STYX PROTOCOL · WHITE PAPER

Styx v2: private payments of any amount on Solana, checked on chain by a hash-based proof

8,900 words · 39 min read · In English

CONTENTS

1. Summary for a Solana user

How to read the rest of this paper

2. Threat model: who sees what

2.1 The receiver

2.2 The payer

2.3 Anyone reading the chain

2.4 Our servers: the site, the worker and the relayer

2.5 The RPC provider (Helius today)

2.6 A device thief

2.7 A quantum adversary

3. Design

3.1 One value pool, notes and records

3.2 Deposits: one size at a time, at one fixed price

3.3 Payments: the proof, the relayer and the verifier

3.4 The network routes: Oblivious HTTP, sealed reads and Tor

3.5 The wallet: keys from 24 words, values in records

3.6 Paying a Styx address

3.7 Requests and bearer claim links

3.8 Your own withdrawals

3.9 Abuse limits that cost no SOL

3.10 v1 notes move into v2

4. Proofs and cryptography

4.1 The proof system

4.2 What runs on chain, and its compute

4.3 Security figures, and what this paper does not claim

4.4 Post-quantum scope, exactly

5. Privacy, as measured

5.1 Method

5.2 Pass 1 (2026-10-06): a link from our own test script

5.3 Pass 2 (2026-10-07, morning): links on the fixed flows

5.4 Pass 3 (2026-10-07, evening): no link between payer and payee

5.5 Accepted residuals, with their numbers

5.6 The operator caveat

5.7 What the passes do not show

6. Performance

7. Economics

8. v1 and v2 compared

9. Status, limits and open items

10. Access and reproducibility

White paper, v2 edition, 2026-10-07. Volta Team, https://styx.cash.

Status in one line. Styx v2 runs on Solana **devnet only**, at styx.cash since 2026-10-07, 10:11 Paris time. It has not been externally audited and it is not on mainnet. v1 stays deployed on devnet as a fallback and will not go to mainnet; its white paper of 2026-09-23 stays online. This paper describes v2 as it runs on devnet on the date above, what it hides and from whom as we measured it, and what it does not do. [W1, W2, W3] Sections 1, 2.4, 3.4, 3.10, 5.6 and 9 were updated on 2026-10-15 for the optional Tor route and that day's release. Sections 3.4 and 9 were updated again on 2026-10-18, for the faster initial sync of 2026-10-17 and the Tor Browser storage test of 2026-10-17.

Section 1 is written to be read on its own. Section 8 compares v1 and v2 in one table.

---

1. Summary for a Solana user

The problem. On Solana, anyone can read who paid whom, how much and when, forever. Pay someone from your main wallet and they, and everyone else, can read your balance and your history.

What v2 does. Styx v2 gives you a private balance. You fill it from your wallet with deposits of fixed sizes, from 0.005 to 5 SOL, and from it you pay any amount up to 10 SOL: to someone's Styx address, against their payment request, by a payment link that anyone holding it can claim without a wallet, or out to any Solana address. [W4, W5] Each payment is a Solana transaction that our relayer signs and pays for, so the chain shows the relayer, not your wallet. [W6] Your browser computes a proof, built from hash functions only, that the payment is backed by notes you own and that none of them is spent twice; a Solana program checks that proof in full before any SOL moves. [W7, W8] A payment inside the pool carries its amount only inside sealed records that the receiver can open; a withdrawal to a Solana address shows its amount and its destination, like any transfer. [W9, W10]

What it hides, and from whom, as measured. Three internal forensics passes attacked v2 on devnet, each with a harness that knows the truth and three independent attackers, from the viewpoints of the receiver, the payer and anyone reading the chain, and, from the second pass on, of our own servers together with the RPC provider. The third pass, on 2026-10-07, found no structural link, no record link and no single party able to join a payer to a payee, on the harness or on any attacker. [W11, W12] The two passes before it did find links; section 5 says what each one was and what changed. The third pass also lists residuals that the project accepts; section 5.5 states them with their numbers.

What it does not hide. Deposits are public: your wallet, the size and the time. [W13] Sends leave at once, so when few people deposited since your own deposits, a receiver who knows the amount and the minute can match your payment to your deposit by its time; the app says so before you send. [W14] When few people deposited at least the amount you pay, the receiver can find you by the amount alone; the app says that too. [W15] Whoever pays with a claim link sees when it is claimed. [W16] Styx runs both the site that relays your

traffic and the worker that submits your payment: the private route splits what each server sees, it does not hide you from Styx itself. [W17] People who already use Tor Browser can take an optional second route through Tor, which keeps their IP address from our servers too (section 3.4). [W63] On devnet, most of the crowd you hide in is our own test wallets. [W18]

Why it is different. The proof uses hash functions only (SHA-256 and Poseidon2), no elliptic curves and no secret setup ceremony, and a verifier program written for Solana checks every proof in full on chain, spread over several transactions. [W7, W19] Records for a Styx address are sealed with ML-KEM-768 combined with X25519. [W20] Your Solana wallet, its signatures and your deposits stay Ed25519, which a large quantum computer would break; section 4.4 sets out exactly what rests on which assumption. [W21]

Where it stands. Devnet only, not externally audited, no mainnet. [W1, W2] Subscriptions are parked for deeper work (founder decision, 2026-10-07): v2 is send and receive; subscriptions already held in the older pools keep running. [W22]

How fast. 30 payments from a funded balance through our hosted worker took a median of 20.0 s from the click to the landing (p90 33.0 s) on 2026-10-06, before the private route was on; one payment through the private route took 32.5 s on 2026-10-07. [W23, W24] Section 6 gives every measured figure with its run and date. Devnet SOL has no value: use v2 to try private payments, not to hold money.

How to read the rest of this paper

A tag such as [W12] points to row W12 of the v2 claims ledger, `docs/CLAIMS-v2.md`, which names the evidence for that fact: a test, a deployment record, a forensics verdict, a founder decision or a file of the code. A tag such as [K18] points to a row of the v1 claims ledger, `docs/CLAIMS.md`.

