ANASTASIA LABS

VIA Midnight Smart Contracts Security Audit

Date: August 1, 2026
Project: VIA Cross-Chain Messaging Protocol — Midnight Contracts
Version: 1.3.0

Contents

Disclosure

This document contains proprietary information belonging to Anastasia Labs. Duplication, redistribution, or use, in whole or in part, in any form, requires explicit consent from Anastasia Labs.

Nonetheless, both the customer and Anastasia Labs are authorized to share this document with the public to demonstrate security compliance and transparency regarding the outcomes of the Protocol.

Disclaimer and Scope

A code review represents a snapshot in time, and the findings and recommendations presented in this report reflect the information gathered during the assessment period. It is important to note that any modifications made outside of this timeframe will not be captured in this report.

While diligent efforts have been made to uncover potential vulnerabilities, it is essential to recognize that this assessment may not uncover all potential security issues in the protocol.

It is imperative to understand that the findings and recommendations provided in this audit report should not be construed as investment advice.

Furthermore, it is strongly recommended that projects consider undergoing multiple independent audits and/or participating in bug bounty programs to increase their protocol security.

The original assessment covered revision `af3c1bac0d63a158c5c9c953b488cceebb1beef8`. The Code base section identifies remediation revision `e1c6099b83ca65296b67e1e71341a44bbb4a4fe0`, which Anastasia Labs subsequently reviewed against the retained findings. Scope is specifically limited to the 17 Compact source files under `contracts/`. TypeScript clients and operational tooling, generated contract runtimes and proof artifacts, local storage and secret management, and platform-level verifier maintenance are outside the report scope. The contracts were not live on mainnet during the engagement. No live Midnight deployment, production relayer, or destination-chain consumer was available for testing. The VIA team subsequently confirmed that the Cardano-side USDM lock-and-release contract rejects outgoing amounts above `2^64 - 1`; the remediation status of VIA-MID-001 relies on that source-side control.

Assessment overview

Anastasia Labs performed a security assessment of the VIA Midnight Compact contracts at the original pinned revision and a focused remediation review of commit `e1c6099b83ca65296b67e1e71341a44bbb4a4fe0`. The report scope contains 17 Compact source files under `contracts/`, organized into 18 validated security surfaces. Other repository components were excluded from reportable findings.

The work combined independent manual discovery passes, line-referenced threat modeling, Compact source/control/sink tracing, attack-path analysis, focused offline reproductions, and dedicated proof-of-concept models. Every reportable Compact issue received a separate technical write-up and remediation test plan.

Phases of code auditing activities included:

- **Planning** — pin the revision, inventory the codebase, define scope and trust assumptions.
- **Threat modeling** — identify custody, supply, message, replay, and on-contract role assets and boundaries.
- **Discovery** — perform independent category-based review across all Compact modules and entry points.
- **Validation** — reconcile source evidence, counterevidence, reachable attack paths, and safe local reproductions.
- **Reporting** — document confirmed findings, deferred questions, systemic hardening opportunities, and regression criteria.

The methodology was informed by official Midnight guidance that witness values are off-chain inputs and must be explicitly authorized, that possession of an own public key is not by itself access control, and that disclosure and state transitions require deliberate review. It also incorporated VIA's layered cross-chain validation model and the OWASP Smart Contract Security project's bridge guidance on message authenticity, replay resistance, integrity, and recovery.

Research references

- Midnight Compact language reference
- Midnight authorization guidance
- VIA Labs developer documentation
- OWASP Smart Contract Security bridge controls

Assessment components

Manual revision

Our manual code auditing is focused on a wide range of attack vectors, including but not limited to:

- Witness authentication and public/private boundary violations
- Owner, VIA-support, and relayer access-control bypass
- Cross-chain source-event authenticity and single-relayer trust
- Message type, version, source, destination, asset, and beneficiary binding
- Replay protection, domain separation, and transaction-identifier collisions
- Canonical amount parsing, zero-value messages, and integer-width truncation
- Custody-before-message ordering and route-specific serialization
- Fee configuration, collection authority, pause, and recovery behavior
- On-contract role rotation and subordinate-role recovery
- Hard-coded route, network, and asset identities

Executive summary

Protocol architecture

The reviewed system implements the Midnight side of VIA's cross-chain messaging and asset flows. `MessageGatewayV4` escrows and releases configured unshielded assets, while `USDM` burns or custodies value for outbound routes and mints value for inbound routes. Public users initiate outbound operations; authorized relayer witnesses submit inbound message facts. Owner and VIA-support witnesses administer endpoints, relayers, fees, pause state, and on-contract role changes.

Within the retained finding set, the dominant security boundaries are canonical cross-chain amount conversion and owner-controlled incident recovery.

Result

We confirmed **2 Compact findings**, both rated **Minor** under the Anastasia Labs template taxonomy. This corresponds to low severity. No Critical, Major, Medium, or Informational finding survived final scoped validation. Both retained findings have high evidence confidence.

The two retained findings are numbered sequentially as VIA-MID-001 and VIA-MID-002.

The first finding concerns noncanonical cross-chain amount conversion: Midnight reads only part of the 32-byte VILR field into a local `Uint<64>`. The Cardano-side USDM lock-and-release contract closes the reachable path by constraining every outgoing value to `Uint<64>` before message construction.

The second finding concerns incident recovery. VIA support must be able to pause USDM, but must not be able to unpause the system after an owner pause.

Remediation review

VIA-MID-001 is Remediated by the Cardano-side lock-and-release contract's `Uint<64>` outgoing-value bound. Although the Midnight decoder remains permissive, a valid Cardano-originated message cannot contain the noncanonical amount required by the finding.

