CD SECURITY

AUDIT REPORT

Privacy Pools

March 2024

Introduction

A time-boxed security review of the **Privacy Pools** protocol was done by **CD Security**, with a focus on the security aspects of the application's implementation.

Disclaimer

A smart contract security review can never verify the complete absence of vulnerabilities. This is a time, resource, and expertise-bound effort where we try to find as many vulnerabilities as possible. We can not guarantee 100% security after the review or even if the review will find any problems with your smart contracts. Subsequent security reviews, bug bounty programs, and on-chain monitoring are strongly recommended.

About **Privacy Pools**

Introducing the Privacy Pools project, a groundbreaking system designed to enhance user privacy and security in financial transactions. This innovative platform operates on a three-stage process:

1. Deposit
2. Wait
3. Withdraw

Features:

- Deposited funds cannot be locked or stolen (non-custodial and non-restrictive)
- Zero knowledge proofs secure user's privacy
- Users have the freedom to choose an anonymity set upon withdrawal
- Removing illicit deposits from an anonymity subset accomplished two things:
- `Proves a withdrawal is not from illicit or sanctioned funds without violating the privacy of the specific user`, and
- `Reduces the anonymity sets of hackers, acting as a deterrent and as a dampening force for illicit activity`
- Enables customizable community driven anti-blackhat and anti-money laundering coordination in a credibly neutral way

More docs

Severity classification

Severity	Impact: High	Impact: Medium	Impact: Low
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low

Severity	Impact: High	Impact: Medium	Impact: Low
Likelihood: Low	Medium	Low	Low

Impact - the technical, economic, and reputation damage of a successful attack

Likelihood - the chance that a particular vulnerability gets discovered and exploited

Severity - the overall criticality of the risk

Security Assessment Summary

***review commit hash* - 440794ab1c0d738b9e88618150958075883e3378**

Scope

The following smart contracts were in scope of the audit:

- `./contracts/ProofLib.sol`
- `./contracts/PrivacyPool.sol`
- `./contracts/PrivacyPoolFactory.sol`
- `./circuits/*`

The following number of issues were found, categorized by their severity:

- Critical & High: 0 issues
- Medium: 0 issues
- Low: 1 issues
- Informational 2 issues

Findings Summary

ID	Title	Severity
[L-01]	Protocol is using a vulnerable library version	Low
[I-01]	Unused custom errors	Informational
[I-02]	Use newer pragma version	Informational

Detailed Findings

[L-01] Protocol is using a vulnerable library version

In `package.json` file in the repository we can see this:

```
"@openzeppelin/contracts": "^4.7.3",
```

This version contains multiple vulnerabilities as you can see here. While the problems are not present in the current codebase, it is strongly advised to upgrade the version to v5.0.2 which is the newest and has fixes for all of the vulnerabilities found so far after v4.7.3.

Contributor:

"The dependency on the OpenZeppelin contracts library has been updated to the latest secure version. This update ensures the incorporation of all the recent security fixes and enhancements available."

[I-01] Unused custom errors

These two custom errors declared in `PrivacyPoolFactory.sol` are not used anywhere in the contracts so they can be removed:

```
error PoolUnknown(address asset, uint256 power, uint256 index);
error WithdrawFailed();
```

Contributor:

"Removed the unused custom error declarations from `PrivacyPoolFactory.sol`."

[I-02] Use newer pragma version

All the contracts use pragma version 0.8.17. Consider using newer version as new features and optimizations are constantly introduced.

Contributor:

"The pragma version in all contracts has been updated to the latest stable release. This change allows to leverage new language features and compiler optimizations, enhancing the security and performance of our smart contracts."