Quantstamp Security Assessment Certificate

February 24th 2021 — Quantstamp Verified

Sequence Smart Wallet

This security assessment was prepared by Quantstamp, the leader in blockchain security.

Executive Summary

Type	Smart Contract Wallet
Auditors	Martin Derka, Senior Research Engineer Alex Murashkin, Senior Software Engineer Sung-Shine Lee, Research Engineer
Timeline	2021-01-28 through 2021-02-08
EVM	Muir Glacier
Languages	Solidity
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review
Specification	Online code walk-through
Documentation Quality	Medium
Test Quality	High

Source Code

Repository	Commit
wallet-contracts	7492cb3

Goals

- Assessing support of ERC1271
- Assessing security of the signing mechanism after the changes

Total Issues	**2** (0 Resolved)
High Risk Issues	0 (0 Resolved)
Medium Risk Issues	0 (0 Resolved)
Low Risk Issues	0 (0 Resolved)
Informational Risk Issues	**2** (0 Resolved)
Undetermined Risk Issues	0 (0 Resolved)

All Branches Covered

High Risk	The issue puts a large number of users' sensitive information at risk, or is reasonably likely to lead to catastrophic impact for client's reputation or serious financial implications for client and users.
Medium Risk	The issue puts a subset of users' sensitive information at risk, would be detrimental for the client's reputation if exploited, or is reasonably likely to lead to moderate financial impact.
Low Risk	The risk is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low-impact in view of the client's business circumstances.
Informational	The issue does not post an immediate risk, but is relevant to security best practices or Defence in Depth.
Undetermined	The impact of the issue is uncertain.

Unresolved	Acknowledged the existence of the risk, and decided to accept it without engaging in special efforts to control it.
Acknowledged	The issue remains in the code but is a result of an intentional business or design decision. As such, it is supposed to be addressed outside the programmatic means, such as: 1) comments, documentation, README, FAQ; 2) business processes; 3) analyses showing that the issue shall have no negative consequences in practice (e.g., gas analysis, deployment settings).
Resolved	Adjusted program implementation, requirements or constraints to eliminate the risk.
Mitigated	Implemented actions to minimize the impact or likelihood of the risk.

Summary of Findings

This audit is based on Quantstamp's previous audit of the Arcadeum Wallet. The only part within the scope of the audit is the pull request #105 concerned with support of ERC1271. Overall, the code provides support for ERC1271, is well organized, respects the best practices (with the exception of the instances listed in this report), and is equipped with a reasonable amount of in-code documentation. The auditors did not discover any serious security concerns, and only missed a more comprehensive external documentation of the codebase in its entirety.

Update: The Sequence Smart Wallet team acknowledged all findings outlined in this report.

ID	Description	Severity	Status
QSP-1	Unhandled Overflows	Informational	Acknowledged
QSP-2	Missing Return Values	Informational	Acknowledged

Quantstamp Audit Breakdown

Quantstamp's objective was to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices.

Possible issues we looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Mishandled exceptions and call stack limits
- Unsafe external calls
- Integer overflow / underflow
- Number rounding errors
- Reentrancy and cross-function vulnerabilities
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting

Methodology

The Quantstamp auditing process follows a routine series of steps:

1. Code review that includes the following
   i. Review of the specifications, sources, and instructions provided to Quantstamp to make sure we understand the size, scope, and functionality of the smart contract.
   ii. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
   iii. Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to Quantstamp describe.
2. Testing and automated analysis that includes the following:
   i. Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
   ii. Symbolic execution, which is analyzing a program to determine what inputs cause each part of a program to execute.
3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4. Specific, itemized, and actionable recommendations to help you take steps to secure your smart contracts.

Findings

QSP-1 Unhandled Overflows

Severity: *Informational*

Status: Acknowledged

File(s) affected: `contracts/utils/LibBytes.sol`

Description: As pointed by an internal code review of the 0xSequence wallet contributors, there are unhandled possible overflows on lines `LibBytes.sol#146` and `LibBytes.sol#185`. Quantstamp confirmed that the calls currently made to the functions in question cannot reach the overflow state, this does not need to be the case for future uses of these function.

Recommendation: Quantstamp recommends checking overflows and reverting if they happen.

QSP-2 Missing Return Values

Severity: *Informational*

Status: Acknowledged

File(s) affected: `contracts/modules/commons/ModuleAuth.sol`

Description: Function `isValidSignature` in `ModuleAuth` is missing a return statement for the else branch. While this does not pose a problem in the current system, a future potential caller attempting to decode the returned byte value may fail.

Recommendation: Quantstamp recommends explicitly returning a value when the signature validation fails.

Adherence to Specification

The code adheres to the specification and ERC1271.