VIA-MID-002 is Remediated by the final-fixes commit. USDM now passes the requested state to `isOwnerOrViaSupport`; VIA support may pause, while unpausing is owner-only.

The same commit changes the shared `isOwnerOrViaSupport` helper to require an `_enabled` argument. `USDM` forwards that argument, while `MessageGatewayV4` still invokes the helper with no arguments. Static Compact arity rules make this a release-blocking source integration error rather than an attacker-triggerable runtime security finding. It is therefore excluded from the finding count, but the gateway must be corrected and both contracts must compile cleanly before release.

Build and validation status

Safe offline parser and authority/state-machine models were executed for the deterministic Compact findings. The VIA team supplied the Cardano-side bound used to disposition VIA-MID-001; that destination-chain contract was outside this Compact audit and was not independently re-audited here. No live Midnight transaction was exercised. A compatible Compact compiler was not installed in the audit environment; the gateway arity defect was established statically from the exact declaration and all direct Compact callers.

Release recommendation

Both reported security findings are remediated. Before mainnet deployment, correct the gateway helper call and obtain clean exact-revision builds for both gateway and USDM. Re-test the Cardano amount bound and the owner-only USDM unpause flow in a disposable network.

Code base

Repository

https://github.com/VIA-Labs-Tech/vg1-midnight-contracts

Commit

e1c6099b83ca65296b67e1e71341a44bbb4a4fe0

Files audited

SHA256 Checksum	Files
c7f05a949a5da760f5c0df93896242cf23f0e5bad7165d87de8d9beac96e2c1f	scope-manifest.sha256 — 17 Compact files

Severity Classification

- **Critical**: This vulnerability has the potential to result in significant financial losses to the protocol. They often enable attackers to directly steal assets from contracts or users, or permanently lock funds within the contract.
- **Major**: Can lead to damage to the user or protocol, although the impact may be restricted to specific functionalities or temporal control. Attackers exploiting major vulnerabilities may cause harm or disrupt certain aspects of the protocol.
- **Medium**: May not directly result in financial losses, but they can temporarily impair the protocol's functionality. Examples include susceptibility to front-running attacks, which can undermine the integrity of transactions.
- **Minor**: Minor vulnerabilities do not typically result in financial losses or significant harm to users or the protocol. The attack vector may be inconsequential or the attacker's incentive to exploit it may be minimal.
- **Informational**: These findings do not pose immediate financial risks. These may include protocol optimizations, code style recommendations, alignment with naming conventions, overall contract design suggestions, and documentation discrepancies between the code and protocol specifications.

Finding severity ratings

The following table defines levels of severity and score range that are used throughout the document to assess vulnerability and risk impact.

Level	Severity	Findings
5	Critical	0
4	Major	0
3	Medium	0
2	Minor	2
1	Informational	0

Findings

ID	Title	Severity	Status
VIA-MID-001	USDM inbound parser ignores high bytes of the 32-byte VILR amount	Minor	Remediated
VIA-MID-002	VIA-support role can unpause USDM after an owner pause	Minor	Remediated

VIA-MID-001 USDM inbound parser ignores high bytes of the 32-byte VILR amount

Severity	Status
Minor	Remediated

Description

The inbound VILR layout allocates 32 bytes to the amount, but the Compact helper converts only the lower 16 bytes into the local `Uint<64>` amount and does not require the upper 16 bytes to be zero. Distinct 256-bit encodings can therefore produce the same local mint amount, and a source amount above the local range is not rejected based on the complete encoded field.

An authorized relayer or an accepted source message with a noncanonical amount can create accounting disagreement between source and destination. Replay identity is not derived from the full canonical payload, so another component may distinguish messages that USDM treats as the same amount or vice versa.

Affected code at the remediation revision: `contracts/HelpersCardano.compact:95-116` and `contracts/USDM.compact:209-227`.

Recommendation

Define the amount as one canonical big-endian `Uint<64>` on both chains. On Cardano, reject any source amount above `2^64 - 1` before constructing the message. On Midnight, require the first 24 bytes of the 32-byte VILR field to be zero and decode only the final eight bytes. Reject noncanonical encodings before replay mutation or minting. Use shared vectors for zero, one, `2^64 - 1`, `2^64`, and isolated nonzero high bytes across both implementations.

Resolution

Remediated. The Cardano-side USDM lock-and-release contract constrains every outgoing amount to `Uint<64>` and rejects values above `2^64 - 1` before a VIA message is constructed. Consequently, a valid Cardano-originated transfer cannot populate the ignored high bytes or reach the Midnight parser with an out-of-range amount. This source-side bound closes the identified cross-chain path. Adding the corresponding Midnight check remains advisable as defense in depth.

VIA-MID-002 VIA-support role can unpause USDM after an owner pause

Severity	Status
Minor	Remediated

Description

At the original audited revision, `USDM.setSystemEnabled` authorized either the owner or VIA-support witness without considering the requested state. VIA support could therefore set `_enabled` to `true` and restore processing after the owner had paused the system for incident response.

Affected code at the original revision: `contracts/ViaAccessControl.compact:52-57` and `contracts/USDM.compact:106-109`.

Recommendation

Remove unpausing capability from the VIA-support role. Permit VIA support to pause the system by setting `_enabled` to `false`, but require the owner witness whenever `_enabled` is `true`. Add regression tests proving that support can pause, cannot unpause, and cannot override an owner-initiated pause.

Resolution

Remediated. Commit `e1c6099b83ca65296b67e1e71341a44bbb4a4fe0` passes the requested state into `isOwnerOrViaSupport` and enforces `isOwner || (isViaSupport && !_enabled)`. VIA support can still pause USDM, but only the owner can unpause it.