



Security Assessment Report



Agora LayerZero Update

October 2025

Prepared for Agora Team

Table of content

Project Summary.....	3
Project Scope.....	3
Project Overview.....	3
Findings Summary.....	5
Severity Matrix.....	5
Detailed Findings.....	6
Informational Issues.....	7
I-01. No pause check during minting tokens.....	7
I-02. Bridging can be maliciously halted due to the rate limit.....	8
Disclaimer.....	9
About Certora.....	9

Project Summary

Project Scope

Project Name	Repository (link)	Commit Hash	Platform
Agora LayerZero Update	https://github.com/amphora-atlas/layerzero-dev	5e01a10 (initial commit) b501b34 (fix commit)	Solidity

Project Overview

This document describes the manual code review findings of **Agora LayerZero Update**. The following contract list is included in our scope:

- layerzero-dev/contracts/AgoraMintBurnOFTAdapter.sol
- layerzero-dev/contracts/AgoraOFTAdapterAccessControl.sol
- layerzero-dev/contracts/AgoraRateLimiter.sol
- layerzero-dev/contracts/LZRateLimiterUpgradeable.sol

The work was undertaken from **September 29, 2025**, to **October 13, 2025**. During this time, Certora's security researchers performed a manual audit of all the Solidity contracts and discovered several bugs in the codebase, which are summarized in the subsequent section.

Protocol and Scope Overview

Agora issues AUSD, a fiat-backed stablecoin fully collateralized by USD reserves. AUSD supports on-chain transfers across multiple networks using LayerZero for cross-chain interoperability.

This review focuses on the LayerZero integration: a MintBurn OFT adapter and supporting tooling that bridge Agora's token. The adapter implements `_debit` and `_credit` (the `OFTAdapterUpgradeable` hooks) by calling the token's `burn` and `mint` functions so cross-chain transfers burn on the source chain and mint on the destination. It uses an

upgradeable rate limiter that caps chain-level outflows and supports a `RATE_LIMIT_EXEMPT_ROLE` for partners.

The changes also integrate the `AgoraAccessControl` module (with explicit grant/revoke helpers and dedicated roles) and expose pausability via OpenZeppelin's `Pausable` for emergency stops.

Findings Summary

The table below summarizes the findings of the review, including type and severity details.

Severity	Discovered	Confirmed	Fixed
Critical	-	-	-
High	-	-	-
Medium	-	-	-
Low	-	-	-
Informational	2	2	1
Total	2	2	1

Severity Matrix

Impact	High	Medium	High	Critical
	Medium	Low	Medium	High
	Low	Low	Low	Medium
		Low	Medium	High
Likelihood				

Detailed Findings

ID	Title	Severity	Status
I-01	No pause check during minting tokens	Informational	Fixed
I-02	Bridging can be maliciously halted due to the rate limit	Informational	Acknowledged

Informational Issues

I-01. No pause check during minting tokens

File: [AgoraMintBurnOFTAdapter.sol](#)

Description

The `_debit` function responsible for burning tokens for bridging contains a `requireNotPaused` check, which prevents the action when the contract is paused. However, such a check is not present in the `_credit` function, responsible for minting tokens to the recipient on the receiving chain. This means that transactions in flight can still get processed if the contract is paused, and the `totalSupply` of the chain can increase. This can be unexpected since bridging is paused.

Recommendations

Consider adding a pause check to the `_credit` function. This will reject in-flight bridging transactions and the transaction will need to be replayed once the pause is removed from the contract.

Customer's response: A pause check has been added to the credit function [here](#)

Fix Review: Fix looks good.

I-02. Bridging can be maliciously halted due to the rate limit

File: [LZRateLimiterUpgradeable.sol](#)

Description

Due to the implementation of the rate limit, any user can halt the bridging for other users by using up the limit. Layerzero will charge the same fees for transactions irrespective of the size, so malicious users can utilize the entire limit in a single cheap transaction as long as they have the funds. While there is no way to profit from this unless there is a price discrepancy on the different chains, this can still be a soft DOS vector.

Recommendations

Since there is no simple way to prevent this issue, consider documenting and acknowledging this behavior.

Customer's response: Yes. Having a rate limit adds a DOS vector, but there is not much we can do about that, short of dynamic fees based on bridging size. Acknowledged.

Disclaimer

Even though we hope this information is helpful, we provide no warranty of any kind, explicit or implied. The contents of this report should not be construed as a complete guarantee that the contract is secure in all dimensions. In no event shall Certora or any of its employees be liable for any claim, damages, or other liability, whether in an action of contract, tort, or otherwise, arising from, out of, or in connection with the results reported here.

About Certora

Certora is a Web3 security company that provides industry-leading formal verification tools and smart contract audits. Certora's flagship security product, Certora Prover, is a unique SaaS product that automatically locates even the most rare & hard-to-find bugs on your smart contracts or mathematically proves their absence. The Certora Prover plugs into your standard deployment pipeline. It is helpful for smart contract developers and security researchers during auditing and bug bounties.

Certora also provides services such as auditing, formal verification projects, and incident response.