P

ProofVault v2.1

AKINDO · Build Privacy-First Apps on Midnight (Wave 3)

`Compact 0.28.0 · compactc 0.36.1` `Ledger 9 · zkir-3.2.1` `Apache-2.0`

MIDNIGHT SELECTIVE-DISCLOSURE TREASURY COMPLIANCE

Confidential Invoice Settlement & Selective-Disclosure Audit Receipts

DAOs and foundations prove **solvency**, **strict budget caps (`amount < cap`)**, and **bound non-sanctioned jurisdiction membership** in one Compact circuit — while keeping counterparty IDs, line items, exact amounts, country leaves, and treasury balances in local witnesses.

5 / 5	62 / 62	24 / 24	6 / 6 Live
COMPILED V2.1 CIRCUITS	VITEST + FUZZ TESTS PASS	JS GUARD MUTANTS KILLED	PREVIEW TXS (V2.0)

Web App: alphonshhhomenicmtu.github.io/proofvault-midnight/

Repo: github.com/alphonshhhomenicmtu/proofvault-midnight · Topic: midnightntwrk

Public Chains Leak Vendor Rates. Off-Chain Wires Break Solvency & Compliance Audits.

DAO finance teams and foundations pay security auditors, core developers, and legal counsel every epoch. Existing rails force a binary tradeoff between public leaks and opaque bookkeeping.

OPTION A · TRANSPARENT L1/L2 LEDGERS

Public Payouts Leak Competitive & Personal Data

- **Rate Exposure:** Competitors and negotiators inspect exact retainers paid to auditors, counsel, and liquidity partners.
- **Contributor Doxxing:** Public transfer graphs expose individual compensation and geographic distribution.
- **Runway Visibility:** Real-time exact treasury balances expose liquidity thresholds during governance votes.

OUTCOME: Finance teams move payouts off-chain.

OPTION B · PRIVATE OFF-CHAIN WIRES

Zero Cryptographic Compliance Proof for Tokenholders

- **No Solvency Floor Proof:** Governance cannot verify reserves stayed above policy minimums after disbursements.
- **No Jurisdiction or Cap Proof:** Tokenholders cannot check non-sanctioned allowlist membership or spend ceilings.
- **Slow Manual Audits:** External auditors reconcile PDFs and spreadsheets with no immutable receipt anchor.

OUTCOME: Governance loses independent compliance verification.

Cryptographic Hardening in `settleConfidentialInvoice()`

v2.1 closes dictionary nullifier attacks, unbound jurisdiction leaves, exact-amount cap leaks, and cosmetic tier labels directly inside `contracts/proofvault.compact`.

STEP 01
Auth & Active Vault
Checks `vaultStatus == Active` and `H(opSk) == treasuryAuthority`.

STEP 02
Open Balance Commit
Opens `persistentCommit(balance, currentSalt)` locally.

STEP 03
Strict Cap & Floor
Enforces strict `0 < amt < cap <= max` and `bal - amt >= floor`.

STEP 04
Tier Cap Rules
`Minimal` locks `cap == max`; `Threshold` enforces `cap < max`.

STEP 05
Bound Merkle Leaf
`jPath.leaf == jLeaf`, `checkRoot()`, and `jLeaf` inside invoice commit.

STEP 06
Keyed Nullifier
Blinds nullifier with secret `H(opSk)` — defeats dictionary grinding.

STEP 07
Fresh Salt & Receipt
Asserts `nextSalt != currentSalt`, re-blinds balance, writes receipt.

TIER 0 · MinimalSolvency

Circuit forces `budgetTierCap == policyMaxSingleSettlement` (hides narrow spend band) and zeroes `auditorTag = 0x00..00`.

TIER 1 · ThresholdBand

Circuit forces `amount < budgetTierCap < policyMaxSingleSettlement` for budget tracking with zeroed `auditorTag`.

TIER 2 · AuditorScoped

Checks `H("auditor-reg:v2", vkHash) == authorizedAuditorCommitment` before disclosing `auditorTag` MAC.

`proveTreasurySolvency()` also requires `operatorSecretKey()` and persists `lastSolvencyAttestation/Floor/Epoch` on-chain.

11 Private Witnesses vs. 21 Audited `disclose()` Call Sites

Compact 0.28.0 static taint tracking ensures private witnesses cannot enter public ledger state except through explicit, audited `disclose(...)` expressions.

