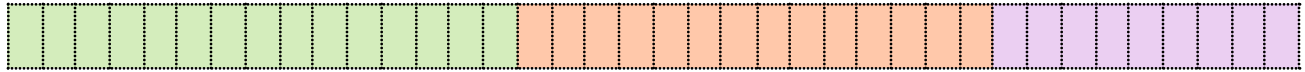


Status: **38 / 38 Resolved**



Fixed Won't Fix Tech Debt In Progress

# AggLayer v0.3.0 Audit Response

Version: 1.0 [DRAFT]

Authors: Dave Huseby , Simon Paitrault

Organization: Polygon AppSec

## Document History

- Mar 18, 2025 – Document Created
- Mar 21, 2025 – Audit 1 Findings Added
- May 14, 2025 – Status Updated, Audit 2 Findings Added
- May 27, 2025 – Status Updated
- Jun 10, 2025 – Submitted fix for 3.2-22.2, Status Updated
- Jul 18, 2025 – Completed

# Table of Contents

## [AggLayer v0.3.0 Audit Response](#)

### [Introduction](#)

### [Scope](#)

[Audit 1 Mar 10, 2025 – Mar 21, 2025](#)

[Audit 2 Mar 24, 2025 – Apr 4, 2025](#)

### [Detailed Findings – Audit 1](#)

[ALGO3-01 - Use of Unhashed Leaf Values in Merkle Tree](#)

[AGLO3-02 - Use of H::Digest::default\(\) for Empty Nodes May Break Non-Inclusion Logic](#)

[AGLO3-03 - Field Omission in L1 Leaf Hash](#)

[AGLO3-04 - Mistaken Network Identity in the Event of Integer Overflow](#)

[AGLO3-05 - Unchecked Use of unwrap\(\) May Cause Panics](#)

[AGLO3-06 - Unchecked TREE\\_DEPTH Values May Cause Compile-Time Panics and Logic Issues](#)

[AGLO3-07 - Potential Cross-Structs Keccak Hash Collisin](#)

[AGLO3-08.1 - Miscellaneous General Comments: Unclear Bit Indexing in Specification Comment](#)

[AGLO3-08.2 - Address TODO Comments](#)

[AGLO3-08.3 Miscellaneous General Comments: Improve Error Context When Deserializing Hex-Encoded Digest](#)

[AGLO3-08.4 - Miscellaneous General Comments: Deterministic Root Initialization](#)

[AGLO3-08.5 - Miscellaneous General Comments: Debug Formatting Inconsistency](#)

### [Detailed Findings - Audit 2](#)

[AGLO3.2-01 - Lack of Authentication/Authorization on RPC Services](#)

[AGLO3.2-02 - Network Tasks Can Drop Certificates if Channel is Full](#)

[AGLO3.2-03 - Unencrypted Communication for RPC and Metric Endpoints](#)

[AGLO3.2-04 - Blind Reproving of Proven Certificates Without State Recovery of Consistency Checks](#)

[AGLO3.2-05 - Missing Recovery for Failed Network Tasks](#)

[AGLO3.2-06 - Unchecked Panic with unwrap\(\) and expect\(\)](#)

[AGLO3.2-07 - Use of unwrap\(\) with gRPC reflection server can result in panic](#)

[AGLO3.2-08 - on\\_proven\\_certificate\(\) Continues on Partial Error](#)

[AGLO3.2-09 - insert\\_certificate\\_header\(\) Has no Conflict Detection for Settled Certificates](#)

[AGLO3.2-10 - No Panic Handling on Orchestrator or RPC Tasks](#)

[AGLO3.2-11 - Potential Resource leak on Panic in LocalExecutor](#)


[AGLO3.2-12 - Potential Resource Leak on Panic Before Shutdown](#)

[AGLO3.2-13 - write\\_batch\(\) Skips default write options](#)

[AGLO3.2-14 - Partial Error Handling in create\\_new\\_backup\(\)](#)  
[AGLO3.2-15 - Unimplemented GPU Prover Type](#)  
[AGLO3.2-16 - Fallback mechanism Cloning Overhead](#)  
[AGLO3.2-17 - Fallback Service Readiness Not Polled in Executor::poll\\_ready\(\)](#)  
[AGLO3.2-18 - No Resource Cleanup At Shutdown](#)  
[AGLO3.2-19 - Hardcoded Default Socket Addresses](#)  
[AGLO3.2-20 - LocalNetworkStateData State Risks Incorrect State Recorded on Error Conditions](#)  
[AGLO3.2-21 - Task Spawning Without Limits](#)  
[AGLO3.2-22.1 - Miscellaneous General Comments: Health Status Set Before Services Start](#)  
[AGLO3.2-22.2 - Miscellaneous General Comments: Blocking Calls Inside Tokio Runtime Context](#)  
[AGLO3.2-22.3 - Miscellaneous General Comments: Redundant Reflection Service Registration](#)  
[AGLO3.2-22.4 - Miscellaneous General Comments: Commented Out Code](#)  
[AGLO3.2-22.5 - Miscellaneous General Comments: Address TODOs](#)

# Introduction

This is a response to the two audit reports produced by Sigma Prime over the Aggregation Layer code.

