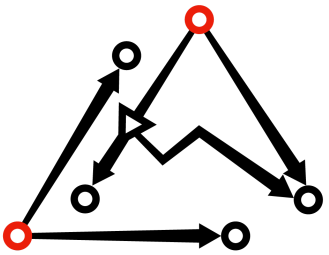

Least Authority

PRIVACY MATTERS

SMPC Protocol (Third Review)

Security Audit Report

Worldcoin

Updated Final Audit Report: 25 February 2025

Table of Contents

Overview

Background

Tools for Humanity Corporation has requested that Least Authority perform a third review of the SMPC Protocol.

Project Dates

- **October 16, 2024 - October 31, 2024:** Initial Code Review *(Completed)*
- **November 22, 2024:** Delivery of Initial Audit Report *(Completed)*
- **February 24, 2025:** Verification Review *(Completed)*
- **February 24, 2025:** Delivery of FInal Audit Report *(Completed)*
- **February 25, 2025:** Delivery of Updated FInal Audit Report *(Completed)*

Review Team

- Anna Kaplan, Cryptography Researcher and Engineer

Coverage

Target Code and Revision

For this audit, we performed research, investigation, and a third review of the SMPC Protocol followed by issue reporting, along with mitigation and remediation instructions as outlined in this report.

The following code repositories are considered in scope for the review:

- SMPC Protocol:
  https://github.com/worldcoin/gpu-iris-mpc
  - Limited to the V2 delta since the second review
  - Excluded: cpu and V3 crates

Specifically, we examined the Git revision for our initial review:

- `2bb21b90ccc6547b525dd299f65586b647bd46ca`

For the review, this repository was cloned for use during the audit and for reference in this report:

- SMPC Protocol:
  https://github.com/LeastAuthority/worldcoin-gpu-iris-mpc-2nd-review/tree/v2-delta

All file references in this document use Unix-style paths relative to the project's root directory.

In addition, any dependency and third-party code, unless specifically mentioned as in scope, were considered out of scope for this review.

Supporting Documentation

The following documentation was available to the review team:

- Website:
  https://worldcoin.org
- Large-Scale MPC: Scaling Private Iris Code Uniqueness Checks to Millions of Users:
  https://eprint.iacr.org/2024/705.pdf

In addition, this audit report references the following documents:

- I. Damgård and R.Thorbek, "Efficient Conversion of Secret-shared Values Between Different Fields." *IACR Cryptology ePrint Archive*, 2008, [DT08]
- Previous Security Audit Report by Least Authority on the SMPC Protocol (Second Review) (pdf) *(delivered via email on 12 September 2024)*

Areas of Concern

Our investigation focused on the following areas:

- Correctness of the implementation;
- Vulnerabilities within each component and whether the interaction between the components is secure;
- Whether requests are passed correctly to the network core;
- Key management, including secure private key storage and management of encryption and signing keys;
- Denial of Service (DoS) and other security exploits that would impact the intended use or disrupt the execution;
- Protection against malicious attacks and other ways to exploit;
- Inappropriate permissions and excess authority;
- Data privacy, data leaking, and information integrity; and
- Anything else as identified during the initial analysis phase.

Findings

General Comments

Our team performed a review of the recent changes to Worldcoin's secure multi-party computation protocol V2 (SMPC Protocol), which is used to match a given iris against a database of iris shares. Our team previously reviewed the implementation's first and second versions. In this third review, we audited the changes implemented in the second version of the protocol since our last protocol review, which included changes across 297 merge commits, starting with the latest version we reviewed (`2ab7ae6`) and ending with the one that was implemented at the start of the audit (`2bb21b9`). More specifically, our team noted that several of these commits include critical code maintenance and health changes, such as commits to mark new releases or chores. We did not identify any issues within this context.