Private Off-Chain Witnesses (11)

WITNESS-KEPT

- `operatorSecretKey()`
- `treasuryBlindingFactor()`
- `invoiceVendorId()`
- `invoiceAmount()`
- `jurisdictionLeaf()`
- `auditorViewingKeyHash()`
- `treasurySecretBalance()`
- `nextTreasuryBlindingFactor()`
- `invoiceLineItemsHash()`
- `invoiceSalt()`
- `jurisdictionMerklePath<8>()`

Public Midnight Ledger State

21 disclose() SITES

- `treasuryAuthority & authorizedAuditorCommitment`
- `treasuryBalanceCommitment (salted persistentCommit)`
- `lastSolvencyAttestation, lastSolvencyFloor, Epoch`
- `approvedJurisdictions: HistoricMerkleTree<8>`
- `usedInvoiceNullifiers: Set<Bytes<32>> (secret-keyed)`
- `auditReceipts: Map<Bytes<32>, AuditReceipt>`

SENSITIVE FIELD	PUBLIC LEDGER SEES	REGISTERED AUDITOR VERIFIES	WITHHELD FROM LEDGER?
Exact Treasury Balance	Salted commitment + persisted solvency floor	Committed balance ≥ reserve floor	YES
Exact Invoice Amount	Strict upper bound (`amount < cap`)	Tier-enforced cap & receipt immutability	YES
Counterparty & Scope	Salted commitment + secret-keyed nullifier	No dictionary grinding; unique anti-replay	YES
Counterparty ISO Country	Historic Merkle root only (leaf bound in commit)	Inclusion in public approved tree	YES (1-of-N)

Full line-by-line witness and disclose() audit committed in PRIVACY_BOUNDARY.md.

5 Hardened v2.1 Circuits Compiled Cleanly (`zkir-3.2.1`)

Command: `compact compile contracts/proofvault.compact contracts/managed/proofvault` → `Compiling 5 circuits: (exit 0)`.

EXPORTED COMPACT CIRCUIT	ZKIR	BZKIR	VERIFIER KEY	PROVER KEY	V2.1 RESPONSIBILITY
settleConfidentialInvoice()	24,040 B	7,445 B	2,121 B	86.0 MB	Strict `amt < cap`, tier rules, bound leaf, keyed nullifier, auditor reg, fresh salt
proveTreasurySolvency()	6,400 B	1,755 B	2,121 B	21.5 MB	Operator-authenticated solvency proof persisted into public ledger fields
registerApprovedJurisdiction()	3,433 B	964 B	2,121 B	10.8 MB	Operator-authenticated leaf append to `HistoricMerkleTree<8>`
updateGovernancePolicy()	3,606 B	1,053 B	2,121 B	10.8 MB	Max cap / reserve floor update, circuit-breaker pause & epoch increment
verifyReceiptOnChain()	969 B	266 B	1,353 B	0.2 MB	Public membership lookup on `auditReceipts` Map

5 PURE HELPER CIRCUITS

`deriveOperatorPublicKey`, `deriveAuditorCommitment`, `deriveNullifierBlindingSalt`, `deriveNullifier`, `deriveInvoiceCommitment`.