Evidence marked `internal:` lives outside the repository: the run records of the v2 work (the devnet deployment record, the mainnet budget, the production checklist, the forensics verdicts and the decision log). They are internal logs, not yet published, and available to reviewers on request. The repository itself has been private since 2026-10-02, so this paper links to no code host; reviewers get access on request (section 10).

Times written with a Z are UTC; Paris time is UTC plus two hours on these dates. Every measured figure names its run and its date. A figure computed from measurements, rather than measured itself, is marked *derived*. Anything we did not measure is marked as not measured. Founder decisions are cited as "point N" of the decision log.

2. Threat model: who sees what

We name seven observers. The table gives each in one line; the subsections say what each learns.

OBSERVER	IN ONE LINE
Receiver	sees the amount and the moment it was paid, never the payer's wallet on chain; can try to match them against public deposits (2.1)
Payer	knows what it paid and when; sees when a claim link it paid is claimed; can look for the receiver's later withdrawal (2.2)
Chain observer	sees every deposit, every pool transaction in one uniform shape, and every withdrawal's amount and destination (2.3)
Our servers (site, worker, relayer)	the site sees your IP address and not the payment; the worker sees the payment and not your IP address; Styx runs both (2.4)
RPC provider (Helius today)	sees our servers, not your browser (2.5)
Device thief	needs your browser storage and your wallet to open the copy kept on the device; the 24 words alone open everything (2.6)
Quantum adversary	breaks Ed25519 signatures; the proofs rest on hashes; record sealing combines ML-KEM-768 with X25519 (2.7)

2.1 The receiver

The receiver sees a payment arrive in its private balance with its amount and, roughly, its time. The pool transaction that carried it is signed and paid for by our relayer and publishes nullifiers, not the commitment of any deposit, so nothing on chain names the payer's deposit. [W6, W25] What the receiver can still try is matching: the amount and the time it knows, against the deposits everyone can see. When the pool is quiet, that can work. The project accepts two forms of it as residuals and states them before you send (section 5.5): timing, when few people deposited since the payer's deposits, and amount, when few people deposited at least the amount paid. [W14, W15] For a payment request, the receiver also knows when it created the request; if the payer's deposit follows it closely and few others deposit, the two can be matched, and the app warns the payer and offers to wait. [W26]

2.2 The payer

The payer knows the amount and the moment of its own payment. For a claim link, the payer can compute the claimed note's nullifier from its own copy of the link, so it sees the moment the link is redeemed, whoever redeems it; this is inherent to links and the app says so. [W16, W27] The payer's other tool is the receiver's later withdrawal to a Solana address, which shows an amount and a destination: section 3.8 describes the rules the wallet applies to a user's own withdrawals for that reason, and section 5.5 what remains.

2.3 Anyone reading the chain

- **Sees.** Every deposit: the depositing wallet, its size and its time. [W13] Every pool transaction: the relayer as signer and fee payer, the nullifiers, two new commitments, two sealed records of 1,344 bytes, and the checkpoint the proof names. [W6, W9, W25, W28] Every withdrawal's amount and destination, which are public inputs of the proof. [W10]
- **Does not see.** The amount of a payment inside the pool (its public withdrawal amount is zero) or which deposit a spend consumes. [W10, W25]
- **Shape.** Pass 3 found the chain layer uniform: one compute limit, one Ed25519 check per pool transaction, a constant statement shape, output records with uniform bytes, and payments, link payments and link redemptions in one compute band. [W29]

2.4 Our servers: the site, the worker and the relayer

On styx.cash, the site relays each payment as a sealed message (Oblivious HTTP): it sees your IP address and not the payment; our worker opens it and sees the payment and not your IP address. The worker holds the relayer key that submits to Solana. [W17, W30] The site also sees the IP address that submits each flow, by design. [W31] Both are run by Styx: the split holds only while the two sets of logs are never joined. [W32] While a flow runs, the worker stores the job body only, and nothing after; no stored byte carries the payer's IP address, its deposit wallet, its funder, the deposit commitment or the request, and the public status endpoint answers state, position, reason and signature only. [W33] Our servers never hold your keys: the proof is computed in your browser. [W34] In Tor Browser, route 2 (section 3.4) sends everything through our onion gateway instead, so neither the site nor the worker sees your IP address. [W135]

2.5 The RPC provider (Helius today)

Since 2026-10-07 the app reads the chain, sends your deposits and waits for your payments through our site; your browser opens no connection to Helius, which sees only our servers. [W35] Reads go sealed to our worker's read route, live since 2026-10-07 at about 18:25 Paris; when that route is unavailable, the site passes them on in clear and sees your IP address and what is read, and the line before you send says which. [W36] Your wallet

extension may contact its own servers when it shows you a transaction to sign. [W37] Before this change, until 2026-10-07 14:44Z, Helius could tie each web user's IP address to its deposit wallet, and the site's submission timing could tie that address to the payment (section 5.4). [W38]

2.6 A device thief

Your private balance's secret is kept in your browser, encrypted under a key that comes from your wallet's signature on a dedicated message. Whoever holds both your browser storage and your wallet can open it. The 24 words open everything on their own, on any device. [W39, W40]

2.7 A quantum adversary

Section 4.4.