Code Documentation

The code adheres to the documentation.

Adherence to Best Practices

- LibBytes.sol#L93: We suggest renaming `isValidSignature()` to `isValidDynamicSignature()` to better capture its purpose without clashing with naming that is not related to dynamic signatures.

- Some simplification of `readyBytes()` appears possible. The value after the byte array can be lifted before the iteration, and then the iteration can handle the entire array:

```
function readBytes(
  bytes memory data,
  uint256 index,
  uint256 size
) internal pure returns (bytes memory a, uint256 newIndex) {
  a = new bytes(size);
  assembly {
    let offset := add(32, add(data, index))
    // Load word after new array
    let suffix := add(a, add(32, size))
    let suffixWord := mload(suffix)
    let i := 0 let n := 32
    // Copy each word, INCLUDING last one
    for { } lt(i, size) { i := n n := add(n, 32) } {
      mstore(add(a, n), mload(add(offset, i)))
    }
    // Restore after array
    mstore(suffix, suffixWord)
    newIndex := add(index, size)
  }
  require(newIndex <= data.length, "LibBytes#readBytes: OUT_OF_BOUNDS");
}
```

Test Results

Test Suite Results

The test suite is comprehensive and adequate.

```
Contract: ERC165
  Implement all interfaces for ERC165 on MainModule
     ✓ Should return implements IModuleHooks interfaceId (125ms)
     ✓ Should return implements IERC223Receiver interfaceId (83ms)
     ✓ Should return implements IERC721Receiver interfaceId (69ms)
     ✓ Should return implements IERC1155Receiver interfaceId (67ms)
     ✓ Should return implements IERC1271Wallet interfaceId (67ms)
     ✓ Should return implements IModuleCalls interfaceId (70ms)
     ✓ Should return implements IModuleCreator interfaceId (65ms)
     ✓ Should return implements IModuleHooks interfaceId (60ms)
     ✓ Should return implements IModuleUpdate interfaceId (62ms)
  Implement all interfaces for ERC165 on MainModuleUpgradable
     ✓ Should return implements IModuleHooks interfaceId (79ms)
     ✓ Should return implements IERC223Receiver interfaceId (78ms)
     ✓ Should return implements IERC721Receiver interfaceId (66ms)
     ✓ Should return implements IERC1155Receiver interfaceId (84ms)
     ✓ Should return implements IERC1271Wallet interfaceId (96ms)
     ✓ Should return implements IModuleCalls interfaceId (87ms)
     ✓ Should return implements IModuleCreator interfaceId (170ms)
     ✓ Should return implements IModuleHooks interfaceId (99ms)
     ✓ Should return implements IModuleUpdate interfaceId (75ms)
     ✓ Should return implements IModuleAuthUpgradable interfaceId (87ms)
  Manually defined interfaces
     ✓ Should implement ERC165 interface (46ms)
     ✓ Should implement ERC721 interface (50ms)
     ✓ Should implement ERC1155 interface (59ms)

Contract: Factory
  Deploy wallets
     ✓ Should deploy wallet (52ms)
     ✓ Should predict wallet address (77ms)
     ✓ Should initialize with main module (141ms)

Contract: GuestModule
  GuestModule wallet
     ✓ Should accept transactions without signature (203ms)
     ✓ Should accept transactions on selfExecute (191ms)
     ✓ Should accept transactions with random signature (181ms)
     ✓ Should accept transactions with random nonce (178ms)
     ✓ Should revert on delegateCall transactions (99ms)
     ✓ Should not accept ETH (93ms)
     ✓ Should not implement hooks (76ms)
     ✓ Should not be upgradeable (407ms)

Contract: LibBytes
  readFirstUint16
     ✓ Should read first uint16
     ✓ Should read first uint16 of 2 byte array (47ms)
     ✓ Should fail first uint16 out of bounds (45ms)
  readUint8Uint8
     ✓ Should read bool and uint8 at index zero (39ms)
     ✓ Should read bool and uint8 at given index (56ms)
     ✓ Should read bool and uint8 at last index (69ms)
     ✓ Should fail read bool and uint8 out of bounds (65ms)
     ✓ Should fail read bool and uint16 fully out of bounds (80ms)
  readAddress
     ✓ Should read address at index zero (41ms)
     ✓ Should read address at given index (49ms)
     ✓ Should read address at last index (40ms)
```

```
    ✓ Should fail read address out of bounds
    ✓ Should fail read address totally out of bounds (69ms)
  readBytes66
    ✓ Should read bytes66 at index zero (175ms)
    ✓ Should read bytes66 at given index (105ms)
    ✓ Should read bytes66 at last index (52ms)
    ✓ Should fail read bytes66 out of bounds (44ms)
    ✓ Should fail read bytes66 totally out of bounds
  readBytes32
    ✓ Should read bytes32 at index zero
    ✓ Should read bytes32 at given index
    ✓ Should read bytes32 at last index (44ms)
    ✓ Should fail read bytes32 out of bounds
    ✓ Should fail read bytes32 totally out of bounds
  readUint16
    ✓ Should read uint16 at index zero
    ✓ Should read uint16 at given index (45ms)
    ✓ Should read uint16 at last index
    ✓ Should fail read uint16 out of bounds
    ✓ Should fail read uint16 fully out of bounds (43ms)
  readBytes
    Big bytes
      ✓ Should read bytes at index zero (233ms)
      ✓ Should read bytes at given index (1481ms)
      ✓ Should read bytes at last index (1003ms)
      ✓ Should fail read bytes out of bounds (664ms)
      ✓ Should fail read bytes totally out of bounds (43ms)
    Max bytes
      ✓ Should read bytes at index zero (3785ms)
      ✓ Should read bytes at given index (4097ms)
      ✓ Should read bytes at last index (4223ms)
      ✓ Should fail read bytes out of bounds (686ms)
      ✓ Should fail read bytes totally out of bounds (290ms)
    Small bytes
      ✓ Should read bytes at index zero (400ms)
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds (49ms)
    0 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index (46ms)
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
    1 bytes
      ✓ Should read bytes at index zero (44ms)
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds (49ms)
    2 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds (306ms)
      ✓ Should fail read bytes totally out of bounds (45ms)
    3 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index (38ms)
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds (46ms)
    4 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index (40ms)
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
    5 bytes
      ✓ Should read bytes at index zero (43ms)
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds (50ms)
    6 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds (136ms)
    7 bytes
      ✓ Should read bytes at index zero (257ms)
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
    8 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
    9 bytes
      ✓ Should read bytes at index zero (52ms)
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds (59ms)
    10 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds (43ms)
      ✓ Should fail read bytes totally out of bounds (367ms)
