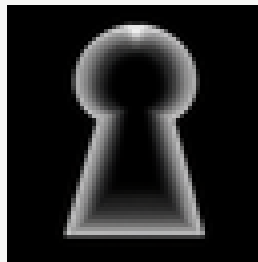


Privacy is normal.

A protocol for unlinkable receiving, share accounting, and fee-funded private backing on Robinhood Chain.



ABSTRACT

Secreteum separates a conventional public ERC-20 market from a private receiving layer. A recipient publishes a stealth meta-address. Each sender derives a fresh one-time destination from that public data and new randomness. Only the recipient can recognise and control the destination. The chain still exposes the sender, amount, timing, announcement data and program totals; the design does not claim sender or amount secrecy.

SECR deposited into the Quadratic Arithmetic Program is represented by internal shares. The Arithmetic Circuit receives a 2% creator fee from Pons trading, offers the executor a 0.5% tip, and by default uses 30% of the remainder to acquire SECR for donation and retains 70% as reserve. Donations increase backing without issuing shares.

Status: unaudited protocol software. This document is descriptive, not a promise of returns, anonymity, or uninterrupted access.

Problem and design principles

A narrow privacy guarantee is more useful than a broad claim the system cannot enforce.

1. Public address reuse

Account-based ledgers make balances and transfers globally inspectable. Reusing a receiving address links counterparties, holdings, timing, and later spending. Creating addresses manually does not solve coordination: a payer needs a destination the recipient can later discover and control without first exchanging a new address privately.

2. Protocol objective

Secreteum provides non-interactive recipient unlinkability. A user registers public key material once. Any payer can derive a unique destination, while the recipient can scan public announcements and recover control locally. The protocol keeps ordinary ERC-20 compatibility outside the private program and transparent reserve accounting inside it.

3. Ordered principles

Priorit y	Principle	Consequence
1	Honest privacy claims	Sender, amount, timing and entry/exit remain public.
2	Recoverability	The same wallet signature deterministically reconstructs viewing and spending keys.
3	Compatibility	SECR remains a standard ERC-20 on its public market.
4	Minimal coordination	No committee or mandatory relayer is required.
5	Auditable backing	Backing and share supply are observable on-chain.
6	Explicit control risk	Emergency powers and dependencies are disclosed.

4. Non-goals

Secreteum is not a mixer, rollup, privacy pool, zero-knowledge system, hidden-amount asset, or guarantee against traffic analysis. It does not conceal the funding of a public wallet, an unshield destination, RPC metadata, or correlations created by user behavior.

Protocol architecture

Four contracts separate discovery, accounting, market execution, and fee conversion.

Wallet / app → BilinearPairings → QuadraticArithmeticProgram ← ArithmeticCircuit ← Pons FeeEscrow

Component	Responsibility	Privilege
SECR	Public Pons ERC-20; 2% creator fee	Defined by Pons launch
Cryptographic Commitment	Stores 66-byte stealth meta-addresses	Ownerless; registrant updates own entry
Quadratic Arithmetic Program	Custodies SECR, records shares, verifies signed movement, emits announcements	Owner can transfer ownership and emergency-drain
Bilinear Pairings	Atomic ETH↔SECR market execution with shield/unshield	One-time token link; no continuing owner
Arithmetic Circuit	Claims creator fees, acquires SECR, donates backing	Owner sets acquisition share, reserve release, drain

5. Lifecycle

Acquire. ETH is exchanged on the current Pons venue and the resulting SECR is shielded. Shield. Public SECR enters the program and becomes shares. Send. Shares move to a one-time address with an announcement. Discover. The receiver scans events using the viewing key. Spend. The stealth key signs a typed request that any relayer may submit. Exit. SECR is unshielded directly or sold for ETH through Bilinear Pairings.

6. Deployment state

The four protocol contracts are deployed. The launched SECR token is 0x2735b276f992dc35558944f08e7e176b0a4b1a4e. Each transaction contract must be linked to this address through its one-time setup before user operations begin.

Stealth-address cryptography

ERC-5564 scheme 1 on secp256k1, with local discovery and one-time control.

7. Long-lived keys

A deterministic wallet signature is domain-separated and expanded into spending scalar k and viewing scalar v . Public points $K = kG$ and $V = vG$ are compressed and concatenated. Cryptographic Commitment accepts exactly 66 bytes and stores the current value under the registrant address.

