veil

A Confidential Token Protocol with Built-in Auditability for Robinhood Chain

Encrypted balances and hidden transfer amounts for standard ERC-20 assets on an unmodified EVM, using ElGamal encryption over Baby Jubjub and Groth16 zero-knowledge proofs, with protocol-level audit visibility designed for regulated, real-world-asset environments.

Whitepaper, Version 1.0 · September 2026

Live on Robinhood Chain · Chain ID 4663

ABSTRACT

Public blockchains expose every balance and every payment to every observer. For payroll, treasury operations, market activity, and ordinary commerce, that level of disclosure is not a feature but a structural defect, and it is felt most acutely on chains that carry regulated, real-world assets. veil is a confidential token protocol, live on Robinhood Chain, that encrypts account balances and transfer amounts for standard ERC-20 assets while keeping the chain itself unchanged: no protocol modification, no coprocessor, no custodian, and no new token. Balances are held as ElGamal ciphertexts over the Baby Jubjub curve; every transfer carries a Groth16 zero-knowledge proof that the hidden amount is well-formed, affordable, and conserved. Confidentiality is paired with accountability: every transfer additionally encrypts its amount to a protocol audit key, giving designated oversight full amount-level visibility without giving it any ability to move funds. A per-asset denomination layer extends the design across stablecoins, wrapped ether, fixed-supply launchpad assets, and ERC-8056 tokenized equities. This paper specifies the construction, its security model and trust assumptions, and the parameters of the live deployment.

1 Introduction

An account on a transparent ledger is a public financial biography. Anyone who learns that an address belongs to a person or a firm can read its balance, its counterparties, its salary, its margins, and its mistakes, in real time and retroactively. As tokenized cash and tokenized securities move on-chain, this transparency becomes commercially untenable: no business publishes its ledger, and no broker publishes its clients' positions.

The difficulty is that the obvious remedies discard properties that regulated environments require. Mixing services buy unlinkability at the price of compliance, and have been sanctioned accordingly. Dedicated privacy chains fragment liquidity and abandon the EVM. Encrypted-computation coprocessors introduce new trust in threshold committees and specialized infrastructure. What a real-world-asset chain needs is narrower and more disciplined: **confidentiality of amounts and balances, on the existing chain, for existing assets, with visibility preserved for designated oversight**.

veil takes the account-based confidential payment approach introduced by Zether [3] and deploys it as a converter-mode protocol on Robinhood Chain, an Arbitrum-Orbit EVM Layer 2. Users shield ordinary ERC-20 tokens into an encrypted balance, transfer value privately inside the shielded pool, and unshield back to plain tokens at will. The protocol deliberately implements a known, published scheme rather than novel cryptography; its contribution is an engineering and deployment one:

- a production deployment of Zether-lineage confidential transfers on a live retail L2, with browser-side proving against unmodified contracts;
- a per-asset denomination layer that reconciles 18-decimal assets with the 40-bit encrypted amount domain required for practical decryption, without touching the circuits;
- audit encryption on every transfer as a first-class protocol rule, matching the compliance posture of a regulated asset chain; and
- a uniform support model spanning stablecoins (USDG), wrapped ether, fixed-supply launchpad assets, and ERC-8056 tokenized equities.

veil charges no protocol fee and is distributed as free software under GPL-3.0. Sections 2–4 present the construction, Section 5 the denomination layer, Section 6 the auditability design, Section 7 the security model, and Sections 8–9 the live deployment and roadmap.

2 Preliminaries

2.1 Setting and notation

Let $\mathbb{G}$ be the prime-order subgroup of the Baby Jubjub elliptic curve [4], a twisted Edwards curve defined over the scalar field of BN254 and therefore efficient to arithmetize inside pairing-based SNARK circuits. We write G for the conventional subgroup generator (`Base8`) and l for the subgroup order,

$$l = 2736030358979909402780800718157159386076813972158567259200215660948447373041 \\ (\approx 2^{251}).$$

Secret keys are scalars $sk \in \mathbb{Z}_l$; public keys are curve points $pk = sk \cdot G$. Users derive their keypair deterministically from an ordinary wallet signature over a fixed message, so the same wallet reproduces the same veil keys on any device and no additional seed material exists to store or lose.