---

3. Design

3.1 One value pool, notes and records

v2 has one pool for SOL, of any amount. Its unit is a **note**: a value and the secrets that own it, of at most 10 SOL each. The pool program `zk_shielded` keeps each note's commitment (a Poseidon2 fingerprint of four field elements) in a Merkle tree with room for 2^22 notes (4,194,304), and records every nullifier it has seen in a nullifier tree, refusing any nullifier twice. [W41, W42]

Every pool action is one **value join-split** (VJS): a single circuit that spends two input slots and creates two output notes, each with a **record** of 1,344 bytes that tells the owner of that output what it holds. A slot not in use carries a dummy input or a zero-value output, so a payment, a merge, a link, a redemption and a withdrawal all have the same shape on chain. [W43, W29] A record is a 16-byte tag, an ML-KEM-768 ciphertext of 1,088 bytes, an X25519 key of 32 bytes and a body of 208 bytes. A record not addressed to a Styx address still carries a real X-Wing encapsulation (ML-KEM-768 with X25519), to a throwaway key, because random bytes can be told apart from ML-KEM ciphertexts. [W9]

New commitments wait in a queue. A crank flushes them into the tree and posts a **checkpoint** every 750 slots; a proof names a checkpoint at most 2,250 slots old. A fresh deposit can be spent once the next checkpoint includes it: one every 750 slots is 179 s at the devnet slot time we measured on 2026-10-05 (238.4 ms). [W44, W45]

3.2 Deposits: one size at a time, at one fixed price

A deposit goes from your wallet into the pool in one of ten sizes, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2 and 5 SOL. It pays 1 % of the deposit to the Styx treasury, 469 lamports to the pool's fee pot, and the Solana fee. [W46, W47] Deposits are public: the wallet, the size and the time can be read by anyone. [W13]

One door per deposit. A wallet deposits one size at a time: within an hour of a deposit it waits, and a top-up that needs several sizes makes one now and the next ones 1 to 6 hours apart. A send that one deposit covers leaves after that deposit. [W48] The reason is measured: forensics pass 2 found one deposit split into 0.01 and 0.05 SOL from one wallet in the same second, which is unique in shape and published the amount deposited. [W49]

One fixed price. Every deposit sets the app's own compute limit (310,000 units) and price (1,000 micro-lamports per unit, so 310 lamports at the limit). Pass 3 found that one real wallet extension added its own compute price to every deposit it signed (fees of 66,246 to 66,264 lamports, against 5,000 for the 154 other deposits): a fingerprint of that user. The fix has been in production since 2026-10-07 at about 20:19 Paris; a live check with a real wallet extension is pending. [W50, W51]

3.3 Payments: the proof, the relayer and the verifier

Proof. Your browser builds the VJS proof: 173,664 bytes, always that length, plus the two records. Proving starts before you press Send. [W7, W52, W53]

Sealed hand-over. The browser sends the proof and the records, with an anonymous proof of work, as a sealed message through the site to our worker (section 3.4). [W30, W54]

Reservation. The worker's relayer opens the flow with one transaction that leases one of its proof buffers (176,416 bytes) and reserves on chain the nullifiers being spent, so no one else can submit them meanwhile. [W55]

Upload and verification. It writes the proof in 46 chunks of 3,840 bytes, then sends 11 verification transactions: 8 fragments of 6 query checks each, and 3 parts of a final phase. [W56]

Settlement. The pool transaction consumes the verified buffer, records the nullifiers, queues the two new commitments, and pays out the withdrawal amount if there is one. [W57]

Over 30 payments on 2026-10-05, a payment took 60.7 Solana transactions on average. [W58] Every pool transaction has one shape: a version-0 transaction with the pool's address lookup table (17 addresses), a compute-budget instruction, an Ed25519 signature check and the pool instruction, at a fee of 10,000 lamports. Every VJS names a one-time Ed25519 key and carries its signature, so a link redemption looks like any other payment. [W59, W28]

Sends leave at once. No send waits for a crowd (founder decision of 2026-10-07, point 82). Below 20 people since your own deposits, the app says in one line that the receiver could match the payment by its time, and it says when few people deposited at least the amount you pay (point 87). [W14, W15] A wallet may also submit its own proofs through a leased protocol-owned buffer, with a bond of 0.01 SOL per lease, instead of using our relayer. [W60]

3.4 The network routes: Oblivious HTTP, sealed reads and Tor

Submissions. The site is the Oblivious HTTP relay and the worker is the gateway; the site holds no gateway key, and the app pins the gateway's public key configuration at build time. [W30] Since 2026-10-07 (about 20:29 Paris), every sealed submission is padded to one size per method, after pass 3 found three distinct sizes that told an operator which kind of flow it was carrying; a live check that every submission now has one size is pending. [W61]

Reads. No user IP address reaches Helius (founder decision, point 86): every chain read of the app goes through our site, sealed to the worker's read route where the worker offers it, in clear through the site's read proxy otherwise, and the route line in the app says which. [W35, W36] In a live sample on 2026-10-07, after the change, 6 users made 0 requests to Helius or any other third party, and 4,575 sealed reads and no plain reads. [W62]

Two routes. Route 1 is the private route above: it is for everyone and needs nothing installed. Route 2 is Tor, optional, for people who already use Tor Browser; nobody is asked to use Tor. [W63]

Route 2, in Tor Browser. The page still loads from https://styx.cash, through a Tor exit (the last Tor relay, which reaches the site for you). The app recognises Tor Browser by its user agent (the name the browser announces) together with the UTC time zone, and treats the page as under Tor only once it reaches our onion address (a server address reachable only through Tor) within 15 s. Every API call then goes to the onion address, sealed as on route 1. [W134] Our onion gateway sees only Tor's local address, keeps no access log and passes the sealed requests on to the site: neither the site nor the worker sees your IP address, and Helius sees only our servers, as on route 1. [W135, W35] If the onion address does not answer within 15 s, the app falls back to route 1 through the Tor exit, so the site sees the exit's address, not yours. [W136] The gateway is an onion service only: it is not a Tor relay, an exit or a Tor client. The worker does not use Tor to reach Helius; it would add nothing, since Helius sees no user's IP address on either route. [W137] The app creates your keys from a signature of a Solana wallet extension, so a Tor Browser user needs one installed in Tor Browser; without one the app cannot be used. [W138]

