certora

Security Assessment & Formal Verification Report

ether.fi

Ether-Fi CashWrapper

September 2024

Prepared for EtherFi

Table of content

Project Summary

Project Scope

Project Name	Repository (link)	Latest Commit Hash	Platform
EtherFi smart contracts	https://github.com/etherfi-protocol/cash-contracts/tree/audit/netherminds/src/cash-wrapper-token	ecb8fd4cdb6780be026642d44c5128812d1e9487	EVM

Project Overview

This document describes the specification and verification of **EtherFi cash-contracts** using the Certora Prover and manual code review findings. The work was undertaken from **22/09/2024** to **23/09/2024.**

The following contract list is included in our scope:

```
src/cash-wrapper-token/CashTokenWrapperFactory.sol
src/cash-wrapper-token/CashWrappedERC20.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, the Certora team didn't discover bugs in the Solidity contracts code.

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	0	0	0

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 8.24 to verify the protocol.

Formal Verification Properties

Cash Wrapped ERC20

Module General Assumptions

- We assume that all loops iterate at most three times.
- We used a simple mock ERC20 as the base token.
- We assume no-overflows in total supply updates

Module Properties

P-01. balance of address(0) is 0.	
Status: Verified	

Rule Name	Status	Description	Link to rule report
zeroAddressNoBalance	Verified	*Verifies that the balance of address(0) is 0.*	*Report*

P-02. only mint and withdraw can change the total supply.	
Status: Verified	

Rule Name	Status	Description	Link to rule report
noChangeTotalSupply	Verified	*Verifies that total supply can be increased only by 'mint' and decreased only by 'withdraw'*	*Report*

P-03. only whitelisted minters can increase the total supply.

Status: Verified	Assumptions:

Rule Name	Status	Description	Link to rule report
onlyWhitelistedCanMint	Verified	*Verifies that only whitelisted minters can increase the total supply.*	[Report]

P-04. only whitelisted recipients can have their balance increased.

Status: Verified	Assumptions:

Rule Name	Status	Description	Link to rule report
onlyWhitelistedRecipientCanIncreaseTheirBalance	Verified	*Verifies that only whitelisted recipients can have their balance increased.*	[Report]

P-05. allowance mechanism correct.

Status: Verified	Assumptions:

Rule Name	Status	Description	Link to rule report
onlyHolderOfSpenderCanChangeAllowance	Verified	*Verifies that only the token holder (or a permit) can increase allowance. The spender can decrease it by using it.*	[Report]

P-06. mint behavior and side effects.	
Status: Verified	Assumptions:

Rule Name	Status	Description	Link to rule report
mint	Verified	*Verifies that ‘mint’ function reverts when it should and works correctly when it doesn’t revert.*	Report

P-07. withdraw behavior and side effects.	
Status: Verified	Assumptions:

Rule Name	Status	Description	Link to rule report
withdraw	Verified	*Verifies that ‘withdraw’ function reverts when it should and works correctly when it doesn’t revert.*	Report

P-08. transfer behavior and side effects.	
Status: Verified	Assumptions:

Rule Name	Status	Description	Link to rule report
transfer	Verified	*Verifies that ‘transfer’ function reverts when it should and works correctly when it doesn’t revert.*	Report

P-09. transferFrom behavior and side effects.	
Status: Verified	Assumptions:

Rule Name	Status	Description	Link to rule report
transferFrom	Verified	*Verifies that 'transferFrom' function reverts when it should and works correctly when it doesn't revert.*	Report

P-10. approve behavior and side effects.	
Status: Verified	Assumptions:

Rule Name	Status	Description	Link to rule report
approve	Verified	*Verifies that 'approve' function reverts when it should and works correctly when it doesn't revert.*	Report

P-11. permit behavior and side effects.	
Status: Verified	Assumptions:

Rule Name	Status	Description	Link to rule report
permit	Verified	*Verifies that 'permit' f works correctly when it doesn't revert.*	Report

P-12. whitelistMinters behavior and side effects.

Status: Verified	Assumptions:

Rule Name	Status	Description	Link to rule report
whitelistMinters	Verified	*Verifies that 'whitelistMinters' function reverts when it should and works correctly when it doesn't revert.*	[Report]()

P-13. whitelistRecipients behavior and side effects.

Status: Verified	Assumptions:

Rule Name	Status	Description	Link to rule report
whitelistRecipients	Verified	*Verifies that 'whitelistRecipients' function reverts when it should and works correctly when it doesn't revert.*	[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.