v2.1 ARTIFACT SHA-256

`proofvault.compact: fc0ac33e60d3c697...625f4a2a`

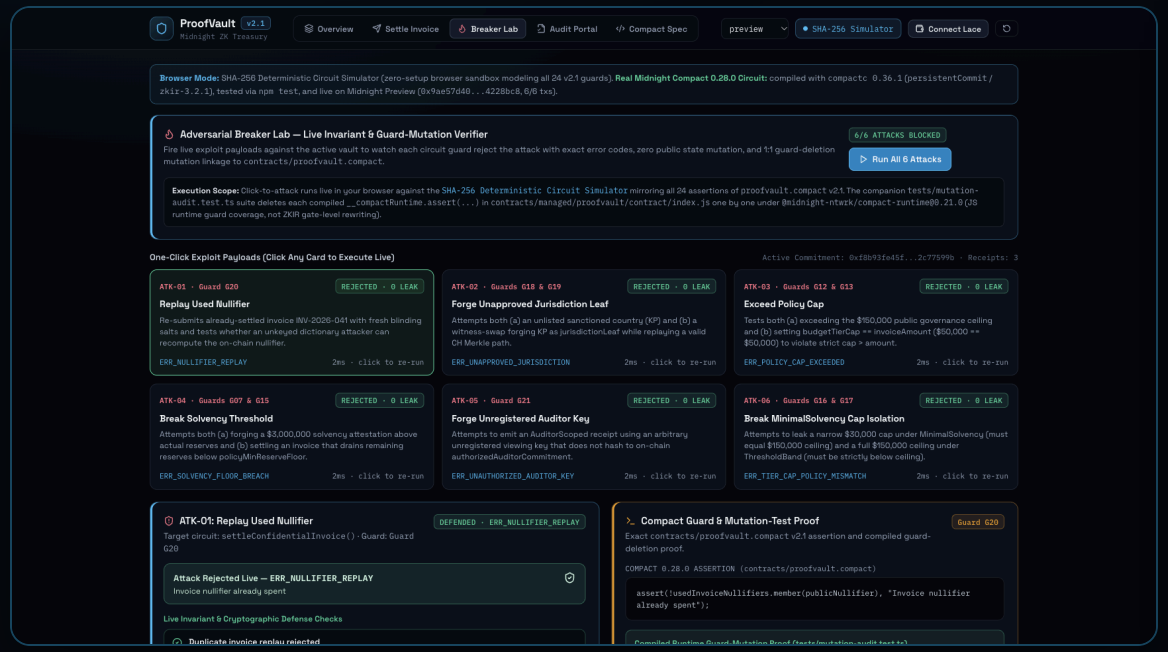
RUNTIME & CONNECTOR

`compact-runtime@0.21.0`
`dapp-connector-api@4.1.0`

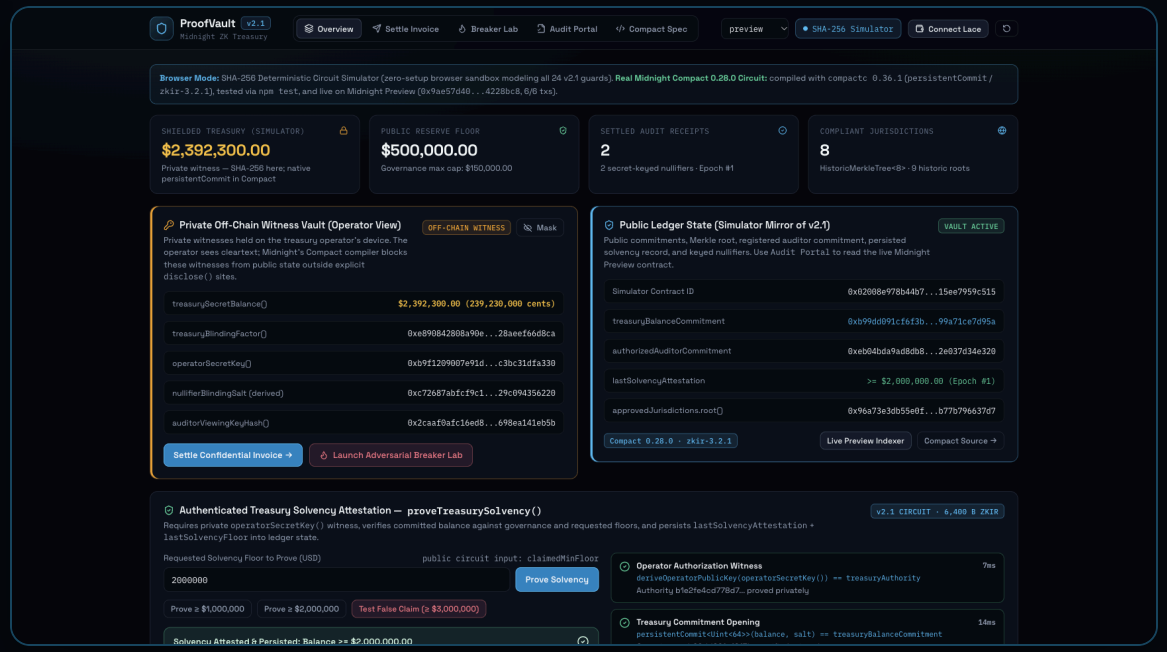
Compiled TypeScript bindings, ZKIR/BZKIR, and manifest committed under contracts/managed/proofvault/.

One-Click Adversarial Breaker Lab Built Directly Into the App

Judges test real exploit payloads in the browser SHA-256 Circuit Simulator and inspect exact error codes, zero public state mutation, and 1:1 guard-deletion mutation linkage.



