PROTOFIRE

February 2026

Smart contract audit

rain.lib.memkv

Content

Project description

This repository contains a Solidity library that implements an optimized in-memory key/value store using Yul for gas efficiency. It utilizes a hash map structure backed by 15 internal linked lists to provide roughly O(1) time complexity for get and set operations.

The library is designed to handle key/value pairs (both bytes32) within the scope of a single transaction. It features a custom MemoryKV pointer type that tracks the head of the lists and the total word count, allowing for efficient O(1) memory allocation when exporting the store. A specialized toBytes32Array function allows the entire data structure to be snapshotted into a flat uint256[] array using a gas-optimized bisection algorithm. This library is intended for complex transaction logic requiring temporary, high-performance structured data storage that does not require persistent contract storage.

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.memkv

Reviews

Date	Repository	Commit
06/02/26	rain.lib.memkv	8000285775e554ae157393dc4e2068ef2ffa7e3a

Scope

Contracts
src/lib/LibMemoryKV.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.