 2025-04\_aggregation-layer\_d7b3dd1c28\_sigma-prime\_audit-part-1.pdf

## Scope

### Audit 1 Mar 10, 2025 – Mar 21, 2025

- <https://github.com/agglayer/agglayer/>
  - P0: crates/pessimistic-proof-program
  - P0: crates/pessimistic-proof-core
  - P1: crates/pessimistic-proof

### Audit 2 Mar 24, 2025 – Apr 4, 2025

- <https://github.com/agglayer/provers>
  - P0: crates/aggchain-proof-program
  - P1: crates/prover-engine
  - P1: crates/prover-executor

# Detailed Findings – Audit 1

## ALGO3-01 - Use of Unhashed Leaf Values in Merkle Tree

Severity: **Medium** ▾

Resolution: **Won't Fix** ▾

<https://github.com/agglayer/security/issues/49>

This approach has been deemed valid by design within a resource constrained, SP1-based zero-knowledge environment where hashing is expensive, tree depth is fixed and privacy is not required.

The position of each node is structurally determined by the key, so there is no ambiguity between leaf and internal nodes. The system does not rely on the cryptographic binding of the leaf value itself but instead on the overall integrity of the tree structure, which is preserved through secure hashing of internal nodes. This design avoids an extra hashing round at the leaf level, which is a practical optimization given the cost of hashing in SP1-based environments.

## AGLO3-02 - Use of `H::Digest::default()` for Empty Nodes May Break Non-Inclusion Logic

Severity: **Medium** ▾

Resolution: **Won't Fix** ▾

<https://github.com/agglayer/security/issues/51>

This design is correct as there is no semantic difference between an "unset" and "zero balance" leaf, they are the same as the chain starts with a 0 balance for each token anyways.

## AGLO3-03 - Field Omission in L1 Leaf Hash

Severity: **Low** ▾

Resolution: **Fixed** ▾

<https://github.com/agglayer/security/issues/56>

In [#725](#) the missing fields were added to the hash commitment and thus the full leaf state.

## AGLO3-04 - Mistaken Network Identity in the Event of Integer Overflow

Severity: **Low** ▾

Resolution: **Fixed** ▾

<https://github.com/agglayer/security/issues/55>

The fix is spread out over three different PRs: [#208](#), [#800](#), and [agglayer/interop #21](#). Essentially we're panicking on overflow in PP programs while also handling overflows gracefully in other contexts where panics must be avoided.

## AGLO3-05 - Unchecked Use of `unwrap()` May Cause Panics

Severity: **Low** ▾

Resolution: **Won't Fix** ▾

<https://github.com/agglayer/security/issues/52>

This is in the SP1 proof code, meaning that if the `unwrap()` triggers the worst that could happen is the proof fails to generate. It might still be an avenue for code quality improvement, that said.

For the `global_index`, the value's type is `U256` that guarantees the fact that we have the amount of bytes needed. There is a comment that explains why we can unwrap.

## AGLO3-06 - Unchecked `TREE_DEPTH` Values May Cause Compile-Time Panics and Logic Issues

Severity: **Informational** ▾

Resolution: **Won't Fix** ▾

<https://github.com/agglayer/security/issues/58>

Leo Gaspard : This is a const value, so if it were to overflow it'd result in a compile-time error. I think this is good enough?

Dave Huseby : Yes, compile-time failures are OK. In fact if there ever is an error, we prefer to catch it at compile time.

## AGLO3-07 - Potential Cross-Structs Keccak Hash Collision

Severity: **Informational**

Resolution: **Tech Debt**

<https://github.com/agglayer/security/issues/54>

This issue is noted and will be addressed as time allows. The team's assessment is that the fix could help with defense-in-depth but isn't an error.