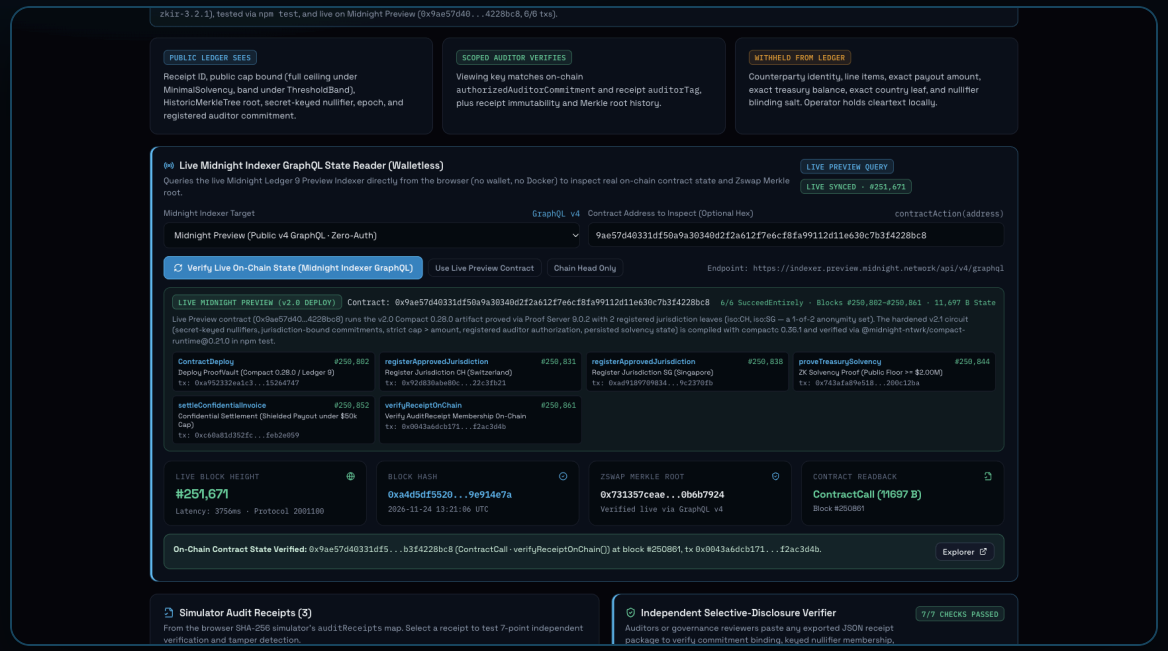
Adversarial Breaker Lab: 1-click buttons for *Replay Used Nullifier*, *Forge Unapproved Jurisdiction Leaf*, *Exceed Policy Cap*, *Break Solvency Threshold*, plus auditor-key and tier-isolation checks.



Dual-Ledger Overview: Explicit runtime clarity bar separating the Browser SHA-256 Simulator from the compiled Compact 0.28.0 contract, plus authenticated `proveTreasurySolvency()`.

Live Midnight Preview Deployment (6/6 Txs) + GraphQL v4 Reader

Inspect the deployed Midnight Preview Ledger 9 contract directly in the browser or via `npm run verify:onchain` — zero wallet and zero Docker required.



Live Midnight Preview Contract (v2.0)

Contract `0x9ae57d40331d...b3f4228bc8` verified live via `indexer.preview.midnight.network` at Block #251,597 (`11,697 B` on-chain state, `0xffa4a91ed51450...`).

6/6 Confirmed `SucceedEntirely` Transactions

`ContractDeploy` (`0xa952332ea1c3...`, #250,802) → `registerApprovedJurisdiction(CH/SG)` → `proveTreasurySolvency(≥$2M)` → `settleConfidentialInvoice(<$50k cap)` (`0xc60a81d352fc...`, #250,852) → `verifyReceiptOnChain` (#250,861).

Honest Wallet & Prover Separation

`@midnight-ntwrk/dapp-connector-api@4.1.0` supports `balanceUnsealedTransaction → submitTransaction` when paired with a Proof Server unsealed tx; static Pages uses `signData` strictly as a wallet connectivity check.

62/62 Vitest Passing · 24/24 Compiled JS Guard Mutants Killed

Tests exercise the real compiled `contracts/managed/proofvault/contract/index.js` via `@midnight-ntwrk/compact-runtime@0.21.0` alongside the browser SHA-256 simulator and Adversarial Breaker Lab.

Compiled JS Guard-Mutation Audit

24 / 24 KILLED

- Maps all 24 user-authored `assert(...)` guards in `proofvault.compact` v2.1 to compiled `__compactRuntime.assert(...)` calls in `index.js`.
- Deletes each single guard in memory and verifies the targeted exploit bypasses (scope: compiled JS interpreter guards, not ZKIR gate rewriting).
- Covers secret-keyed replay, leaf-swap, strict `amount < cap`, tier isolation, auditor registration, authenticated solvency, and salt freshness.