8. Sender derivation

For each payment, the sender samples a fresh random scalar r , computes ephemeral point $R = rG$ and shared point rV , then hashes the shared point to scalar s . The one-time public key is $P = K + sG$. Its Ethereum address is the destination.

```
r ← random scalar R = rG s = keccak256(rV) P = K + sG destination = address(P)
```

9. Receiver discovery

The recipient computes vR , equal to rV , derives the same s and tests $\text{address}(K + sG)$ against the announcement. The one-time private scalar is $p = (k + s) \bmod n$. A one-byte view tag permits fast rejection of most unrelated announcements before full elliptic-curve work.

10. Announcement record

`Announcement(1, stealthAddress, caller, ephemeralPubKey, metadata)` is public and permanent. The event makes discovery possible; it does not encrypt the sender, value, block, ephemeral key, view tag, or destination.

11. Key compromise

A leaked viewing key reveals which incoming announcements belong to the user but cannot authorize spending. A leaked spending key, or the wallet capability that deterministically recreates both keys, compromises control. Users should treat wallet-signature access as equivalent to access to the private account.

Share accounting

A deterministic conversion layer makes donations accrue proportionally to existing shares.

12. State and conversion

Let B denote totalBacking, S totalShares, $V = 1,000,000$ virtual shares, a an asset quantity and s a share quantity.

$$\text{shares}(a) = \text{floor}[a(S + V)/(B + 1)] \quad \text{assets}(s) = \text{floor}[s(B + 1)/(S + V)]$$

Shielding transfers SECR into the program, increases B , mints the computed shares, increases S , and credits the destination. Unshielding computes required shares and adds one share if the rounded asset conversion would otherwise be below the requested amount. This conservative rule places rounding cost on the withdrawer.

13. Donation effect

`donate(amount)` transfers SECR into the program and increases B without changing S . Existing shares therefore represent more backing. A plain ERC-20 transfer to the program does not update B and is not an accounted donation.

14. Invariants and auditability

Successful shield, unshield, donation and emergency-drain paths emit `VaultUpdated(B,S)`. The represented balance of an address is `assets(sharesOf[address])`. Internal shares are not an ERC-20. Observers can compare recorded backing with the program token balance, but unsolicited token transfers may make the raw balance exceed recorded backing.

15. Empty-state defense

The virtual offset defines conversions before organic shares exist and raises the cost of first-deposit inflation attacks. It is a mitigation, not a formal proof against every economic manipulation or implementation defect.

Market and fee mechanics

Public liquidity supplies the fee stream; bounded execution converts it into program backing.

16. Market path

SECR trades against native ETH on a Pons bonding curve and, after graduation, through the associated Uniswap v4 pool. Shared market logic chooses the active venue. Bilinear Pairings acquires and shields in one transaction or unshields and sells under signed minimum-output terms.

17. Arithmetic Circuit

harvest() is permissionless at contract level, although the website exposes its control only in the developer admin panel. The call attempts graduation settlement, sweeps venue fees, claims from Pons FeeEscrow, updates the program price observation, calculates the caller tip, applies the acquisition/reserve split, buys SECR, and donates successful output.

Stage	Rule
Creator fee	2% / 200 bps at the Pons launch
Executor tip	0.5% / 50 bps of newly available ETH
Default acquisition share	30% / 3,000 bps of the post-tip remainder
Default reserve share	70% / 7,000 bps of the post-tip remainder
Donation	Acquired SECR increases backing without minting shares

18. Bounded execution

The Arithmetic Circuit sets minimum output at 97% of the EMA-implied result. If the buy reverts, both candidate ETH and minimum output are halved, up to eight attempts. Unspent ETH remains for a future call. These bounds reduce, but cannot eliminate, sandwiching, stale-price and thin-liquidity risk.

Price observation and relaying

Typed authorization lets one-time addresses move value without holding native gas.

19. Capped moving average

QuadraticArithmeticProgram.poke() reads tokens per ETH from the active market. On initialization, the spot becomes the average. On later blocks, the prior spot is clipped to the interval $[0.9 \cdot \text{EMA}, 1.1 \cdot \text{EMA}]$ and folded into the next value:

```
EMA' = floor[(7 · EMA + clippedSpot)/8]
```

Only one prior observation is folded per block. This produces a slowly moving execution reference, not an oracle guarantee. Permissionless calls can update it, while venue liquidity and transaction ordering still affect observations.

