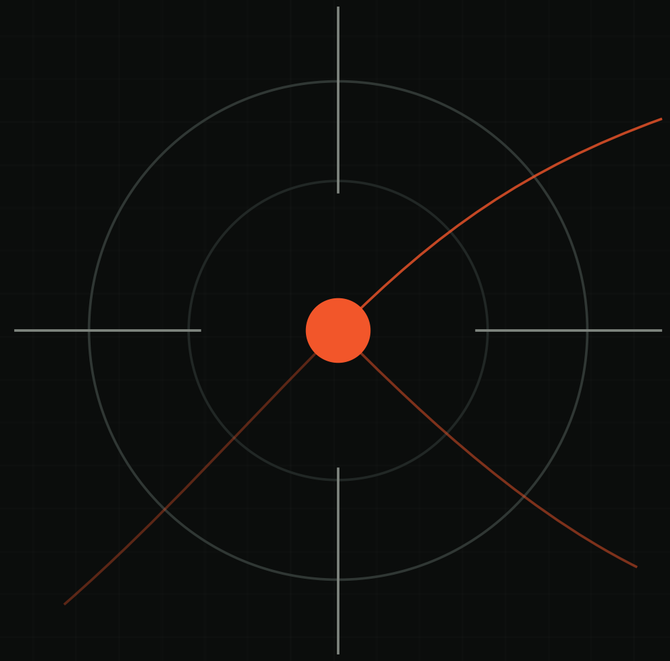
A PRIVACY LINTER & OBSERVATORY FOR ZCASH

Leakscope

The vault is safe.
The doors leak.



leakscope-cyan.vercel.app

github.com/AdityaWisnuu/leakscope

SHIELDED POOLS HIDE EVERYTHING, EXCEPT THEIR EDGES

Inside the pool you are invisible. *Crossing its edge* is public.

INSIDE A SHIELDED POOL

Hidden

- Sender and receiver
- Amount
- Memo

AT EVERY BOUNDARY

Published, exactly

- **Shield** t → pool: inputs, addresses, amount
- **Deshield** pool → t: address, amount, time
- **Pool crossing** (turnstile): the exact value moved

WHY NOW

Everyone must cross

- NU6.3 froze Orchard (withdraw-only) at block 3,428,143
- Every Orchard holder crosses a boundary at least once
- Wallets don't show users what that reveals

No cryptography is broken. This is what the transaction format reveals at the edges.

FIRST 75.6 DAYS OF IRONWOOD · PUBLIC CHAIN ONLY

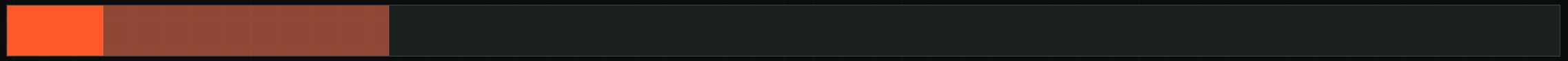
24.6%

of withdrawals since Ironwood launched *can be traced.*

20,343 / 82,581
deshielding transactions linked to where the money came from or went next

584,497
mainnet transactions indexed, blocks 3,428,143–3,515,250 (after a 30-day warm-up)

241,314
boundary events analysed



5,128 (6.2%) linked back to their deposit by exact amount + fee

15,215 (18.4%) more linked forward by a round trip

Source: web/public/data/summary.json · a lower bound: no network metadata, no off-chain data

MAINNET · VERIFIABLE ON ANY EXPLORER

Same coins in and out, *seven blocks apart.*

IN · SHIELDED INTO SAPLING · BLOCK 3,428,153

2.50121076 ZEC

c6b771a76de69246…845157fd

→ +7 BLOCKS

OUT · DESHIELDED FROM SAPLING · BLOCK 3,428,160

2.50106076 ZEC

57d7d945f70c8885…2e34fe3d

2.50121076 − 2.50106076 = **0.00015000 ZEC** the withdrawal's own 15,000-zat fee

The fee is public, so **deposit = withdrawal + its fee** pins the match to one exact value. No other deposit in the 30-day window fits: anonymity set of **1**. Leakscope's /check flags it as a high-severity quick round trip.