2.2 Additively homomorphic ElGamal encryption

A balance or amount m is encrypted to key pk with randomness r as the exponential ElGamal ciphertext [1]

$$\mathrm{Enc}(pk, m; r) \quad = \quad (C_1, C_2) \quad = \quad (r \cdot G, \quad m \cdot G + r \cdot pk),$$

which is computationally hiding under the decisional Diffie–Hellman assumption in $\mathbb{G}$ and additively homomorphic: componentwise addition of ciphertexts encrypts the sum of the plaintexts. Decryption computes $m \cdot G = C_2 - sk \cdot C_1$ and recovers m by solving a discrete logarithm that is deliberately bounded: all encrypted quantities in veil satisfy $m < 2^{40}$, so a baby-step giant-step search [7] with a table of at most 2^{20} entries recovers m in interactive time (Section 8). This bounded-exponent regime is the standard device of the Zether family and is the origin of the protocol's 40-bit amount domain.

2.3 Zero-knowledge proofs

Statements about hidden values are proven with Groth16 [2], a pairing-based zk-SNARK with constant-size proofs (eight field elements, 256 bytes of calldata) and inexpensive on-chain verification. Circuits are written in circom and proven with snarkjs [5]. Groth16 requires a per-circuit trusted setup; veil's setup posture is described, with the rest of the trust model, in Section 7.

3 Protocol Overview

veil consists of two core contracts and three verifier contracts, all immutable after deployment. The **Registrar** binds exactly one Baby Jubjub public key to each externally owned account. The **Vault** holds deposited ERC-20 tokens and, for every (asset, account) pair, an encrypted balance consisting of one ciphertext (C_1, C_2) and a monotonically increasing nonce. Each of the three operations that requires a proof (registration, confidential transfer, withdrawal) has a dedicated Groth16 verifier contract generated from its circuit.

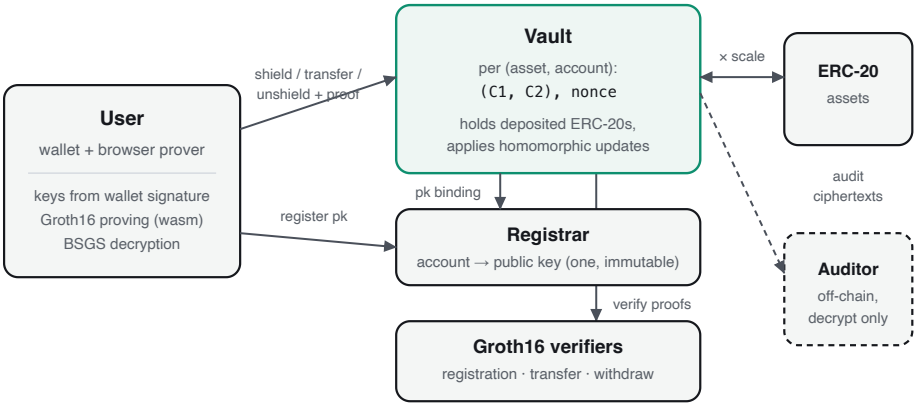

Figure 1. Protocol architecture. Proving and decryption happen client-side; the chain verifies proofs and applies homomorphic state updates; the auditor observes encrypted amounts from events and holds no spending power.

The lifecycle of funds has four movements.

Register. An account proves in zero knowledge that it knows the secret key for its claimed public key, with the account address and chain identifier bound into the statement so a proof cannot be replayed to register a key to someone else, or on another chain. Registration is one transaction per account, ever.

Shield. A deposit requires no proof at all. When an account shields v encrypted units of an asset, the Vault itself computes the trivial ciphertext $(O, v{\cdot}G)$ on-chain and adds it homomorphically to the account's balance while pulling the corresponding ERC-20 amount in. Because the credited ciphertext is computed by the contract from the public deposit amount, a depositor can only ever credit exactly what was received; an exact-receipt check additionally rejects nonstandard assets whose transfers deliver less than their face amount. Deposit amounts are public by nature; privacy begins after entry.

Transfer. A confidential transfer replaces the sender's ciphertext with a proven re-encryption of the reduced balance, homomorphically credits the receiver, and emits an audit ciphertext, all without revealing the amount or either balance (Section 4).

Unshield. A withdrawal proves ownership and affordability of a public amount v and pays out plain ERC-20 tokens. Exit amounts, like entry amounts, are public by nature.