20. EIP-712 requests

Transfers and unshields commit to owner, destination, amount, relayer fee, nonce and deadline. Transfers also commit to a hash of announcement data. Unshields can bind extra terms; when extra is nonzero, only the destination contract may submit. Bilinear Pairings uses that rule to bind the ETH recipient and minimum sale output.

21. Network fee

The suggested fee is $\text{floor}(\text{maxOpCostWei} \times \text{EMA} / 10^{18})$, with a floor of one base unit and uint128 cap. It updates no more than once per hour. maxOpCostWei is immutable after deployment. The suggestion is not mandatory: the signer selects the exact fee and relayers decide whether it is economical.

22. Replay and expiry

Every owner has an independent nonce. A valid request increments it before movement, so replay fails. The contract rejects requests after their deadline and accepts ECDSA or compatible contract-wallet signatures through signature checking.

Privacy analysis

Recipient unlinkability is conditional on cryptography, local scanning, and disciplined behavior.

Signal	Visibility	Implication
Recipient identity	Not encoded in fresh destination	May still be inferred through later behavior
Sender or relayer	Public	Sender privacy is out of scope
Amount	Public fixed denomination	Equal sizes help only when the class is populated
Timing and block	Public	Immediate movement creates correlation
Shield/unshield endpoints	Public	Returning to a known wallet links activity
RPC queries	Visible to provider	Network metadata can reveal scanner interest
Viewing key	Private local secret	Leak reveals receipt history, not spending authority

23. Anonymity set

Announced payments are restricted to 1,000, 10,000, 100,000 or 1,000,000 SECR. A value may be composed from several outputs. Privacy improves when many unrelated users choose the same class; it weakens when a class is rare, timing is distinctive, or outputs are consolidated together.

24. Operational guidance

Use a fresh ETH destination when exiting. Avoid immediate movement after receipt. Avoid consolidating many one-time addresses in one burst. Do not reuse stealth destinations. Consider the RPC provider part of the metadata threat model. These practices reduce correlation but cannot guarantee anonymity.

25. Explicit exclusions

No zero-knowledge proof hides a sender or amount. No protocol mechanism obscures gas funding or network origin. No claim is made against global timing analysis, compromised clients, malicious wallet software, endpoint surveillance, or voluntary disclosure of a viewing key.

Security, deployment, and references

The system is verifiable, but it is unaudited and includes material owner powers.

26. Trust boundaries

Actor	Capability / risk
Program owner	Can transfer ownership and emergency-drain all program SECR and ETH, setting recorded backing to zero.
Circuit owner	Can set acquisition share from 0–100%, release reserve, drain Circuit ETH/SECR, and transfer ownership.
Bilinear Pairings initializer	Can link the token once; no continuing owner function after linkage.
Relayer	Can refuse or delay; cannot alter a valid signed request.
Sequencer / RPC / markets	Can delay, censor, observe metadata, or impair execution; cannot forge a user signature.

Unaudited. The contracts have not completed an independent security audit. Owner emergency powers are a custodial risk. Use only amounts consistent with that risk.

27. Deployed protocol addresses

Contract	Address
ArithmeticCircuit	0x931ed3f4537E437c49ff8EA3F3e9A049364F8Cd1
QuadraticArithmeticProgram	0x6bCedCdfF4244467eE6093Cf51f2fA70317144de
BilinearPairings	0xbe462B87B65E662586c119ECA67dAC9a6C3Ae6c6
CryptographicCommitment	0x62308ef2ED5479cEf0DE1c70D00336d78fF44c28
Developer / owner	0x6737c89192bF6ED589508eDA234B4059a8012f4C

Network: Robinhood Chain Mainnet · Chain ID 4663 · Gas asset: ETH · Scan origin: block 81,080,667. The token is linked after launch and can be read from QuadraticArithmeticProgram.zero().

28. References

[1] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008. [2] ERC-20. [3] ERC-5564. [4] ERC-6538. [5] EIP-712. [6] SEC 2, secp256k1. [7] Uniswap v4 Core documentation. [8] Pons V2 documentation. [9] Secreteum deployed interfaces and Solidity source, October 2026.