PRECISION MEASURED, NOT ASSUMED

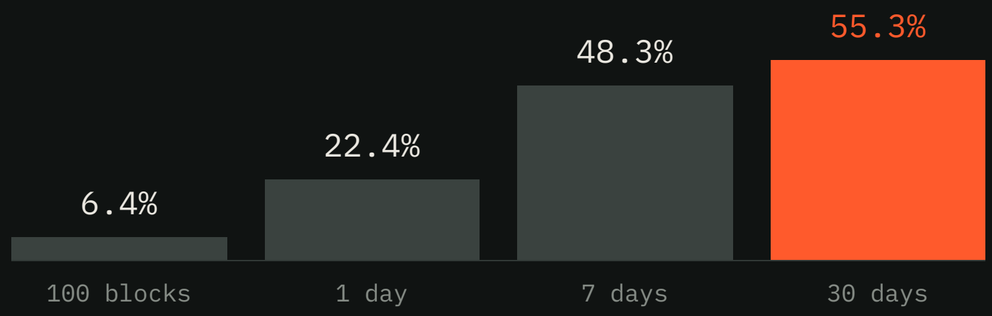
Run the rule *backwards in time.*

Money can't leave before it arrives. Matching each withdrawal against deposits made **after** it can only find coincidences, which gives the false-link rate on the chain's real amounts.

TIER WE REPORT	LINKS	COINCID.	PRECISION
Non-round, ≤100 blocks	2,535	1	≈99%
Non-round, ≤1 day	128	19	≈85%
Round, ≤10 blocks (small sample)	13	0	≈90%
~~Non-round, >1 day apart~~	~~447~~	~~603~~	~~dropped~~

More coincidences than links beyond a day, so those links are **not reported at all**. Round amounts >10 blocks apart: also dropped.

NAIVE RULE: “ONLY MATCHING DEPOSIT” · EST. FALSE LINKS BY WINDOW



At 30 days the naive rule is worse than a coin flip: 2,284 coincidences vs 4,133 links.

Source: web/public/data/evaluation.json · engine/scripts/evaluate.ts

75.6 DAYS · 584,497 TRANSACTIONS

The pools are busy. *The edges are not.*

15,668

Round trips: pool → t → pool

Re-shielding the very coins just deshielded links both sides deterministically. **7,370** within 10 blocks; spread over **17,326** addresses, so it is common behaviour, not one actor. **756** left frozen Orchard this way (~101k ZEC) instead of the turnstile.

3.27M *ZEC*

Orchard → Ironwood migrations, amounts public

24,725 migration transactions. **74%** used standard ZIP-318 denominations (good). **3,763** used an amount unique in its window: a fingerprint.

16,510

Address reuse

Shieldings from an address that earlier received a deshielding, across **1,442** addresses. **66.3%** come from one address, very likely a service rather than a person.