Property Fuzzing & Zero-Leak Audits

7 FUZZ SUITES

- **Compiled Runtime Fuzzing:** Random valid and invalid tuples across all 3 tiers assert exact receipt binding and zero mutation on failure.
- **Serialized State Privacy Scan:** Scans raw `ContractState.serialize()` bytes for operator key, nullifier salt, auditor vkHash, vendor, line items, and raw `Uint<64>` exact payout bytes.
- **Zero-PII JSON Export:** Strips `localMemo` and verifies single-field tamper detection across all receipt fields.

VITEST SUITE	RESULT	WHAT IT VERIFIES
`tests/mutation-audit.test.ts`	**32 PASS**	24/24 compiled JS guard-deletion mutants + guard inventory + 7 `fast-check` fuzz properties
`tests/compact-contract.test.ts`	**9 PASS**	Compiled v2.1 contract via `compact-runtime@0.21.0` (keyed nullifier, bound leaf, tiers, zero-leak)
`proof-engine (8) · indexer-client (7) · wallet-adapter (6)`	**21 PASS**	All 6 Breaker Lab attacks, 7-point receipt verifier, live GraphQL v4 client, Lace/1AM adapter

What the Circuit Enforces vs. What Remains Trusted Today

Clear boundaries beat marketing claims. Every actor role, trust assumption, and known v2.1 limitation is documented in `PRIVACY_BOUNDARY.md`.

ACTOR	ENFORCED BY COMPACT CIRCUIT	SEES	EXPLICIT BOUNDARY / OPEN LIMITATION
Treasury Operator	Solvency floor, strict `amt < cap`, tier rules, bound Merkle leaf, keyed nullifier, registered auditor, fresh salt.	Cleartext witnesses locally	Trusted to match off-chain bank/fiat wires until v3 Zswap shielded payouts; single authority key in v2.1.
Registered Auditor	Cannot claim receipts outside `authorizedAuditorCommitment`; verifies 7-point receipt immutability.	Scoped receipts + viewing key	Does not learn exact amounts or vendor names unless operator opens `invoiceCommitment` off-chain.
Public Observer	Verifies Ledger 9 state transitions, keyed nullifiers, persisted solvency floors, and historic roots walletlessly.	Public commitments & caps	Jurisdiction anonymity equals registered leaf count (2 on Preview v2.0; 8 in simulator); tree is append-only in v2.1.
Browser Simulator	Models all 24 v2.1 guards + Breaker Lab attacks deterministically in static browsers.	Local browser memory	Uses SHA-256, distinct from Compact's native `persistentCommit` in `npm test` & Preview.

DICTIONARY RESISTANCE

Nullifiers key on `H(operatorSecretKey)`, stopping public bounty-board preimage grinding.

SELF-HOSTED PROOF SERVER

Production witnesses must stay on localhost (`127.0.0.1:6300`) or operator-owned VPC.

NO FAKE INTEGRATIONS

Zero unshipped SDK claims; every circuit, indexer query, and wallet call runs against real Ledger 9 APIs.

Open-Core Treasury Compliance Layer for Midnight Mainnet

Built as an Apache-2.0 compliance contract paired with self-hosted prover tooling for DAO treasuries, foundations, and Web3 audit firms.

WHO USES IT

Target Teams

- **DAO & Foundation Treasuries** running grant, audit, and core-dev budgets with public governance floors.
- **Bounty & Contractor Ops** needing non-sanctioned jurisdiction proof without leaking vendor pricing.
- **Web3 CPA & Audit Firms** verifying scoped compliance receipts without collecting full vendor ledgers.

OPEN-CORE MODEL

Deployment & Revenue

- **Apache-2.0 Circuits:** Public Compact contracts and verification CLI maximize governance auditability.
- **Managed Adapters ($1.5k–$5k/mo):** Self-hosted VPC Proof Server packaging, Safe/ERP export connectors, and multi-auditor key lifecycle tooling.
- **First Wedge:** Cardano & Midnight ecosystem grant treasuries publishing verifiable milestone compliance.

V3 ROADMAP

Next Engineering Milestones

- **Live Preview v2.1 Redeploy & Preprod:** Broadcast hardened v2.1 contract to Preview and Blockfrost Preprod with full ISO allowlist seeded.
- **Zswap Shielded Payout Coupling:** Couple confidential settlement directly to native shielded transfers to remove off-chain wire trust.
- **Revocable Sanctions Tree & M-of-N Auth:** Replace append-only `HistoricMerkleTree` with a revocable epoch accumulator + threshold operator keys.

ProofVault v2.1 — Hardened Selective-Disclosure Treasury Compliance on Midnight

Web App: alphonshhhomenicmtu.github.io/proofvault-midnight/ · Video: youtube.com/watch?v=2SAttW3TaCU

MIDNIGHT WAVE 3