Two representational invariants matter throughout. Balances are initialized as the canonical encryption of zero with fixed randomness, $\mathrm{Enc}(pk, 0; 1) = (G, pk)$, so that C_1 is never the group identity, a point the arithmetic circuits cannot process. And every proof about a balance is bound to the exact ciphertext currently in storage, so a proof formed against stale state is rejected rather than misapplied.

4 The Cryptographic Construction

4.1 Registration

STATEMENT · REGISTRATION (≈4,100 CONSTRAINTS)

Public: pk, account address a, chain identifier c. Private: sk.

1. $pk = sk \cdot G$.

Binding a and c as public inputs makes the proof valid only for the submitting account on the deployed chain. The Registrar additionally requires pk to be a valid, non-identity curve point and permits one immutable key per account.

4.2 Confidential transfer

Let $(C^{\text{in}}_1, C^{\text{in}}_2)$ be the sender's current on-chain ciphertext with nonce n, and let pk_s, pk_r, pk_A be the registered sender key, registered receiver key, and the protocol audit key. The sender proves the following statement, with balance b and amount v remaining private.

STATEMENT · TRANSFER (≈29,600 CONSTRAINTS)

Public: pk_s, pk_r, pk_A, $(C^{\text{in}}_1, C^{\text{in}}_2)$, n, and outputs $C^{\text{out,s}}$, $C^{\text{out,r}}$, C^{aud} (23 field elements in total). Private: sk, b, v, randomness r_1, r_2, r_3.

1. Key ownership: $pk_s = sk \cdot G$.
2. Balance opening: $(C^{\text{in}}_1, C^{\text{in}}_2)$ decrypts to b under sk.
3. Range and affordability: b, v, and $b - v$ each decompose into 40 bits, which enforces $0 \le v \le b < 2^{40}$ with no field wraparound.
4. Sender output: $C^{\text{out,s}} = \text{Enc}(pk_s, b - v; r_1)$.
5. Receiver output: $C^{\text{out,r}} = \text{Enc}(pk_r, v; r_2)$.
6. Audit output: $C^{\text{aud}} = \text{Enc}(pk_A, v; r_3)$, the same v by construction.

On-chain, the Vault checks that pk_s and pk_r match the Registrar's records for the transacting parties, that $(C^{\text{in}}_1, C^{\text{in}}_2)$ equals the sender's current stored ciphertext, and that n equals the stored nonce; only then does it verify the proof. On success it *replaces* the sender's ciphertext with $C^{\text{out,s}}$, increments the nonce, homomorphically *adds* $C^{\text{out,r}}$ to the receiver's ciphertext, and emits C^{aud} in the transaction's event log for off-chain audit decryption. The asymmetry between replace and add is what lets a sender spend without knowing anything about the receiver's balance.

4.3 Withdrawal

STATEMENT · WITHDRAW (≈14,400 CONSTRAINTS)

Public: v, pk_s, $(C^{\text{in}}_1, C^{\text{in}}_2)$, n, output $C^{\text{out,s}}$ (12 field elements). Private: sk, b, r_1.

1. Key ownership and balance opening as above.
2. 40-bit range checks on b, v, $b - v$.
3. $C^{\text{out,s}} = \text{Enc}(pk_s, b - v; r_1)$.

The amount is public here because the tokens leave the shielded pool as ordinary ERC-20s; the proof's role is to show the withdrawal is affordable and that the replacement balance is correctly formed.

4.4 Replay and state binding

Every proof is single-use by construction. The per-account nonce appears as a public input and is incremented on every state change, so a verified proof can never be submitted twice. Ciphertext binding means a proof built against any earlier balance state is rejected as stale; the client library serializes an account's outgoing operations and retries on the rare rejection (Section 7.5).

5 The Denomination Layer

Bounded-exponent decryption fixes the encrypted amount domain at 40 bits: a balance must satisfy $b < 2^{40}$ or its owner could neither decrypt it interactively nor satisfy the circuits' range decompositions. Six-decimal assets fit this domain natively (2^{40} base units $\approx$ 1.1 million whole tokens), but an 18-decimal asset overflows it with a fraction of one token. veil resolves this with a per-asset **scale**, fixed immutably when an asset is admitted:

$$1 \text{ encrypted unit} \quad = \quad \textit{scale} \text{ base units.}$$

Every encrypted quantity, balance, transfer amount, and cap, is denominated in encrypted units; only the two ERC-20 legs multiply by *scale*, with deposits pulling $v \cdot \textit{scale}$ base units and withdrawals paying the same. Three properties follow.