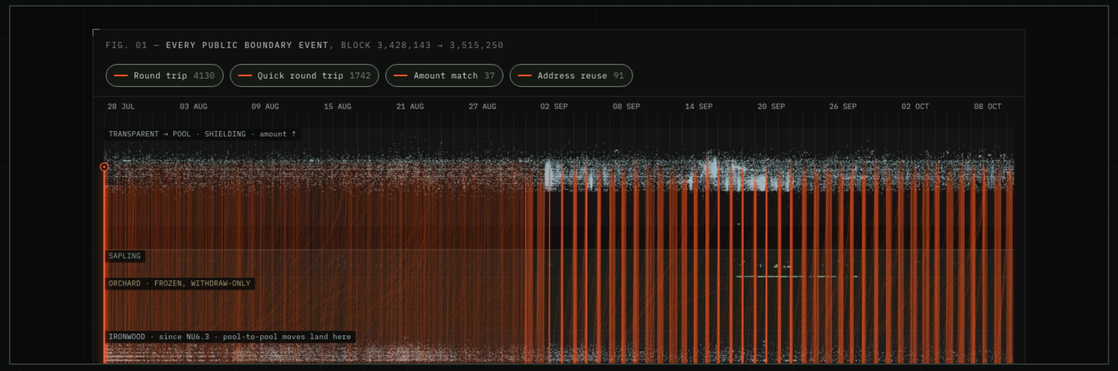
3 Sep *burst*

One address, ~19× normal shielding

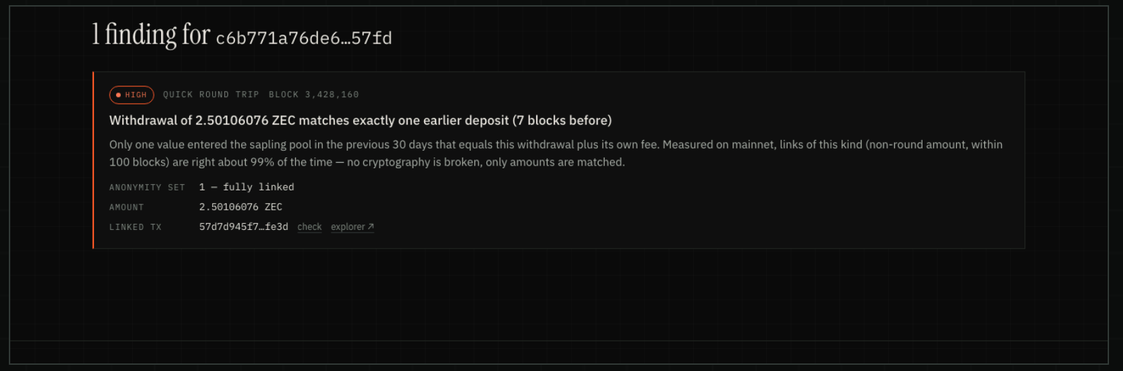
~5,500 shieldings into Ironwood in one day (blocks ~3,470,300–3,470,960); **1,548** of them (~3,000 ZEC) funded from a single transparent address. Visible on the observatory's daily chart.

Source: summary.json · docs/FINDINGS.md

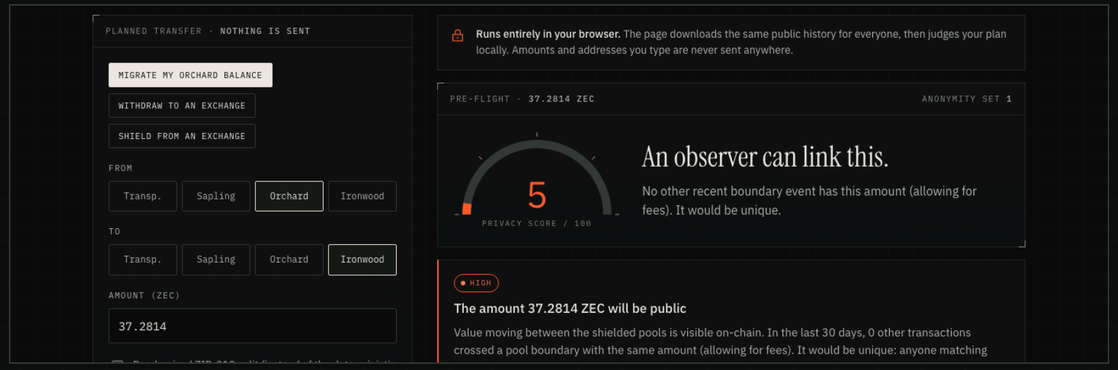
See it *before they do.* Four tools, all running in your browser.



