certora

Formal Verification Report

Origin Protocol

December 2024

Prepared for Origin

certora

Table of content

certora

Project Summary

Project Scope

Project Name	Repository (link)	Latest Commit Hash	Platform
Origin Dollar	Origin-Dollar	5e57112	EVM

Project Overview

This document describes the specification and verification of Origin-Dollar using the Certora Prover. The work was undertaken from **Nov 26th 2024** to **Dec 12th 2024.**

The following contract list is included in our scope:

`contracts/contracts/token/OUSD.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.

certora

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		

certora

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.

General Assumptions and Simplifications

1. We used Solidity Compiler version 8.28 to verify the protocol.
2. In some rules we have assumed that the contract has enough credit balance resolution, i.e. rebasingCreditsPerToken_ >= 1e18.
   Otherwise, rounding errors could become significant when the numbers are low.
   The balance of a user could deviate from the intended (rounding-error free) result by at most 1e18/rebasingCreditsPerToken_.

Formal Verification Properties

OUSD

Module General Assumptions

Module Properties

P-01. Account Invariants	
Status: Verified	

Rule Name	Status	Description	Link to rule report
DelegationAccountsCorrelation	Verified	*Any non-zero valued YieldTo points to an account with a YieldFrom pointing back to the starting account and vice versa.*	*Report*
DelegationValidRebaseState	Verified	*Any non-zero valued YieldTo points to an account Iff that account is in YieldDelegationSource state and any non-zero valued YieldFrom points to an account Iff that account is in YieldDelegationTarget state.*	*Report*
nonZeroAlternativeCreditsPerTokenStates	Verified	*Any account with a value of 1e18 in alternativeCreditsPerToken has a rebaseState that is one of (StdNonRebasing, YieldDelegationSource).*	*Report*
stdNonRebasingDoesntYield	Verified	*Any account in StdNonRebasing state doesn't yield to no account.*	*Report*
alternativeCreditsPerTokenIsOneOrZeroOnly	Verified	*alternativeCreditsPerToken can only be set to 0 or 1e18, no other values.*	*Report*
yieldDelegationSourceHasNonZeroYeildTo	Verified	*Any account with rebaseState = YieldDelegationSource has a nonZero yieldTo.*	*Report*

yieldDelegationTargetHasNonZeroYeildFrom	Verified	*Any account with rebaseState = YieldDelegationTarget has a nonZero yieldFrom.*	*Report*
zeroAlternativeCreditsPerTokenStates	Verified	*Any account with a zero value in alternativeCreditsPerToken has a rebaseState that is one of (NotSet, StdRebasing, or YieldDelegationTarget).*	*Report*
yieldFromOfZerolsZero	Verified	*yieldFrom of zero is zero.*	*Report*
yieldToOfZerolsZero	Verified	*yieldTo of zero is zero.*	*Report*
cantYieldFromSelf	Verified	*yieldFrom of an account can't be the same as the account.*	*Report*
cantYieldToSelf	Verified	*yieldTo of an account can't be the same as the account.*	*Report*
onlyDelegationChangesDelegateState	Verified	*Only delegation changes the different effective identity.*	*Report*

P-02. Balance Invariants

Status: Violated	Assumptions: - Token has enough resolution (rebasingCreditsPerToken_ >= 1e18) - The total supply is at least 10^16

Rule Name	Status	Description	Link to rule report
stdNonRebasingBalanceEqCreditBalances	Verified	*The balanceOf of any account in StdNonRebasing state equals the account's credit balance.*	*Report*
sumAllNonRebasingBalancesEqNonRebasingSupply	Verified	*The sum of all StdNonRebasing accounts equals the nonRebasingSupply.*	*Report*
sumAllRebasingCreditsEqRebasingCredits	Verified	*The sum of the credits in all NotSet, StdRebasing, and YieldDelegationTarget accounts equal the rebasingCredits.* *This property is violated for both rebaseOptIn and governanceRebaseOptIn which we show in 'sumAllRebasingCreditsAndTotalRebasingCreditsCorelation' that the violation is bounded.*	*Report*
sumAllRebasingCreditsAndTotalRebasingCreditsCorelation	Verified	*Ensure correlation between the delta in the sum of the credits in all NotSet, StdRebasing, and YieldDelegationTarget accounts match the delta in rebasingCredits allowing for a bounded rounding error calculated as `rebasingCreditsPerToken / 1e18` for both rebaseOptIn and governanceRebaseOptIn.*	*Report*
totalSupplyLessThanMaxSupply	Verified	*Verify that the total supply remains within the maximum allowable limit.*	*Report*
undelegateYieldPreservesSumOfBalances	Verified	*Verify that the total balance of delegator and delegatee remains unchanged after undelegation.*	*Report*

delegateYieldPreservesSumOfBalances	Verified	*Verify that the total balance of delegator and delegatee remains unchanged after yield delegation.*	*Report*
transferPreservesSumOfBalances	Verified	*Both transfer methods must preserve the sum of balances. The total supply and any balance of a third party cannot change.*	*Report*
sumOfTwoAccountsBalancesLETotalSupply	Verified	*The sum of balances of any two accounts cannot surpass the total supply.*	*Report*
changeSupplyPreservesSumOFRebasingLesEqTotalSupply	Verified	*The sum of all rebasing account balances cannot surpass the total supply after calling for changeSupply.*	*Report*

P-03. Balance Integrities

Status: Verified	Assumptions: - Token has enough resolution (rebasingCreditsPerToken_ >= 1e18) - The total supply is at least 10^16

Rule Name	Status	Description	Link to rule report
burnIntegrity	Verified	*A successful burn() call by the vault results in the target account's balance decreasing by the amount specified.*	*Report*
mintIntegrity	Verified	*A successful mint() call by the vault results in the target account's balance increasing by the amount specified.*	*Report*
rebaseOptInIntegrity	Verified	*After a non-reverting call to rebaseOptIn() the alternativeCreditsPerToken[account] == 0 and does not result in a change in account balance.*	*Report*

governanceRebaseOptInIntegrity	Verified	*After a non-reverting call to governanceRebaseOptIn() the alternativeCreditsPerToken[account] == 0 and does not result in a change in account balance.*	Report
rebaseOptOutIntegrity	Verified	*After a non-reverting call to rebaseOptOut() the alternativeCreditsPerToken[account] == 1e18 and does not result in a change in account balance.*	Report
burnIntegrityThirdParty	Verified	*Any third-party account balance should not change after a burn operation.*	Report
mintIntegrityThirdParty	Verified	*Any third-party account balance should not change after a mint operation.*	Report
transferIntegrityTo	Verified	*Recipient and sender (msg.sender) account balances should increase and decrease respectively by the amount after a transfer operation. Account balance should not change after a transfer operation if the recipient is the sender.*	Report
transferThirdParty	Verified	*Transfer doesn't change the balance of a third party.*	Report
whoCanChangeBalance	Verified	*Only transfer, transferFrom, mint, burn, and changeSupply result in a change in any account's balance.*	Report
whoCanChangeNonRebasingBalance	Verified	*Only transfers, mints, and burns change the balance of StdNonRebasing and YieldDelegationSource accounts.*	Report
balanceOfIntegrity	Verified	*Verify account balance integrity based on rebase state.* *Ensures balances are correctly calculated for Yield Delegation Targets, Standard Rebasing, Non-Rebasing, and undefined (NotSet) states to maintain consistency in OUSD accounting.*	Report

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.