- **The circuits are untouched.** Scaling lives entirely in the Vault's ERC-20 accounting; the proof system continues to operate on 40-bit integers.
- **Rounding dust cannot exist.** Deposits are denominated in encrypted units, so sub-scale amounts are unrepresentable rather than truncated: the vault only ever moves exact multiples of *scale*, and the encrypted credit always equals the tokens received.
- **Balances stay decryptable by construction.** Each asset carries a deposit cap constrained on-chain to $(0, 2^{40}-1]$ encrypted units. Since total shielded supply never exceeds the cap, no individual balance can ever leave the provable, decryptable domain. An uncapped mode is deliberately impossible.

Table 1. Representative scale assignments across supported asset classes.

Asset class	Decimals	Scale	Encrypted unit	40-bit ceiling
Stablecoin (USDG)	6	1	10^{-6} token	≈1.10 M tokens
Wrapped ether (WETH)	18	10^{14}	10^{-4} WETH	≈1.10×10^{8} WETH
Fixed-supply launchpad asset (10^{9} supply)	18	10^{15}	10^{-3} token	≈1.10×10^{9} tokens (full supply)
Tokenized equity (≈2×10^{4} shares)	18	10^{11}	10^{-7} share	≈1.10×10^{5} shares

For fixed-supply assets the scale is chosen mechanically as the smallest power of ten whose ceiling covers the entire supply; for open-supply assets it is a precision choice, with the deposit cap serving as the operative exposure bound. Tokenized equities under ERC-8056 [8] are a particularly clean fit: corporate actions such as dividends and splits adjust an on-chain multiplier rather than rebasing balances, so the

vault's conservation accounting, which assumes static token balances, holds across corporate actions by standard design.

6 Compliance and Auditability

Confidential finance on a regulated asset chain is viable only if confidentiality is bounded by accountability. veil builds the boundary into the transfer statement itself: condition 6 of Section 4.2 forces every transfer to carry an encryption of its exact amount to a protocol-level audit key, proven consistent with the amount the receiver is credited. The audit ciphertext is emitted in the event log of the very transaction it describes, so oversight requires no cooperation from either counterparty, no separate reporting channel, and no delay.

The audit key's power is deliberately asymmetric. Together with the inherently public entry and exit amounts, transfer-amount visibility gives the key holder a complete amount-level view of activity inside the pool. It confers no spending power of any kind: the key cannot move, freeze, or mint funds, and the protocol grants its holder no transactional role. Compromise of the audit key is therefore a confidentiality event, never a solvency event (Section 7.4).

veil operates strictly downstream of asset-issuer authority. Supported assets retain whatever controls their issuers built; the vault neither bypasses nor amplifies them. Paxos retains its standard freeze authority over USDG, and Robinhood's tokenized equities remain issuer-upgradeable instruments. These retained authorities are part of the protocol's stated trust surface (Section 7.6) and are disclosed in the user application alongside each asset.

7 Security Model and Trust Assumptions

7.1 What is hidden and what is not

veil provides *confidentiality*, not anonymity. Hidden from all parties except the counterparties and the audit key holder: transfer amounts and account balances. Public on the chain: the existence and timing of every operation, the addresses of sender and receiver, and the amounts of every shield and unshield. The protocol conceals how much moved, never that something moved; its privacy is Zether-class confidentiality, deliberately distinct from mixer-class unlinkability. The effective privacy of an exit additionally depends on pool activity between entry and exit, and is weakest when activity is low.

7.2 Soundness and external review

Solvency of the pool reduces to the soundness of Groth16 and the correctness of three circuits totaling roughly 48,000 constraints. The circuits implement a published construction and are exercised by an adversarial test suite (tampered outputs, replayed proofs, stale state, malformed keys, all rejected on-chain). Independent external review is a v1 roadmap item (Section 9); per standard practice for early-stage proof systems, protocol-enforced deposit caps bound the total value in the pool until it completes (10,000 USDG at mainnet launch), and the cap schedule follows the review schedule.