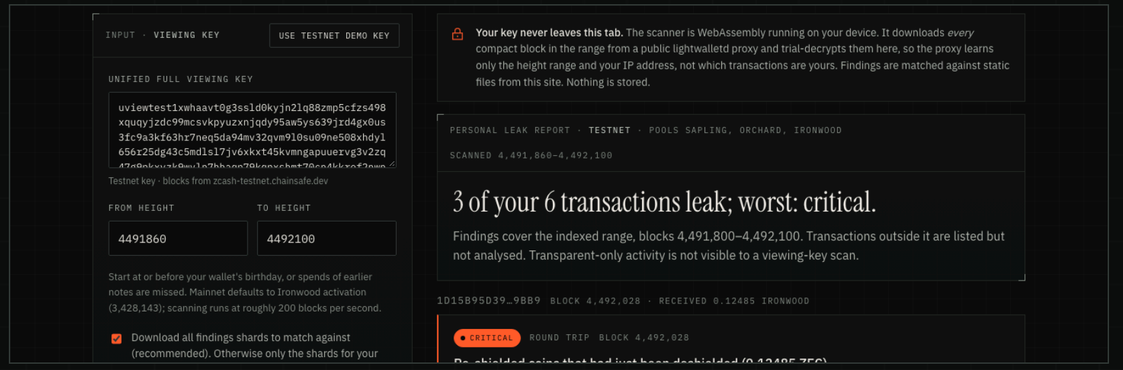
01 **Observatory** every boundary event since Ironwood; links drawn as threads



02 **Check** any txid; fetches only a 1/256 shard



03 **Plan** pre-flight score + ZIP-318 split with random delays



04 **Scan** viewing-key scanner in WASM; the key never leaves the tab

@LEAKSCOPE/ENGINE · PRE-FLIGHT SDK

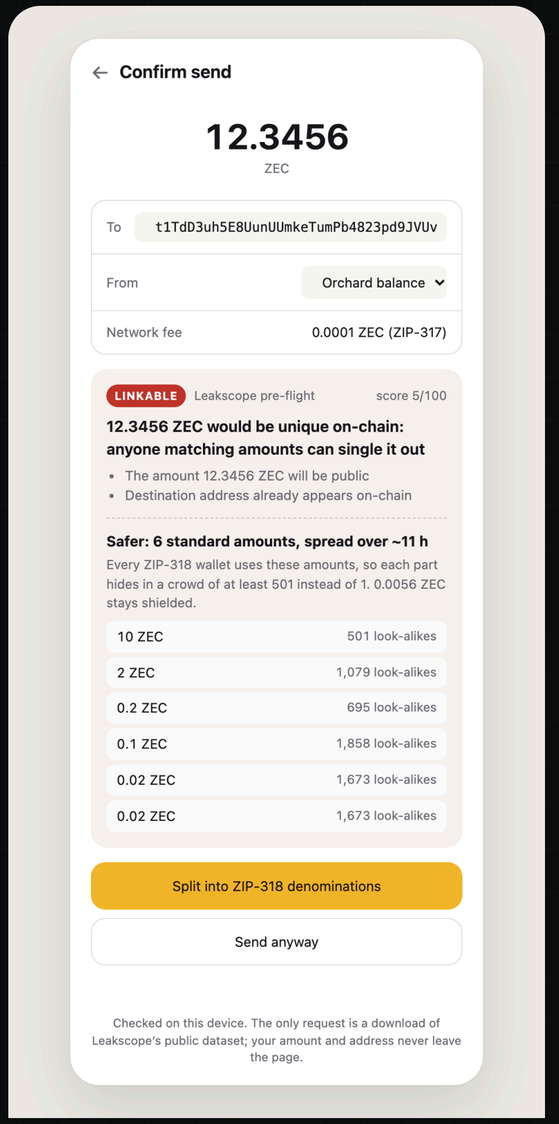
Lint the send *before Send.*

```ts
import { checkSend, loadPublishedContext } from '@leakscope/engine';

const ctx = await loadPublishedContext();  // public dataset, once
const v = checkSend({
  from: 'orchard', to: 'transparent',
  amountZat: 1_234_560_000, toAddress: 't1…',
}, ctx);

v.level;           // 'danger'   score 5 / 100
v.headline;        // '12.3456 ZEC would be unique on-chain: …'
v.suggestedSplit; // ZIP-318 parts + random broadcast schedule
```