Route 2, measured. These figures come from a Firefox that sent Tor Browser 15.0.25's user agent on the real Tor network, not from Tor Browser itself. The onion route was taken in 10 of 10 direct loads and 9 of 10 loads through obfs4 bridges (unlisted entry relays that disguise Tor traffic) on 2026-10-14, and in 6 of 6 loads on 2026-10-15; a page load took a median of 2.7 s direct and 16.7 s through bridges (2026-10-14, 10 loads each). [W139] On 2026-10-15,

on the production build a4ced058, 3 payments made 0 requests to the site's API, opened 0 WebSockets (long-lived connections to a server) and sent 0 unsealed chain reads, and every payment landed. The initial sync of a new device took the payer 447.8 to 805.7 s, and click to landing took 181.5 to 532.6 s; the slowest payment read the pool's whole history a second time after one failed read. [W140] Since the release of 2026-10-15 at about 06:00 Paris (e781a4aa), a failed read keeps the windows (blocks of the pool's public history) already on the device: only a window that the device's own check refuses is fetched again, and the whole history only when it truly disagrees with the chain. After the release, 7 pages measured on clearnet never read the whole history a second time; before it, that happened once in about 4.8 page-hours on clearnet (2026-10-15). [W142] Before a4ced058, each payment under Tor sent 3 to 12 sealed reads to the site through the Tor exit and tried to open WebSockets to the site, and the payer's device synced twice (2026-10-14). [W141]

Route 2, the initial sync. Through Tor, the initial sync of a new device took minutes on 2026-10-15 (above). [W140] Since the release of 2026-10-17 (eb0f35a3), a new device reads the pool's history in aligned ranges of 256 windows, then 16, then single windows, and checks every window of a range as it checks a single one: the history as it stood on 2026-10-17 at 17:55Z took 37 requests instead of about 6,217. On one computer, with our site and worker run locally and Tor's delays simulated from the 2026-10-14 measurements, a new device was ready in 19.0 to 25.8 s over 3 runs, while the same device reading single windows was not ready after 20 minutes (2026-10-17). The real Tor network has not been measured with ranges yet. [W149]

Route 2, what the device keeps. The app keeps your private balance's secret in the browser's storage (IndexedDB), encrypted under a key from your wallet's signature. We tested Tor Browser 15.0.25 itself on 2026-10-17, on Linux in a test container, not on Windows, quitting it cleanly. At its default settings ("Always use private browsing mode" on), quitting Tor Browser erases everything styx.cash stored in it, that encrypted copy included; closing only the Styx tab keeps it; New Identity erases it. With that setting off, Tor Browser keeps styx.cash's data on disk across restarts, in a profile folder named after styx.cash, until New Identity or until the site's data is cleared. [W147] When the app finds nothing stored, it opens a new, empty private balance. The old one comes back only from its 24 words, with "Restore my private balance", which the app offers while the new balance holds no funds and no requests. So a Tor Browser user must keep the 24 words. [W148]

3.5 The wallet: keys from 24 words, values in records

- **One secret, shown as 24 words.** A private balance has a 32-byte master secret drawn from the device's random generator and shown as 24 words (BIP-39, English). It is never derived from a wallet signature. [W39]
- **Plug and play.** Connecting asks for two signatures: one for your keys, as before, and one on a dedicated, domain-bound message that unlocks the copy kept encrypted in your browser. Neither sends a transaction or costs anything. The balance is created silently, syncs in the background, and the app reminds you to write the 24 words down. [W40, W64]

- **Values in records.** In this recoverable wallet, the value of each change note travels in its own record's body under a one-time pad derived from the balance's secret, and the payer of a request writes the amount under the request's pad. To anyone else a body is uniformly random. With these, the 24 words alone restore the balance: a words-only recovery is tested end to end. [W65, W66] A payment against a fixed-amount request is recoverable from the words only when the payer's app has this change. [W65]
- **Computational hiding.** A recoverable wallet's own records are sealed under keys derived from its random secret. That rests on the hash function: it is computational, not unconditional. [W67]
- **Open before mainnet.** A forensics pass on these record bodies is required before any mainnet launch (point 79). Pass 3 read 176 output records and found their bytes uniform with no pad reused; the requirement stays on the production checklist. [W68, W29]

3.6 Paying a Styx address

A Styx address is a static address you can share. It carries an X-Wing public key (ML-KEM-768 with X25519, 1,216 bytes) derived from your secret. A payment to it is an ordinary note whose secrets the payer draws fresh; its record's tag is a SHA3-256 hash of the shared secret and the payment's nullifier, and its body is encrypted with ChaCha20-Poly1305 under a key derived from that shared secret. Your words alone are enough to find such payments. [W20, W69] On chain, the record has the same 1,344 bytes as every other. [W9]

3.7 Requests and bearer claim links

- **Requests.** A request lets someone pay into your private balance, the amount set by you or by the payer. [W5] The requester knows when it created the request; if the payer's deposit follows closely and few others deposit, the two can be matched, so the app warns the payer and offers a Wait button. [W26]
- **Bearer links.** A claim link works like cash. The payer's wallet draws a one-time Ed25519 key, the claim note belongs to that key, and the link carries it: whoever holds the link redeems it, with no wallet and no SOL, and the relayer pays. The redeemer's own key never appears on chain. The app says that a link must be shared privately and that the payer sees when it is claimed. [W27, W70] Until someone redeems it, the payer's copy is a link like any other. [W70]
- **Measured.** One link was paid and redeemed on devnet on 2026-10-07, through a local worker: the payment landed 15.9 s after the click, and a holder with no Solana key at all redeemed it in 13.0 s. [W71] A link opened within about one checkpoint of its payment has to be retried; the claim page retries by itself. [W72]
- **Share links.** A request, a Styx address or a claim link is shared as an https link to styx.cash/pay; the payment data sits after the #, the part of an address the browser never sends to a server. [W73]

3.8 Your own withdrawals

You can withdraw to any Solana address, from 0.0016 SOL, on a 0.0001 SOL grid; a withdrawal publishes its amount and its destination. [W10, W74] A payment to someone else's Solana address is a send and leaves at once. [W75] A withdrawal to one of your own wallets (the one you connected, one you withdrew to before, or one that funded your deposits) keeps three protections, because the person who paid you knows the amount and the moment it paid: [W76]

- **Never alone in its window.** It waits until a number of withdrawals to other wallets, drawn once per received payment between 1 and 40, have landed since the latest payment you received. [W77]
- **At most about 10 minutes.** It leaves at the latest at a limit drawn at random, uniformly between 480 and 720 seconds after the click, then goes anyway (point 84). [W78]
- **A shifted amount.** By default the total is drawn uniformly on the grid between nothing and twice the amount asked (plus or minus 100 %), the extra taken from the rest of your private balance and any shortfall kept private, and redrawn whenever it would fall close to a payment you received. You can still withdraw the exact amount, after a warning. [W79]

Before a withdrawal to an address that is one of your own deposit wallets or a wallet that funded one, the app warns you and lets you cancel. If it could not read who funded one of your deposit wallets, it says so and asks you to confirm before any withdrawal to an address. [W80] What these rules buy, and what they leave, is measured in section 5.5.