7.3 Trusted setup

Groth16 requires a per-circuit parameter ceremony. veil v1 uses the community Perpetual Powers of Tau [6] for phase 1 and a team-executed phase 2 per circuit. Integrity of the phase-2 parameters is accord-

ingly a v1 trust assumption: its compromise would undermine proof soundness, with exposure bounded by the same deposit caps. A public multi-party phase-2 ceremony that retires the assumption is a roadmap item (Section 9).

7.4 Audit key custody

The audit key can decrypt every transfer amount, so its compromise is equivalent to publication of amount data, though never to loss of funds. In v1 the key is fixed at deployment and rotation requires redeployment; custody procedures, and eventually threshold custody, are roadmap items with the same priority class as the ceremony.

7.5 Concurrency and liveness

Because proofs bind to the sender's exact current ciphertext, any state change that lands between proving and inclusion, including an incoming transfer, invalidates an in-flight proof, which then simply reverts and is re-proven. This is a liveness cost, not a safety cost: no reordering or interleaving of operations can misdirect or destroy funds. A counterparty deliberately spamming an account with dust transfers can degrade its outgoing latency; it can never touch its balance.

7.6 Asset-level and platform trust

Shielded assets remain instruments of their issuers: a frozen or blocklisted vault address would freeze the corresponding pool, and issuer-upgradeable assets (including ERC-8056 equities, which are beacon proxies) could acquire transfer restrictions in the future. The protocol inherits, and its documentation discloses, the ordinary trust assumptions of the underlying chain: Robinhood Chain's sequencer and data availability, and the finality of its Arbitrum-Orbit stack.

7.7 Contract-level protections

- single-use proofs via nonce binding; stale-ciphertext rejection; registered-key binding of both counterparties and the audit key;
- deposit caps constrained on-chain to the decryptable domain $(0, 2^{40}-1]$, with no uncapped mode;
- exact-receipt conservation on deposits, rejecting fee-on-transfer and rebasing assets rather than mispricing them;
- immutable per-asset scale once set; reentrancy guards on all fund-moving paths; immutable core contracts with a minimal owner role (asset admission and caps only).

8 Implementation and Live Deployment

The circuits are written in circom 2.1 and proven with snarkjs; contracts are Solidity 0.8.28; proving and decryption run client-side in the browser against locally served artifacts, with no external service in the loop. Key derivation, encryption, and decryption in the client are byte-compatible with the circuit semantics by construction and by test. Decryption uses a baby-step giant-step table sized to the asset's cap rather than the full 40-bit domain, built once per session in the background.

Table 2. Measured operation profile (Robinhood Chain, September 2026). Gas figures are from live transactions; proving times are client-side measurements.

Operation	Constraints	Proof size	Gas (measured)	Client time (measured)
Register	≈4,100	256 B	308,722	≈0.2 s proving (in-browser)
Shield	none	—	395,907	no proof required
Confidential transfer	≈29,600	256 B	561,866	seconds-scale on multicore hardware; ≈25 s in a single-threaded test environment
Unshield	≈14,400	256 B	419,441	≈0.5 s proving (test environment)
Balance decryption	—	—	—	≈2.8 s first use (table build), ≈20 ms thereafter

veil is live on Robinhood Chain mainnet, with on-chain proof verification confirmed at deployment, and on the public testnet, where the complete lifecycle, shield, confidential transfer, and unshield with real proofs and correct decryption on both sides, is exercised end to end against the deployed contracts.

Table 3. Deployment registry. Mainnet chain ID 4663; testnet chain ID 46630. The canonical USDG address is published by the chain's asset registry. Identical addresses across networks result from deterministic contract addressing (CREATE) under a common deployer.

Contract	Mainnet	Testnet
Vault	`0x4FfCE0CEE82D7c9443e8dEA179Fc63d15Dd5552b`	`0x4E384486F1aB08d6f958367167dA2785C3bb79Da`
Registrar	`0xfE564C639652f8d9CCf94664Ff95B85e0238446b`	`0xfE564C639652f8d9CCf94664Ff95B85e0238446b`
Registration verifier	`0x63E918ea5af2dC2C7fa27C747440c1F0aE4620fc`	`0x63E918ea5af2dC2C7fa27C747440c1F0aE4620fc`
Transfer verifier	`0x227877644f0922211F1cbb2f81096FC9dF29B492`	`0x227877644f0922211F1cbb2f81096FC9dF29B492`
Withdraw verifier	`0x23cDBaB9B3259Be79aFCC72A99767A676Eb934e2`	`0x23cDBaB9B3259Be79aFCC72A99767A676Eb934e2`
USDG (asset, canonical)	`0x5fc5360D0400a0Fd4f2af552ADD042D716F1d168`	test assets only