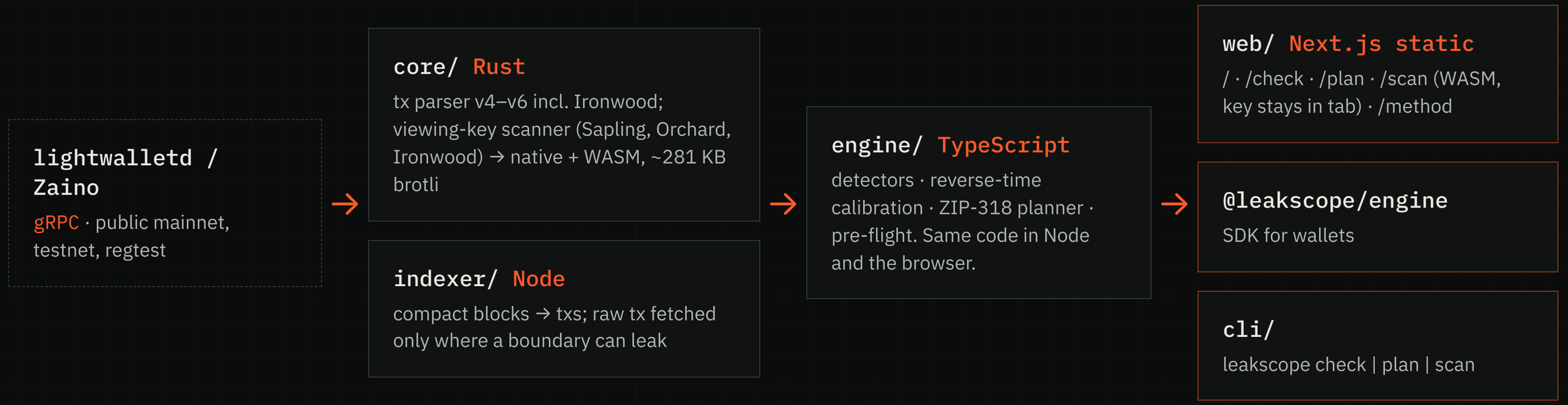
~32 KiB ESM, zero deps **Node + browser** **28** tests runs **on-device**



● examples/wallet-confirm · a “Confirm send” screen using checkSend()

OPEN SOURCE · MIT OR APACHE-2.0 · FULLY REPRODUCIBLE

One Rust core, *everywhere.*



TURNING “LEAKS ARE DISQUALIFYING” INTO A PRODUCT

Privacy fails at the edges. *Now it's measured.*

FOR USERS

Know before you send.

A score, the crowd your amount hides in, and a safer ZIP-318 plan, computed locally. Plain advice: don't deshield and re-shield the same coins; don't withdraw the note you just deposited.

FOR WALLETS

Warn at the confirm screen.

The same detectors that produced this report run as an embeddable SDK: refuse round trips, flag reused addresses, default to standard denominations and randomized delays.

FOR ZCASH

Evidence for ZIP-318.

First post-Ironwood measurement of boundary privacy. **74%** of migrations already use standard denominations; the other **26%**, and **3,763** unique amounts, show what wallets still need to fix.

Every number in this deck is reproducible from public chain data and open code: `report.ts` + `evaluate.ts`. Ground truth: our own deliberate testnet mistakes were caught, each leak class reproduced on regtest, mainnet examples spot-checked on an independent explorer.

GOALS, NOT ANNOUNCEMENTS

From a report to *a default.*

SHIPPED · OCT 2026

Live toolchain

Observatory, Check, Plan, Scan

Calibrated detectors + CLI

Engine SDK with wallet example

NEXT

Publish the SDK

`@leakscope/engine` on npm

PCZT input: lint the exact tx a wallet is about to sign

THEN

Into wallets

Target integrations: Zashi, Zingo and other Zcash wallets

Warnings on round trips, reuse, unique amounts

ONGOING

Stay current

Continuous indexing, live observatory

NU7 support and re-validation

Ironwood-only analytics: timing & arity inside the pool

- Integrations listed are targets we intend to pursue, not existing partnerships

TRY IT IN TWO MINUTES

Check the leak *before* you send.

LIVE	leakscope-cyan.vercel.app
CODE	github.com/AdityaWisnuu/leakscope
FINDINGS	docs/FINDINGS.md
VIDEO	demo video · link in the submission

