PROTOFIRE

February 2026

Smart contract audit

rain.lib.hash

Content

Project description

The rain.lib.hash repository is a native Solidity library designed to provide gas-efficient hashing primitives that avoid memory allocation. Its primary objective is to replace the standard, often gas-intensive keccak256(abi.encode(...)) pattern with a highly optimized assembly-based approach.

The library exports the LibHashNoAlloc library, which provides functions to hash bytes, arrays of words, and combine hashes (hashBytes, hashWords, combineHashes). By operating directly on existing memory structures and utilizing the EVM's scratch space, the library eliminates the need for the memory expansion and data copying inherent in ABI encoding. This approach significantly reduces gas costs for cryptographic operations, particularly in high-throughput or compute-constrained environments, while maintaining cryptographic security through deterministic and injective hashing patterns.

Executive summary

Type	Library
Languages	Solidity
Methods	Architecture Review, Manual Review, Unit Testing, Functional Testing, Automated Review
Documentation	README.md
Repositories	https://github.com/rainlanguage/rain.lib.hash

Reviews

Date	Repository	Commit
02/02/26	rain.lib.hash	33e17004ed6f9248d5cbb139c93da7190770c1f4

Scope

Contracts
src/LibHashNoAlloc.sol

Technical analysis and findings

Critical	0
High	0
Medium	0
Low	0
Informational	0

Security findings

**** Critical

No critical severity issue found.

*** High

No high severity issue found.

** Medium

No medium severity issue found.

* Low

No low severity issue found.

Informational

No informational severity issue found.

Approach and methodology

To establish a uniform evaluation, we define the following terminology in accordance with the OWASP Risk Rating
Methodology:

	**Likelihood** Indicates the probability of a specific vulnerability being discovered and exploited in real-world scenarios
	**Impact** Measures the technical loss and business repercussions resulting from a successful attack
	**Severity** Reflects the comprehensive magnitude of the risk, combining both the probability of occurrence (likelihood) and the extent of potential consequences (impact)

Likelihood and impact are divided into three levels: High H, Medium M, and Low L. The severity of a risk is a blend of these two factors, leading to its classification into one of four tiers: Critical, High, Medium, or Low.

When we identify an issue, our approach may include deploying contracts on our private testnet for validation through testing. Where necessary, we might also create a Proof of Concept PoC to demonstrate potential exploitability. In particular, we perform the audit according to the following procedure:

	**Advanced DeFi Scrutiny** We further review business logics, examine system operations, and place DeFi-related aspects under scrutiny to uncover possible pitfalls and/or bugs
	**Semantic Consistency Checks** We then manually check the logic of implemented smart contracts and compare with the description in the white paper.
	**Security Analysis** The process begins with a comprehensive examination of the system to gain a deep understanding of its internal mechanisms, identifying any irregularities and potential weak spots.