certora

Security Assessment & Formal Verification Report

BORED GHOSTS DEVELOPING aave

Liquid EModes

September-2024

Prepared for
BGD Labs on Aave

Table of content

Project Summary

Project Scope

Project Name	Repository (link)	Latest Commit Hash	Platform
Liquid eModes	https://github.com/aave-dao/aave-v3-origin	dd0bbec	EVM/Solidity 0.8

Project Overview

This document describes the specification and verification of the **Liquid eModes** using the Certora Prover and manual code review findings. The work was undertaken from **8 September 2024 to 19 September 2024**.

The following contract list is included in our scope:

- AaveProtocolDataProvider.sol
- IPoolConfigurator.sol
- IPool.sol
- IPoolDataProvider.sol
- EModeConfiguration.sol
- ReserveConfiguration.sol
- Errors.sol
- ConfiguratorLogic.sol
- EModeLogic.sol
- GenericLogic.sol
- LiquidationLogic.sol
- ValidationLogic.sol
- DataTypes.sol
- PoolConfigurator.sol
- Pool.sol

The Certora Prover demonstrated that the implementation of the Solidity contracts above is correct with respect to the formal rules written by the Certora team. In addition, the team performed a manual audit of all the Solidity contracts**.** During the verification process and the manual audit, no bug was discovered.

Protocol Overview

The contracts under review are part of the Aave Protocol and introduce new eMode logic to the Aave Protocol which is an evolution of the original eModes feature which includes:

- Removal of the eMode oracle which was never in use in order to save gas on storage packing
- Borrowable in eMode feature which allows a configuration of which assets are borrowable for a certain eMode. This means each asset must be allowed to be borrowable in a certain eMode. If an asset is borrowable in eMode and then removes it from being borrowable, existing positions will remain intact but no further exposure will be allowed. In order to enter a certain eMode, all assets in a current position must be borrowable in the same eMode or repaid first.
- Collateral in eMode feature which allows a configuration of which assets are collateral for a certain eMode. This means each asset must be allowed to be collateral in a certain eMode. If an asset is not a collateral in eMode it can still be a collateral but with no eMode parameters calculated for the collateral but the normal LTV/LT/etc.
- When being liquidated and during any health factor calculation, the eMode's LT/LB/etc, will only apply to eMode collaterals, other collaterals are still possible, but with the collateral's original LT/LB/etc.
- Assets no longer have a specific eMode but can have multiple eModes which allows to open new eMode categories where assets can be in multiple categories.

Coverage

1. We wrote a couple of rules in order to check the setters / getters mechanism. See more information later.
2. With respect to manual auditing we have checked the following:
    - We have checked that for a user to be able to enter/switch an eMode: All borrowed assets must be borrowable in the new eMode.
    - We have checked that for a user to be able to enter/switch an eMode: The health factor of the user must be >= 1 after the switch.
    - We have checked that in case an asset was borrowable in eMode and then removed from it, the position stays intact, but the exposure can no longer be increased.

- We have checked that when being liquidated (and any Health Factor calculation) the eMode LT/LB will only apply to the position's eMode collaterals.
- We have checked that a user can only borrow assets that are borrowable in his specific eMode.
- We have checked that positions can use any collateral while being in an eMode. The eMode LT/LTV will only apply for the ones listed as eModeCollateral for that specific eMode, while the others will use their non-eMode LT/LTV.
- We have checked that for an asset to be borrowable in eMode it has to be borrowable outside of the eMode. (There is a possibility that the flag of asset is borrowable in eMode is true while the asset is not borrowable, but borrows of the asset will not be possible even with the flag raised).

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			
Total			

Severity Matrix

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

Formal Verification

Verification Notations

Formally Verified	The rule is verified for every state of the contract(s), under the assumptions of the scope/requirements in the rule.
Formally Verified After Fix	The rule was violated due to an issue in the code and was successfully verified after fixing the issue
Violated	A counter-example exists that violates one of the assertions of the rule.

Formal Verification Properties

In the table below we specify all the formally verified rules that we wrote for the verification of the liquid eModes, and give a detailed description for them. A link to the Certora's prover report can be found here.

P-01. setter/getter integrity

Status: Verified	Property Assumptions:

Rule Name	Status	Description	Rule Assumptions
setCollateralIntegrity/ setBorrowableIntegrity	Verified	*Check the integrity of the functions setReserveBitmapBit (which is a setter) and isReserveEnabledOnBitmap (which is a getter), simply by setting an arbitrary value to arbitrary location, and then reading it using the getter.*	

P-02. independency of setReserveBitmapBit

Status: Verified	Property Assumptions:

Rule Name	Status	Description	Rule Assumptions
independencyOfCollateralSetters/ independencyOfBorrowableSetters	Verified	*Check that when calling to setReserveBitmapBit (index,val) only the value at the given index may be altered.*	

Disclaimer

The Certora Prover takes a contract and a specification as input and formally proves that the contract satisfies the specification in all scenarios. Notably, the guarantees of the Certora Prover are scoped to the provided specification and the Certora Prover does not check any cases not covered by the specification.

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.