    11 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index (54ms)
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds (46ms)
    12 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds (53ms)
    13 bytes
      ✓ Should read bytes at index zero (42ms)
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index (56ms)
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
    14 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index (74ms)
      ✓ Should read bytes at last index (361ms)
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
    15 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds (43ms)
      ✓ Should fail read bytes totally out of bounds
    16 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index (39ms)
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
    17 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
    18 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds (62ms)
      ✓ Should fail read bytes totally out of bounds (352ms)
    19 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds (70ms)
      ✓ Should fail read bytes totally out of bounds (44ms)
    20 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index (73ms)
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds (45ms)
      ✓ Should fail read bytes totally out of bounds
    21 bytes
```

```
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index (52ms)
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   22 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index (335ms)
      ✓ Should read bytes at last index (55ms)
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   23 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index (42ms)
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds (38ms)
   24 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index (49ms)
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   25 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds (48ms)
   26 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index (72ms)
      ✓ Should read bytes at last index (339ms)
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   27 bytes
      ✓ Should read bytes at index zero (40ms)
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   28 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   29 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   30 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds (65ms)
   31 bytes
      ✓ Should read bytes at index zero (375ms)
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   32 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   33 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index (54ms)
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   34 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   35 bytes
      ✓ Should read bytes at index zero
      ✓ Should fail read bytes totally out of bounds
   35 bytes
      ✓ Should read bytes at index zero (42ms)
      ✓ Should read bytes at given index (288ms)
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds (53ms)
      ✓ Should fail read bytes totally out of bounds
   36 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index (49ms)
      ✓ Should read bytes at last index (52ms)
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   37 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   38 bytes
      ✓ Should read bytes at index zero (48ms)
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   39 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index (60ms)
      ✓ Should fail read bytes out of bounds (266ms)
      ✓ Should fail read bytes totally out of bounds
   40 bytes
      ✓ Should read bytes at index zero (50ms)
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index (62ms)
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   41 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index (74ms)
      ✓ Should read bytes at last index (43ms)
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   42 bytes
      ✓ Should read bytes at index zero (44ms)
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   43 bytes
      ✓ Should read bytes at index zero (47ms)
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index (60ms)
      ✓ Should fail read bytes out of bounds (267ms)
      ✓ Should fail read bytes totally out of bounds
   44 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   45 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   46 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   47 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   48 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index (66ms)
      ✓ Should read bytes at last index (57ms)
      ✓ Should fail read bytes out of bounds (199ms)
```

```
      ✓ Should fail read bytes totally out of bounds
   49 bytes
      ✓ Should read bytes at index zero (38ms)
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds (48ms)
   50 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   51 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   52 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index (39ms)
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   53 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds (38ms)
   54 bytes
      ✓ Should read bytes at index zero (208ms)
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds (45ms)
      ✓ Should fail read bytes totally out of bounds
   55 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   56 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   57 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds (46ms)
   58 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index (54ms)
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   59 bytes
      ✓ Should read bytes at index zero (77ms)
      ✓ Should read bytes at given index (92ms)
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   60 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index (59ms)
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   61 bytes
      ✓ Should read bytes at index zero (46ms)
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   62 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   63 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index (63ms)
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   64 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   65 bytes
      ✓ Should read bytes at index zero (104ms)
      ✓ Should read bytes at given index (40ms)
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   66 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds (170ms)
      ✓ Should fail read bytes totally out of bounds (138ms)
   67 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds (43ms)
      ✓ Should fail read bytes totally out of bounds
   68 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds (50ms)