At mainnet launch a single asset is admitted, the Paxos-issued Global Dollar (USDG, six decimals, scale 1), with a deposit cap of 10,000 USDG pending external audit. The testnet deployment additionally exercises wrapped ether at scale 10^{14}, fixed-supply launchpad-style assets at scale 10^{15}, and an ERC-8056-style equity mock at scale 10^{11}, demonstrating the full denomination layer of Section 5. veil is free software: the protocol is distributed under GPL-3.0, with first-party contracts additionally offered under MIT.

9 Roadmap

- **External circuit audit**, followed by staged raising of deposit caps; the cap schedule is the audit schedule.
- **Public phase-2 ceremony**, retiring the team-run setup assumption of Section 7.3.
- **Asset expansion** on mainnet in order of settledness: wrapped ether; fixed-supply launchpad assets via the automated admission pipeline; ERC-8056 tokenized equities, each subject to a transferability re-verification at admission time.
- **Audit-key hardening**: documented custody, then threshold custody.
- **Public source release and hosted application**, completing the GPL distribution obligations for end users.

10 References

1. T. ElGamal. *A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms.* IEEE Transactions on Information Theory, 1985.
2. J. Groth. *On the Size of Pairing-Based Non-interactive Arguments.* EUROCRYPT 2016.
3. B. Bünz, S. Agrawal, M. Zamani, D. Boneh. *Zether: Towards Privacy in a Smart Contract World.* Financial Cryptography and Data Security, 2020.
4. B. WhiteHat, J. Baylina, M. Bellés. *EIP-2494: Baby Jubjub Elliptic Curve.* Ethereum Improvement Proposals, 2020.
5. iden3. *circom: A Circuit Compiler for Zero-Knowledge Proofs*; *snarkjs.* Free software, GPL-3.0.
6. Privacy & Scaling Explorations. *Perpetual Powers of Tau.* Community multi-party ceremony (phase 1).
7. D. Shanks. *Class Number, a Theory of Factorization, and Genera.* Proceedings of Symposia in Pure Mathematics, 1971. (Baby-step giant-step.)
8. ERC-8056. *Tokenized Equity Standard* (on-chain corporate-action multiplier), as implemented by Robinhood Stock Tokens.
9. Robinhood Markets. *Robinhood Chain Documentation.* docs.robinhood.com/chain.

Legal Disclaimer

This whitepaper is provided for general informational purposes only. It describes the design and current deployment of open-source software and does not constitute, and shall not be construed as, an offer to sell or a solicitation of an offer to buy any security, financial instrument, or digital asset in any jurisdiction; investment, legal, accounting, or tax advice; or a recommendation or invitation to engage in any transaction. Nothing in this document constitutes an offer, promotion, or solicitation of any token or other digital asset.

The software described herein is experimental cryptographic software distributed free of charge under the GNU General Public License v3.0, WITHOUT WARRANTY OF ANY KIND, express or implied, including the warranties of merchantability and fitness for a particular purpose. As of the date of this document the protocol's circuits have not undergone independent external audit; deposit limits described in Section 7 exist for this reason. Use of the protocol involves material risks, including without limitation the risk of total loss of digital assets arising from defects in cryptographic circuits or smart contracts, compromise of setup parameters, failures of the underlying blockchain, and actions of asset issuers (including freezing, blocklisting, or contract upgrades) over which the protocol has no control. Digital assets are volatile, and the regulatory treatment of confidential-transfer technology is evolving and differs by jurisdiction; users are solely responsible for their own compliance with applicable law.

Statements in this document that are not historical fact, including the roadmap in Section 9, are forward-looking statements based on current intentions and assumptions; they involve uncertainty, are not commitments, and may change without notice. Amount-level transaction data within the protocol is decryptable by the holder of the protocol audit key as described in Sections 6 and 7.4. Asset names and trademarks (including USDG and Robinhood) are the property of their respective owners; their use here is descriptive and implies no affiliation with or endorsement of veil.

---

veil · Confidential Token Protocol · Whitepaper v1.0 · September 2026 · Distributed under GPL-3.0. Contract addresses in Table 3 are the sole authoritative identifiers of the deployment; verify addresses independently before interacting.