## AGLO3-08.1 - Miscellaneous General Comments: Unclear Bit Indexing in Specification Comment

Severity: **Informational**

Resolution: **Fixed**

<https://github.com/agglayer/security/issues/53>

Leo Gaspard : I'm not sure I understand the issue here? The comment does define it as the 64th bit from the right; the comment is just MSB order to make it clear for the user (large values are always represented MSB) while the code refers to the bit in an LSB fashion

***SigP:** Makes sense, just a minor confusion when reading the comment/code. Perhaps could consider just a simple addition to the comment that the layout is described in MSB-first order, and that bit positions in code are interpreted in LSB-first order. Just for clarity for future contributors. No issue otherwise.*

This issue was addressed in [agglayer/interop #30](#) by adding a comment and renaming the constant to be `MAINNET_FLAG_LSB_OFFSET`

## AGLO3-08.2 - Address **TODO** Comments

Severity: **Informational**

Resolution: **Tech Debt**

<https://github.com/agglayer/security/issues/22>

Leo Gaspard : Unfortunately, I think we'll have to postpone that until later, unless you can see it becoming a vulnerability: some of them are very long-term planned improvements

### AGLO3-08.3 Miscellaneous General Comments: Improve Error Context When Deserializing Hex-Encoded Digest

Severity: **Informational**

Resolution: **Won't Fix**

<https://github.com/agglayer/security/issues/64>

This still needs to be addressed. The fix is a simple change of the `format!` string.

### AGLO3-08.4 - Miscellaneous General Comments: Deterministic Root Initialization

Severity: **Informational**

Resolution: **Won't Fix**

<https://github.com/agglayer/security/issues/60>

We recompute the roots for all trees at first and an empty root has semantics to refer to empty trees.

### AGLO3-08.5 - Miscellaneous General Comments: Debug Formatting Inconsistency

Severity: **Informational**

Resolution: **Won't Fix**

<https://github.com/agglayer/security/issues/90>

Adding the suggested `debug` wrapper does not bring a lot of value since we moved all of the `root` and `digest` values to newtypes. The existing `fmt::Debug` impl for `Digest` makes the output of the newtypes more readable.



# Detailed Findings - Audit 2

## AGLO3.2-01 - Lack of Authentication/Authorization on RPC Services

Severity: **Medium** ▾

Resolution: **Tech Debt** ▾

<https://github.com/agglayer/security/issues/92>

The team acknowledges that they need to add authentication/authorization to the RPC services. There is some design required to properly add it and that will be completed in the next version of AggLayer.

## AGLO3.2-02 - Network Tasks Can Drop Certificates if Channel is Full

Severity: **Medium** ▾

Resolution: **Won't Fix** ▾

<https://github.com/agglayer/security/issues/93>

No data is lost if the channel fills because the Certificates are already in the pending pool. Also, current refactoring will likely remove this channel altogether.

## AGLO3.2-03 - Unencrypted Communication for RPC and Metric Endpoints

Severity: **Low** ▾

Resolution: **Tech Debt** ▾

<https://github.com/agglayer/security/issues/94>

This is similar to the authentication/authorizing finding above. The team acknowledges that encryption needs to be added to RPC communications and it will be addressed in a future version of AggLayer.

## AGLO3.2-04 - Blind Reproving of Proven Certificates Without State Recovery of Consistency Checks

Severity: **Low** ▾

Resolution: **Tech Debt** ▾

<https://github.com/agglayer/security/issues/95>

This is an improvement, not a flaw, and will be addressed in the next round of AggLayer work.

## AGLO3.2-05 - Missing Recovery for Failed Network Tasks

Severity: **Low** ▾

Resolution: **Fixed** ▾

<https://github.com/agglayer/security/issues/96>

This issue was fixed in [#812](#). The code was restructured to remove the spawned network task if poll the future results in an `Err`.

## AGLO3.2-06 - Unchecked Panic with `unwrap()` and `expect()`

Severity: **Low** ▾

Resolution: **Won't Fix** ▾

<https://github.com/agglayer/security/issues/97>

Per this [comment](#), this issue won't be fixed because it is desired behavior in our recursive ZK proof generation design.

## AGLO3.2-07 - Use of `unwrap()` with gRPC reflection server can result in panic

Severity: **Low** ▾

Resolution: **Fixed** ▾

<https://github.com/agglayer/security/issues/98>

This issue is fixed in [agglayer/provers #216](#) that improves the error handling in the reflection build methods.

### AGLO3.2-08 - **on\_proven\_certificate()** Continues on Partial Error

Severity: **Low** ▾

Resolution: **Fixed** ▾

<https://github.com/agglayer/security/issues/99>

Addressed in [#819](#)

### AGLO3.2-09 - **insert\_certificate\_header()** Has no Conflict Detection for Settled Certificates

Severity: **Low** ▾

Resolution: **Fixed** ▾

<https://github.com/agglayer/security/issues/100>

Simon Paitrault : This has been mitigated by adding more complex checks on a higher level function that handle L1 communication and extra checks.

### AGLO3.2-10 - No Panic Handling on Orchestrator or RPC Tasks

Severity: **Low** ▾

Resolution: **Won't Fix** ▾

<https://github.com/agglayer/security/issues/101>

Leo Gaspard : I just checked, and even a single-thread tokio runtime will not cause a panic in a task to kill the process:

<https://play.rust-lang.org/?version=stable&mode=debug&edition=2024&gist=7f6695143e5a937c011498707387f019>

No panic will kill the process. Closing, as “won’t fix”.