We reviewed the main change in this new version – adding HNSW search using Shamir secret sharing for both the iris code and iris masks in the database – in PRs #292, #327, #371, #483, #499, and #559. While doing so, the Worldcoin team updated the masks to be handled in a trimmed manner in PR #270. This entire change improves the privacy of the iris masks and uses the homomorphic properties possible for Shamir secret shares. To check the correctness of sets of iris codes and masks, the distance is computed through the Shamir secret shares and then converted to a replicated share. This can be improved by representing the distances as Shamir secret shares (Suggestion 2). The Worldcoin team has also added new benchmark tests for this functionality. We did not identify any issues relating to these areas of concern.

The Worldcoin team has also added support for both eyes to the system, in PRs #226 and #249, as well as for iris deletion, in PRs #320, #328 and #335. We did not identify any issues within these added functionalities.

Additionally, the Worldcoin team has updated the code to entail the encryption and decryption of iris code shares, which led to updates in the serialization and deserialization process (PRs #204 and #284).

Our team additionally reviewed the changes to batching, where batch size is passed as an argument, and batch timeouts are included (e.g., in PR #224). We did not identify any areas of concern.

Moreover, the Worldcoin team has removed the option to seed the randomness through the client (PR #369), a change we encourage to guarantee cryptographically sound randomness as reported in the previous review. Key management has also been improved by adding handling for cases where a previous key pair does not exist (PR #282).

Code Quality

We performed a manual review of the repositories in scope and found the code to be well-organized and in adherence to development best practices, especially for Rust programming with `async-await` properties.

Tests

The repositories in scope include sufficient test coverage. Additionally, during our review, we found that the tests for the functionalities that changed since our last review were also updated and added to their respective components.

Documentation and Code Comments

The project documentation provided by the Worldcoin team was generally sufficient in describing the intended functionality of the system. The Worldcoin team has also updated the architecture diagram,incorporated additional details on their code of conduct and disclosure policy for security vulnerabilities, and added dual-licensing to all code. Moreover, we found that code comments sufficiently describe the intended behavior of security-critical components and functions. In addition to these improvements, we suggest documenting a specification for both the protocol and code (Suggestion 1).

Scope

The scope of this review was sufficient, as it was limited to the differences between the second review and the current one.

Dependencies

Our team did not identify any vulnerabilities in the implementation's use of dependencies. (See the previous report, Issue A, for comments on dependency usage in this project.)

Specific Issues & Suggestions

We list the issues and suggestions found during the review, in the order we reported them. In most cases, remediation of an issue is preferable, but mitigation is suggested as another option for cases where a trade-off could be required.

ISSUE / SUGGESTION	STATUS
Suggestion 1: Document Protocol Specification	Resolved
Suggestion 2: Consider Changing Representation of Distances	Planned

Suggestions

Suggestion 1: Document Protocol Specification

Location

Entire repository.

Synopsis

In this update, the secret sharing protocol has been modified to share iris masks, which are then used to perform an HNSW search through the database.

Mitigation

We recommend writing a detailed specification of the protocol and code, including information on the data types and handling of the iris code and mask, with a particular focus on the search functionality.

Status

The Worldcoin team has provided a specification for the HNSW search on top of the MPC functionality.

Verification

Resolved.

Suggestion 2: Consider Changing Representation of Distances

Location

`src/hawkers/galois_store.rs#L149-L156`

`src/hawkers/galois_store.rs#L208-L213`

`iris-mpc-cpu/src/protocol/ops.rs#L248-L265`

Synopsis

Currently, two iris entries are compared by computing both the dot product of the iris code shares and the iris mask shares. These partial Shamir secret shares (the distances) are trimmed, converted to a replicated share by masking it with a zero sharing, and then matched to a threshold.

Mitigation

To improve efficiency, we recommend updating the code to consider distances as Shamir secret shares.

Status

The Worldcoin team acknowledged the suggestion and stated that the representation of distance is currently under consideration, with development efforts expected to be allocated in the coming weeks.

Additionally, the Worldcoin team clarified that they are currently exploring the addition of a precomputation step and, as they explore this preprocessing phase, will evaluate to what extent parts of the protocol can be restructured to a Shamir shares representation. However, the team noted that achieving efficiency with this approach may be challenging due to the high computational cost of multiplication operations in the Shamir shares Galois ring. Nevertheless, despite the trade-off involved, the team acknowledges its significance and will take it into consideration.

