PROTOFIRE

February 2026

Smart contract audit rain.sol.binmaskflag

Content

Project description

This repository, rain.sol.binmaskflag, is a native Solidity library designed to provide a comprehensive set of pre-defined constants for binary sequences and bitmasks. Its primary objective is to replace runtime computations of powers of two and bitwise operations with gas-efficient compile-time constants.

The library exports uint256 constants representing specific binary patterns (e.g., B_111 for binary 111) and explicit bitmasks ranging from 1 bit up to 64 bits (e.g., MASK_1BIT, MASK_64BIT). This approach enhances code readability, reduces the likelihood of errors associated with manual bitwise arithmetic, and optimizes gas consumption during contract execution by offloading computation to the compiler.

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.sol.binmaskflag

Reviews

Date	**Repository**	**Commit**
02/02/26	rain.sol.binmaskflag	41d8dfb2edf9695d1344e8d56ddae7c1789690c4

Scope

Contracts
src/Binary.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.

General Risks

- Bitmask Literal Maintainability Risk: The contract defines many bitmask constants using visually counted binary patterns (e.g., B_11111111111111111111111111111111). While functionally correct, this approach relies on manual verification of the number of 1 bits in each literal. This creates a maintenance risk: future modifications, copy-paste edits, or reviews may fail to detect an incorrect bit length, leading to subtle logic errors in bit-packing, masking, or bounds enforcement.

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.