3.9 Abuse limits that cost no SOL

Every submission carries an anonymous proof of work; the gate takes no client parameter. [W54, W33] The on-chain reservation of section 3.3 stops two submitters from racing for the same notes. The relayer has a budget with a stop-loss read from the chain. [W81] A flow cut by a network event after its opening transaction loses its fees so far and, if it never lands, its two reservation bonds: about 2.31 million lamports at most per flow (*derived*). [W82] A third party who learns a payment's nullifiers when its reservation lands can block those notes for up to 2,250 slots at a time, paying 0.002 SOL of bonds for each block; that is not free, and not prevented. [W83] How long a mid-range phone takes to solve the proof of work is not measured. [W84]

3.10 v1 notes move into v2

Notes held in the v1 pools of 0.1, 1 and 10 SOL count in your balance once the app has found them, and move into v2 when a send needs them. Since 2026-10-15 the app looks for them only when you ask ("Look for my old v1 funds"); our site can see that a device searched, not what it found. [W145] Notes of the closed 100, 500 and 1,000 SOL pools cannot move; the app's Advanced section keeps the v1 recovery tools. [W85] A note from a v1 pool with

almost no other depositors keeps that small crowd when it moves: pass 2 matched one migration to the only v1 0.1 SOL deposit in six weeks, a test setup it did not count as a link. [W86]

---

4. Proofs and cryptography

4.1 The proof system

The proof is a STARK. The statement ("I own notes in this tree worth this much, their nullifiers are these, and the outputs are these") is written as a table that must satisfy fixed rules; the table is padded with random values drawn from the device's random generator; the rules are checked at a random point (DEEP-ALI), and a low-degree test (FRI) checks the committed data. Every random challenge is recomputed from a SHA-256 transcript of the proof (Fiat-Shamir), and the proof's Merkle trees use SHA-256. Commitments, nullifiers and the pool's tree use Poseidon2 (width 12, over the Goldilocks field, with a digest of four field elements). [W19, W87]

One circuit (identifier 8) covers every pool action. Its proofs use a table of 4,096 rows with 61 constrained columns, 48 query positions checked in 8 fragments of 6, and 40 public inputs; every challenge is drawn uniformly from a cubic extension of the field, by rejection sampling. [W88] Two of these choices answer the two root findings of v1's internal audit: v1 packed each fingerprint into one field element, so two notes could share one and one deposit could be withdrawn twice (F03), and it drew its challenges from the base field, so a forger could re-roll them (F52). v2's fingerprint has four elements and its challenges come from the extension field. [W89, K2, K20, K21] That is the design answer. Whether it holds is for the external review (section 4.3).

4.2 What runs on chain, and its compute

Two programs run v2 on devnet: the pool program `zk_shielded` (`GbVM5yve…`), upgraded for v2 on 2026-10-04 and last on 2026-10-06, and the verifier `p01_stark_verifier_v2` (`DpqfTShw…`), deployed on 2026-10-04 and last upgraded on 2026-10-06. The deployment record names no program upgrade after 2026-10-06. [W90] The verifier was written for Solana, with no proving library at run time. [W8]

Solana allows a transaction at most 1,399,850 compute units, the cap the deployment record checks every measurement against. [W91] Measured on devnet over 14 flows after the last verifier upgrade (2026-10-06):

TRANSACTION	WORST COMPUTE UNITS
a verification fragment (8 per payment)	956,318

TRANSACTION	WORST COMPUTE UNITS
a part of the final phase (3 per payment)	830,451
the pool transaction, with the final check	238,630
the opening transaction (lease and reservations)	111,463

[W92] The project keeps a margin rule for a fragment: the limit divided by 1.46, or 958,801. The worst fragment measured is about 2,500 units under it: a thin margin for any future check. [W93] Pass 3 measured pool transactions of payments, link payments and redemptions between 228,371 and 239,483 units. [W29] Summed over all its transactions, one payment used 10,382,910 units (30 payments, 2026-10-05). [W58] A deposit used 303,150 units in the program's tests on 2026-10-04; a crank flush at most 1,245,666 live, and a checkpoint 7,453. [W94]

4.3 Security figures, and what this paper does not claim

The project's security calculator (`tools/security-levels`) computes soundness figures for v2's proof format, but the generated security document, `docs/SECURITY-LEVELS.md`, does not publish them: they stay candidates until the calculator is re-run on the circuit as built and the format is reviewed externally. This paper therefore quotes no v2 soundness figure. [W95]

Whether a v2 proof reveals anything beyond its statement is argued in the format's specification (section 12) and exercised by nine families of tests that passed on 2026-10-04. The specification itself requires an external review before any public sentence relies on that argument, and none has happened, so this paper does not rely on it. [W96]

4.4 Post-quantum scope, exactly

- **The proofs** use hash functions only (SHA-256, Poseidon2): no elliptic curve and no setup ceremony. For a quantum attacker on such proofs, the calculator counts half the classical figure (the rule known as CMS19). [W19, W97]
- **Records** carry an X-Wing key encapsulation (ML-KEM-768 combined with X25519) on every record, real or to a throwaway key. [W9, W20]
- **A recoverable wallet's own records** are sealed under keys derived with a hash from its random secret: that sealing is computational. [W67]
- **Ed25519 remains** for your Solana wallet and every signature it makes, deposits included; for the relayer's signatures; for the one-time key of a claim link, checked on chain; and for the wallet signature from which the key of your browser copy is derived. [W21, W27, W40] A large quantum computer, if one is built, could forge those signatures. Nothing in v2 changes that: Solana verifies Ed25519 signatures. [W21]