Verification

Planned.

About Least Authority

We believe that people have a fundamental right to privacy and that the use of secure solutions enables people to more freely use the Internet and other connected technologies. We provide security consulting services to help others make their solutions more resistant to unauthorized access to data and unintended manipulation of the system. We support teams from the design phase through the production launch and after.

The Least Authority team has skills for reviewing code in multiple Languages, such as C, C++, Python, Haskell, Rust, Node.js, Solidity, Go, JavaScript, ZoKrates, and circom, for common security vulnerabilities and specific attack vectors. The team has reviewed implementations of cryptographic protocols and distributed system architecture in cryptocurrency, blockchains, payments, smart contracts, zero-knowledge protocols, and consensus protocols. Additionally, the team can utilize various tools to scan code and networks and build custom tools as necessary.

Least Authority was formed in 2011 to create and further empower freedom-compatible technologies. We moved the company to Berlin in 2016 and continue to expand our efforts. We are an international team that believes we can have a significant impact on the world by being transparent and open about the work we do.

For more information about our security consulting, please visit https://leastauthority.com/security-consulting/.

Our Methodology

We like to work with a transparent process and make our reviews a collaborative effort. The goals of our security audits are to improve the quality of systems we review and aim for sufficient remediation to help protect users. The following is the methodology we use in our security audit process.

Manual Code Review

In manually reviewing all of the code, we look for any potential issues with code logic, error handling, protocol and header parsing, cryptographic errors, and random number generators. We also watch for areas where more defensive programming could reduce the risk of future mistakes and speed up future audits. Although our primary focus is on the in-scope code, we examine dependency code and behavior when it is relevant to a particular line of investigation.

Vulnerability Analysis

Our audit techniques include manual code analysis, user interface interaction, and whitebox penetration testing. We look at the project's website to get a high level understanding of what functionality the software under review provides. We then meet with the developers to gain an appreciation of their vision of the software. We install and use the relevant software, exploring the user interactions and roles. As we do this, we brainstorm threat models and attack surfaces. We read design documentation, review other audit results, search for similar projects, examine source code dependencies, skim open issue tickets, and generally investigate details other than the implementation. We hypothesize what vulnerabilities may be present and possibly resulting in Issue entries, then for each, we follow the following Issue Investigation and Remediation process.

Documenting Results

We follow a conservative and transparent process for analyzing potential security vulnerabilities and seeing them through successful remediation. Whenever a potential issue is discovered, we immediately create an Issue entry for it in this document, even before having verified the feasibility and impact of the issue. This process is conservative because we document our suspicions early even if they are later shown to not represent exploitable vulnerabilities. We generally follow a process of first documenting the suspicion with unresolved questions, then confirming the issue through code analysis, live experimentation, or automated tests. Code analysis is the most tentative, and we strive to provide test code, log captures, or screenshots demonstrating our confirmation. After this, we analyze the feasibility of an attack in a live system.

Suggested Solutions

We search for immediate and comprehensive mitigations that live deployments can take, and finally, we suggest the requirements for remediation engineering for future releases. The mitigation and remediation recommendations should be scrutinized by the developers and deployment engineers, and successful mitigation and remediation is an ongoing collaborative process after we deliver our Initial Audit Report, and before we perform a verification review.

Before our report, including any details about our findings and the solutions are shared, we like to work with your team to find reasonable outcomes that can be addressed as soon as possible without an overly negative impact on pre-existing plans. Although the handling of issues must be done on a case-by-case basis, we always like to agree on a timeline for a resolution that balances the impact on the users and the needs of your project team.

Resolutions & Publishing

Once the findings are comprehensively addressed, we complete a verification review to assess that the issues and suggestions are sufficiently addressed. When this analysis is completed, we update the report and provide a Final Audit Report that can be published in whole. If there are critical unaddressed issues, we suggest the report not be published and the users and other stakeholders be alerted of the impact. We encourage that all findings be dealt with and the Final Audit Report be shared publicly for the transparency of efforts and the advancement of security learnings within the industry.