### AGLO3.2-11 - Potential Resource leak on Panic in **LocalExecutor**

Severity: **Low** ▾

Resolution: **Won't Fix** ▾

<https://github.com/agglayer/security/issues/102>

Further testing showed this to not be a concern. The only possibility for resource leaking would be if there is manual resource clean-up code that won't be run if the task panics. There is no manual cleanup code so this isn't an issue.

### AGLO3.2-12 - Potential Resource Leak on Panic Before Shutdown

Severity: **Low** ▾

Resolution: **Fixed** ▾

<https://github.com/agglayer/security/issues/103>

This issue is fixed in [agglayer/provers #225](#). The cancellation token gets a **DropGuard** added now that will cancel the CancellationToken when it is dropped (i.e. during a stack unwind on panic)

### AGLO3.2-13 - **write\_batch()** Skips **default\_write\_options**

Severity: **Low** ▾

Resolution: **Tech Debt** ▾

<https://github.com/agglayer/security/issues/104>

This issue will be addressed in a future version of AggLayer.

### AGLO3.2-14 - Partial Error Handling in **create\_new\_backup()**

Severity: **Low** ▾

Resolution: **Tech Debt** ▾

<https://github.com/agglayer/security/issues/105>

This issue will be addressed in a future version of AggLayer.

### AGLO3.2-15 - Unimplemented GPU Prover Type

Severity: **Informational** ▾

Resolution: **Fixed** ▾

<https://github.com/agglayer/security/issues/106>

The **todo!** was changed to return an error instead of panicking.

## AGLO3.2-16 - Fallback mechanism Cloning Overhead

Severity: **Informational**

Resolution: **Won't Fix**

<https://github.com/agglayer/security/issues/107>

The cloning is unavoidable with the code as-is because the handle to the fallback service gets moved to the closure encapsulated by the Future returned from the `call` method.

## AGLO3.2-17 - Fallback Service Readiness Not Polled in `Executor::poll_ready()`

Severity: **Informational**

Resolution: **Fixed**

<https://github.com/agglayer/security/issues/108>

This issue is fixed in [agglayer/provers #222](#). It adds a check for the fallback's readiness.

## AGLO3.2-18 - No Resource Cleanup At Shutdown

Severity: **Informational**

Resolution: **Fixed**

<https://github.com/agglayer/security/issues/109>

This issue is fixed in [agglayer/provers #221](#). It adds configurable shutdown timeouts and ensures all resources are cleaned up.

## AGLO3.2-19 - Hardcoded Default Socket Addresses

Severity: **Informational**

Resolution: **Fixed**

<https://github.com/agglayer/security/issues/110>

This issue is fixed in [agglayer/provers #221](#). It moves the hard coded port values to be the defaults in the config.

## AGLO3.2-20 - LocalNetworkStateData State Risks Incorrect State Recorded on Error Conditions

Severity: **Informational**

Resolution: **Won't Fix**

<https://github.com/agglayer/security/issues/111>

The code operates on a clone of the data so that any interruption doesn't cause partial updates. No fix needed.

## AGLO3.2-21 - Task Spawning Without Limits

Severity: **Informational**

Resolution: **Tech Debt**

<https://github.com/agglayer/security/issues/112>

Currently the network ID is bounded by the L1 and isn't an issue that requires an immediate fix. This will be addressed in a future version of AggLayer.

## AGLO3.2-22.1 - Miscellaneous General Comments: Health Status Set Before Services Start

Severity: **Informational**

Resolution: **Won't Fix**

<https://github.com/agglayer/security/issues/113>

This isn't actually an issue because the health service is designed to only be exposed to the API and the other services are already read. This isn't providing invalid state.

## AGLO3.2-22.2 - Miscellaneous General Comments: Blocking Calls Inside Tokio Runtime Context

Severity: **Informational**

Resolution: **Fixed**

<https://github.com/agglayer/security/issues/114>

This issue is fixed in [agglayer/provers #235](#). It adds appropriate comments.

### AGLO3.2-22.3 - Miscellaneous General Comments: Redundant Reflection Service Registration

Severity: **Informational**

Resolution: **Fixed**

<https://github.com/agglayer/security/issues/115>

This issue is fixed in [agglayer/provers #216](#). It fixes the redundancy in the prover engine logic.

### AGLO3.2-22.4 - Miscellaneous General Comments: Commented Out Code

Severity: **Informational**

Resolution: **Fixed**

<https://github.com/agglayer/security/issues/116>

This issue is fixed in [agglayer/provers #221](#). The commented out code is uncommented as it should be.

### AGLO3.2-22.5 - Miscellaneous General Comments: Address TODOs

Severity: **Informational**

Resolution: **Tech Debt**

<https://github.com/agglayer/security/issues/117>

Currently being addressed.