ZEROPOOL

Tornado.Cash Trees Audit

Igor Gulamov, 2 March 2021

Introduction

ZeroPool conducted the audit of Tornado.Cash Trees smart contracts and circuits.

This review was performed by an independent reviewer under a fixed rate.

Scope

https://github.com/tornadocash/tornado-trees/blob/59f13c3aaa2f5a41091db3db110aae42de9b9c26/circuits/BatchTreeUpdate.circom

https://github.com/tornadocash/tornado-trees/blob/59f13c3aaa2f5a41091db3db110aae42de9b9c26/circuits/Utils.circom

https://github.com/tornadocash/tornado-trees/blob/59f13c3aaa2f5a41091db3db110aae42de9b9c26/contracts/TornadoTrees.sol

Issues

We found no critical issues. One major issue found and fixed at 93dd5d5e861a7d2b78ac2ee8233be573ee1a7e97.

We consider the commit 93dd5d5e861a7d2b78ac2ee8233be573ee1a7e97 as a safe version from the informational security point of view.

Major

1. **Utils.circom#L24-L25 Utils.circom#L37**

Output result for `Num2Bits(n)` is ambiguous for the case n≥254

We recommend replacing it with `Num2Bits_strict`, or use 253bit or lesser numbers, or check all input data inside the preimage at the contract level.

Comment

Fixed at 93dd5d5e861a7d2b78ac2ee8233be573ee1a7e97

Warnings

1. **https://github.com/tornadocash/tornado-trees/blob/59f13c3aaa2f5a41091db3db110aae42de9b9c26/circuits/Utils.circom#L27-L32**

255th and 256th bits are always zero and corresponding signals could be hardcoded to zero for better optimization.

Comment

Fixed at 93dd5d5e861a7d2b78ac2ee8233be573ee1a7e97

2. **TornadoTrees.sol#L162 TornadoTrees.sol#L207**

Unnecessary argument `_argsHash`. This variable is computed onchain.

Comment

Won't fix We use it to be able to have a separate revert message instead of generic invalid proof

3. **TornadoTrees.sol#L143**

`elementExists` is called multiple times from cycle. We propose replacing `encodeWithSignature` with `encodePacked` and using raw selectors, to reduce calls to the hash function.

Comment

Won't fix It's used only once during the contract deployment, so we rather have better code readability.

4. [TornadoTrees.sol#L160]

We propose adding field overflow checks to `updateDepositTree` by the same way, as for `updateWithdrawalTree`.

Comment

Won't fix All field overflow checks for the preimage will be moved to the circuit

Severity Terms

Comment

Comment issues are generally subjective in nature, or potentially deal with topics like "best practices" or "readability". Comment issues in general will not indicate an actual problem or bug in code.

The maintainers should use their own judgment as to whether addressing these issues improves the codebase.

Warning

Warning issues are generally objective in nature but do not represent actual bugs or security problems.

These issues should be addressed unless there is a clear reason not to.

Major

Major issues will be things like bugs or security vulnerabilities. These issues may not be directly exploitable, or may require a certain condition to arise in order to be exploited.

Left unaddressed these issues are highly likely to cause problems with the operation of the contract or lead to a situation which allows the system to be exploited in some way.

Critical

Critical issues are directly exploitable bugs or security vulnerabilities.

Left unaddressed these issues are highly likely or guaranteed to cause major problems or potentially a full failure in the operations of the contract.