5. Privacy, as measured

5.1 Method

Each forensics pass ran on devnet, read over the Helius Developer plan. A harness that knows the ground truth (who paid whom, from which deposit, to which exit) scores every viewpoint; three independent attackers (one reading the transaction graph, one statistical, one looking for fingerprints) name their best guesses, and each guess is scored against the truth. A positive control, a link planted on purpose, must be found or a clean result does not count. [W98] From pass 2 on, the viewpoints are four: receiver to sender, sender to receiver, chain observer, and our servers with the RPC provider, from a network capture. [W99] A **link** is a structural path, a record that names a party, or one party able to join a payer and a payee. Small-crowd matches by timing on sends (point 82), by amount in a quiet pool (point 87) and by time on a user's own withdrawals (point 84) are accepted residuals: each pass reports them apart, with numbers. [W100]

5.2 Pass 1 (2026-10-06): a link from our own test script

Verdict: **link found**, in the observer view, and no structural link from the receiver's or the sender's view. The link was a direct transfer, not a path through the pool: our test script had swept an exit wallet back into the depositing wallet. [W101] The pass also reported small-crowd matches: 30 test payments in which only 9 wallets could have covered the amount; requests whose creation time fell within 20 seconds of the payer's deposit; and exits equal to the paid amount minus 0.0007 SOL, which named the payee's exit wallet. Claim links were no easier to match than immediate payments. [W102] What changed: the wallet warns before any transfer between your own wallets and the scripts no longer sweep (point 54); the crowd counts only depositors able to pay the amount (point 57); withdrawals are spread and their amounts no longer equal what was received (points 58 and 66); the request time is warned about (point 59); a claim no longer carries any key of the payee (points 60 and 63). [W103]

5.3 Pass 2 (2026-10-07, morning): links on the fixed flows

Verdict: **link found**, on flows made after the fixes above. [W104]

- **The guard's order.** At the time, the wallet held a send until 20 newer eligible deposits existed; so the held payer was always the oldest eligible deposit in its crowd, and all three attackers named it.
- **Withdrawal timing.** A payee's withdrawal was the only one in its window, so the payer could name its wallet by time.
- **A split deposit.** One deposit split into two sizes from one wallet in the same second published the amount.

- **A test artefact.** An observer trace ran through test notes that never spend.
- **Compute.** Two link redemptions used 11 to 13 more compute units than payments.

What changed: sends no longer wait for a crowd (point 82), which removes the guard's order; a wallet deposits one size at a time; a user's own withdrawal is never alone in its window and waits at most 8 to 12 minutes (section 3.8); and over 20 transactions of each kind on the real programs the compute bands of payments, links and redemptions overlap, so no program change was made. [W105]

5.4 Pass 3 (2026-10-07, evening): no link between payer and payee

Data: every v2 pool transaction from 08:11Z to 17:47Z on 2026-10-07, read over Helius: 1,222 pool transactions, 159 deposits, 88 pool payments; 64 cases scored against the truth, with a live sample on the production deployment from 17:17Z to 17:47Z. [W106]

Verdict: NO_LINK between payer and payee. No structural link, no record link and no single party joining a payer and a payee, in any viewpoint, on the harness or on any of the three attackers. [W11, W12] The attackers' correct guesses (graph 19 of 172, statistics 15 of 192, fingerprint 33 of 201) were all production-period flows named by timing, by amount feasibility or, for the fingerprint attacker in the operator view, by the RPC tie that was closed that afternoon; none of the flows made after the pass-2 fixes was named, the true sender ranking 5th to 111th in crowds of 27 to 127. From the sender's side, each attacker got 0 of 44 right. The positive control, a withdrawal back to its own deposit wallet, was found. [W12] The chain layer was uniform (section 2.3). [W29]

Tells it found, and their fixes:

TELL	FIX	STATE ON 2026-10-07
One wallet extension added its own compute price to every deposit (section 3.2)	every deposit sets the app's fixed price	in production; live check pending [W50, W51]
Until 14:44Z, Helius tied each web user's IP address to its deposit wallet	every read goes through our servers (section 3.4)	live since 15:28Z; sealed reads from the worker since 16:25Z [W38, W62]
Split deposits from one wallet in the same second (before the fix)	one size per deposit	no split deposit after the fix [W49]
Pass 2's guard order	sends no longer wait	the payer was never the earliest eligible deposit in 28 guarded sends [W105]
Three sizes of sealed submission, which told an operator the kind of flow	one padded size per method	in production; live check pending [W61]

5.5 Accepted residuals, with their numbers

RESIDUAL	RULING	WHAT PASS 3 MEASURED
A send right after the payer's own deposit	point 82	of 14 browser and live payments, the payer's deposit was the newest before the payment in 10 (82 to 378 s before it, with 1 to 2 deposits in the 10 minutes before); the attackers named 7 to 11 of the 14 in the receiver view and 8 to 14 in the observer view
The amount alone, in a quiet pool	point 87	a 1.3 SOL payment had one depositor in range able to cover it, the payer itself; over three days the attackers found 2 wallets able to, the harness's amount method 98
The time of a user's own withdrawal	point 84	one withdrawal took 556 s from click to landing with no other withdrawal in the hour; another, at the exact amount chosen after the warning, took 590 s with one other withdrawal in the hour, so its payer could name the wallet by time and amount
The moment a claim link is redeemed	point 63	the claimer's sync starts right after the link lands, and the payer sees the redemption
A deposit and a withdrawal to the same wallet	the user's choice, after a warning	seen on test wallets and on the founder's own wallet

[W107, W108, W109, W110, W111]

