; it cannot produce a valid user authorization without the relevant key.

Availability also depends on Pons contracts, the active curve or v4 pool, RPC access and client-side event scanning. Privax introduces no committee and no designated relayer, but permissionless execution does not guarantee that an economically willing relayer exists at every moment.

18 Conclusion

Privax combines a conventional market asset with a deliberately narrow privacy primitive: unlinkable receiving through one-time addresses. Its accounting and fee flows remain publicly auditable; its limits remain publicly stated. The design favors interoperability, recoverability and explicit risk over broader but unsupported privacy claims.

A Deployment

Contract	Address
PrivaxMath	Awaiting Privax deployment
PrivaxProtocol	Awaiting Privax deployment
CryptographicPrivax	Awaiting Privax deployment
PrivaxFeeCircuit	Awaiting Privax deployment
PRIVAX token	Awaiting Privax deployment

B Protocol constants

Constant	Value
Creator fee	3% / 300 bps
Fee directed to protocol	100%
Maximum Privax Fee Circuit slippage	3% / 300 bps
Maximum EMA input step	10% / 1,000 bps
EMA weighting	7/8 prior average + 1/8 clipped observation
Virtual shares	1,000,000
Network fee interval	1 hour
Stealth scheme	ERC-5564 scheme 1, secp256k1

References

- [1] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008.
- [2] ERC-20: Token Standard.
- [3] ERC-5564: Stealth Addresses.
- [4] ERC-6538: Stealth Meta-Address Registry.
- [5] EIP-712: Typed structured data hashing and signing.
- [6] SEC 2: Recommended Elliptic Curve Domain Parameters, secp256k1.
- [7] Uniswap v4 Core technical documentation.
- [8] Pons V2 protocol documentation, docs.ponsfamily.com.
- [9] Privax Solidity source and deployed contract interfaces, October 2026.