The rules for a user's own withdrawals have a measured cost. In a simulation of 300 runs, with the 8 to 12 minute limit, a withdrawal right after a payment is the next withdrawal after it in 226 runs with 60 other users, against 274 with no rule and 0 with the rule and no limit; with 200 other users, 98 of 300. [W112] The amount shift brings the payer's best guess of the withdrawing wallet, with 60 other users, from 86 to 19 correct out of 300, against 12 by chance, when the payee holds other private money; 37 against 15 by chance when it holds none. [W113] The app says what it cannot do: when you have no other private balance to mix a withdrawal with, it tells you that the person who paid you has fewer people to hide you among. [W114]

5.6 The operator caveat

The site and the worker are both ours. Oblivious HTTP splits what each one sees; the split holds only while their logs are never joined. Pass 2 recorded one such join: the site saw one IP address post both a claim redemption and the claimer's withdrawal, and with the payer's copy of the link that names the claimer's withdrawal wallet; the pass reported it rather than counting it, since it needs two parties. [W32, W31] Under route 2, in Tor Browser, the site sees our onion gateway's address instead of yours, so it cannot group requests by your IP address; Styx still runs the site, the worker and the gateway. [W63, W135]

5.7 What the passes do not show

Every crowd in pass 3, apart from two wallets of the project's founder, was made by us; how v2 covers real users in real traffic is untested. [W115] Some cases had low power: 2 payments against fixed-amount requests, no wallet restore on chain in the period, and one payee's withdrawal that had not landed when the verdict was written. [W115] Every result is statistical over small devnet crowds, and these crowd sizes say nothing about mainnet. [W116]

---

6. Performance

Every figure is a devnet measurement over the Helius Developer plan unless the row says otherwise; times run from the click to the landing that the wallet sees.

FLOW	N	RESULT	CONDITIONS	RUN, DATE
Payment from a funded balance, hosted worker	30	median 20.0 s, p90 33.0 s (min 17.6, max 33.6)	relayer on Railway at 0.25 vCPU; before the private route; 6 of 30 needed a second attempt, cause fixed after the run	I53, 2026-10-06 [W23]
Payment from a funded balance, hosted worker, through the private route	10	median 9.01 s, p90 10.16 s (min 8.34, max 10.37)	production e781a4aa, a headless browser on the open internet, proof done before the click; the same night on a4ced058: median 10.25 s, p90 12.70 s	speedstudy-after, 2026-10-15 [W146]
The same, worker on our machine	30	median 8.7 s, p90 11.6 s	worker on our PC over HTTP, proof of work on	I51, 2026-10-06 [W117]
Payment through the private route	1	32.5 s	production, Oblivious HTTP	p03-pay-ohttp, 2026-10-07 [W24]
Withdrawal to a fresh address	2	23.3 s and 24.4 s	in a browser	2026-10-07 [W118]
Payment from an empty balance, one deposit included	1	134 s	left 108 s after the click, once a checkpoint held its deposit; the receiver's app showed it 156 s after it landed	2026-10-07 [W119]
Claim link: payment, then redemption	1	15.9 s, then 13.0 s	local worker; the holder had no Solana key	2026-10-07 [W71]

FLOW	N	RESULT	CONDITIONS	RUN, DATE
Deposit of 0.2 SOL	40	median 888 ms, p90 1,159 ms	with the 1 % fee	run-live-2, 2026-10-04 [W120]
Proving a payment	30	median 3,087 ms	in a test process on our machine, not a browser; starts before the click	run-live-4m, 2026-10-05 [W53]
Deposit to spendable	derived	179 s	one checkpoint every 750 slots at the measured 238.4 ms per slot	2026-10-05 [W44]

Not in the records this paper draws on: proving in a browser on a phone, the proof of work on a mid-range phone, payments through the private route at scale, and anything on mainnet. [W84] Our latency goal was 5 s; it is a goal, not a gate, and no sample reached it. [W121]

---

7. Economics

What a user pays (devnet, set on 2026-10-04).

- **Deposit:** 1 % of the deposit to the Styx treasury, 469 lamports to the pool's fee pot, and the Solana fee: 5,000 lamports for the signature, plus 310 lamports for the fixed compute price since 2026-10-07 (*derived*). [W46, W47, W50]
- **Payment or withdrawal:** 0.00065 SOL from your private balance. The relayer pays the Solana fees. [W122]
- **Limits:** at most 10 SOL per payment; deposits in ten sizes from 0.005 to 5 SOL. [W4]
- The treasury rate is a setting of the pool's configuration, capped at 5 %, and changed only by a proposal followed, after a delay, by its application. [W123]

What the operator pays (measured on devnet, derived for mainnet). A payment cost the relayer 308,833 lamports of Solana fees on average over 30 payments (2026-10-05), and 308,432 over 169 flows (2026-10-06); the relayer receives 649,062 lamports of each payment's fee, so it nets about 340,229 lamports per payment (*derived*, before the uniform transaction shape of 2026-10-06, which costs it 5,000 lamports more per pool transaction). [W124, W28] The crank is not paid for its checkpoints (5,000 lamports each): at 400 ms per slot, an assumed mainnet speed, that is 1,440,000 lamports a day (*derived*), and relayer and crank together cover their costs from about 5 payments a day (*derived*). [W125]

The launch budget. A mainnet launch of v2 as built would lock about 16.43 SOL (*derived*): the two programs' rent (about 9.43 and 3.66 SOL), the pool's accounts (1.97 SOL), one

operator buffer (1.23 SOL), a relayer float (0.1 SOL) and the crank's funding (0.05 SOL); the deployer would hold about 22 SOL for the minutes of the deployment. [W126] That is the gross figure of this release, not a floor: a lean-down pass is planned before any mainnet launch, and its outcome is open. [W127] On devnet, the relayer's budget is 1,183,834,168 lamports with a stop-loss after a loss of 750,000,000. [W81]

Regulatory and licensing questions are open. Whether running the site, the worker and the relayer, taking a deposit fee and a payment fee, and offering bearer payment links brings licensing, money-transmission, sanctions-screening or record-keeping duties in any jurisdiction, and on what terms a partner or an exchange could receive withdrawals, has not been answered by counsel. This paper names these questions and does not answer them. [W128]

---

8. v1 and v2 compared

	V1 (WHITE PAPER OF 2026-09-23)	V2 (DEVNET SINCE 2026-10-07)
Amounts	fixed notes; the web app uses the 1 SOL pool [K5]	any amount up to 10 SOL per payment from one private balance; deposits in ten sizes from 0.005 to 5 SOL [W4]
Privacy model	by default the operator hands you an older note it deposited; it can recompute, spend and recognise every note it issues [K10, K14]	you deposit yourself, one size at a time; every spend names no deposit; our relayer submits it; the operator issues no notes [W25, W48, W6, W129]
What the chain shows	every deposit; a C7 spend names no commitment; a one-time fee payer funded by the operator's float [K9, K11, K48]	every deposit (wallet, size, time); pool transactions in one shape, signed by the relayer, with sealed records; a payment's amount stays inside its records; withdrawals show amount and destination [W13, W29, W10]
Proofs	hash-based STARK; a C7 proof is 79,405 bytes, verified in one transaction (1,083,301 units) [K53, K55]	hash-based STARK; 173,664 bytes, verified over 11 transactions, the heaviest 956,318 units [W52, W56, W92]
Root flaw	one-element fingerprints: one deposit can be withdrawn twice (F03, critical); base-field challenges allow forgery by re-rolling (F52) [K2, K20, K21]	four-element fingerprints and extension-field challenges, the design answer to both; soundness figures not yet published [W89, W95]

	V1 (WHITE PAPER OF 2026-09-23)	V2 (DEVNET SINCE 2026-10-07)
Keys and recovery	your notes come from one wallet signature: whoever gets the wallet key re-derives them; an issued note's only copy sits in the browser [K15, K35]	a random secret shown as 24 words, never derived from a wallet signature; two signatures at connect; the words alone restore the balance [W39, W40, W66]
Sealing	stealth addresses with X25519 and ML-KEM-768 keys derived from the wallet key [K38]	every record carries an X-Wing encapsulation; Styx address keys come from your secret [W9, W20]
Speed, measured	withdrawal median 32.3 s over 30 runs, test harness, warm cache, 2026-09-23 [K143]	payment median 20.0 s over 30 payments through the hosted worker, 2026-10-06; two browser withdrawals in 23.3 and 24.4 s, 2026-10-07 [W23, W118]
Fees	deposit 1 SOL plus 0.3 %; a withdrawal pays 0.995 SOL [K44, K8]	1 % of each deposit plus 469 lamports; 0.00065 SOL per payment or withdrawal; the relayer pays Solana fees [W46, W47, W122]
Status	devnet; internal audit, 79 findings, one critical; never mainnet; kept as the fallback [K1, K4, K18, W3]	devnet since 2026-10-07; not externally audited; no mainnet [W1, W2]

The speed rows measure different flows on different days and are not a like-for-like race.

---

9. Status, limits and open items

Status. Devnet only, since 2026-10-07. Not externally audited: v2's evidence is internal (tests, targeted mutation runs, the three forensics passes). No mainnet deployment. v1 stays on devnet as the fallback. [W1, W2, W3, W130] Subscriptions are parked for deeper work (founder decision, 2026-10-07): v2 is send and receive; subscriptions already held in the older pools keep running. [W22]

Limits, in plain words.

- Timing and amount on sends, in quiet periods, as stated in section 5.5. The app warns; v2 does not protect you against these in a quiet pool. [W14, W15]
- The payer of a claim link sees when it is claimed. [W16]
- Styx runs the relay and the worker: route 1, the private route, does not hide you from Styx itself. Route 2, Tor for Tor Browser users, keeps your IP address from both, and Styx still runs the onion gateway, the site and the worker. [W17, W63, W135]

- Tor Browser needs a Solana wallet extension installed in it. Through Tor, the initial sync of a new device took minutes on 2026-10-15; the faster sync of 2026-10-17 was measured with Tor's delays simulated on one computer, not on the real Tor network. [W138, W140, W149]
- In Tor Browser at its default settings, quitting erases the private balance kept on the device; it comes back only from the 24 words (Tor Browser 15.0.25, tested 2026-10-17 on Linux, not on Windows). [W147, W148]
- A recoverable wallet's records are sealed computationally. [W67]
- After a restore from the 24 words, the wallet's checks see your own deposit wallets and past withdrawals again, but not the wallets that funded your deposits: those cannot be recovered, so the app asks you to confirm a withdrawal to any other wallet. [W143]
- A payment link made since 2026-10-15 can be taken back from the 24 words after it expires; links made before cannot. [W144]
- On devnet, most of the crowd is ours. [W18]

Open items (2026-10-07).

Live checks of the two last fixes: one deposit signed by a real wallet extension must match an injected test wallet's exactly, and every sealed submission must have one size. [W51, W61]

A forensics pass on the new record bodies, required before mainnet (point 79). [W68]

A withdrawal split into standard parts can still start up to 2 hours after the click; the 10-minute limit applies to the default path only. [W131]

The relayer's stop-loss does not yet count an open migration buffer; the code fix follows. [W132]

The verification fragment has about 2,500 compute units of margin left. [W93]

A third party can block a payment's notes for a while by paying bonds (section 3.9). [W83]

An external review of the proof format, its soundness figures and the argument of section 4.3, then an external audit of the programs and the clients. [W95, W96, W2]

A realistic crowd: forensics with independent users, which devnet does not give us. [W115]

The mainnet lean-down and the regulatory questions of section 7. [W127, W128]

---

10. Access and reproducibility

Since 2026-10-02 the repository is private and the code is not published. Reviewers get access to the repository and to the internal logs on request. [W133] Every transaction this paper cites is public on Solana devnet; the claims ledger gives the signatures and slots

where the records name them. The tests named in the ledger run from the repository; the deployment record gives the build hashes of the programs on devnet, which reviewers can rebuild and compare. [W90]