      ✓ Should fail read bytes totally out of bounds
   69 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index (63ms)
      ✓ Should read bytes at last index (109ms)
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds (47ms)
   70 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   71 bytes
      ✓ Should read bytes at index zero (93ms)
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   72 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index (44ms)
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   73 bytes
      ✓ Should read bytes at index zero (49ms)
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds (38ms)
   74 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds (115ms)
   75 bytes
      ✓ Should read bytes at index zero (39ms)
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   76 bytes
      ✓ Should read bytes at index zero (39ms)
      ✓ Should read bytes at given index
```

```
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   77 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds (53ms)
   78 bytes
      ✓ Should read bytes at index zero (42ms)
      ✓ Should read bytes at given index (80ms)
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds (53ms)
   79 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   80 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index (48ms)
      ✓ Should read bytes at last index (93ms)
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   81 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index (76ms)
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   82 bytes
      ✓ Should read bytes at index zero (66ms)
      ✓ Should read bytes at given index (83ms)
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   83 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index (45ms)
      ✓ Should read bytes at last index (60ms)
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds (54ms)
   84 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds (57ms)
      ✓ Should fail read bytes totally out of bounds (111ms)
   85 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   86 bytes
      ✓ Should read bytes at index zero (43ms)
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds (50ms)
   87 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   88 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   89 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   90 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   91 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index (39ms)
      ✓ Should read bytes at last index (117ms)
      ✓ Should fail read bytes out of bounds (39ms)
      ✓ Should fail read bytes totally out of bounds
   92 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index (40ms)
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds (39ms)
   93 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   94 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   95 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   96 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   97 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds (134ms)
      ✓ Should fail read bytes totally out of bounds (191ms)
   98 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds (40ms)
   99 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index (39ms)
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   100 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   101 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index (50ms)
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   102 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
   103 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index (48ms)
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds (119ms)
      ✓ Should fail read bytes totally out of bounds
   104 bytes
```

```
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index (49ms)
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
    105 bytes
      ✓ Should read bytes at index zero (39ms)
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
    106 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
    107 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
    108 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds (42ms)
      ✓ Should fail read bytes totally out of bounds
    109 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index (41ms)
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds (52ms)
      ✓ Should fail read bytes totally out of bounds (115ms)
    110 bytes
      ✓ Should read bytes at index zero (42ms)
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index (40ms)
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
    111 bytes
      ✓ Should read bytes at index zero (44ms)
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index (45ms)
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
    112 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds (39ms)
      ✓ Should fail read bytes totally out of bounds (71ms)
    113 bytes
      ✓ Should read bytes at index zero (83ms)
      ✓ Should read bytes at given index (63ms)
      ✓ Should read bytes at last index (236ms)
      ✓ Should fail read bytes out of bounds (122ms)
      ✓ Should fail read bytes totally out of bounds
    114 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index (44ms)
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
    115 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
    116 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
    117 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
    118 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds (43ms)
    119 bytes
      ✓ Should read bytes at index zero (38ms)
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds (118ms)
      ✓ Should fail read bytes totally out of bounds (59ms)
    120 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds (42ms)
      ✓ Should fail read bytes totally out of bounds
    121 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index (58ms)
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds (38ms)
      ✓ Should fail read bytes totally out of bounds
    122 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index (74ms)
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
    123 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds (51ms)
    124 bytes
      ✓ Should read bytes at index zero (68ms)
      ✓ Should read bytes at given index (72ms)
      ✓ Should read bytes at last index (52ms)
      ✓ Should fail read bytes out of bounds (87ms)
      ✓ Should fail read bytes totally out of bounds
    125 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
    126 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds (40ms)
    127 bytes
      ✓ Should read bytes at index zero (45ms)
      ✓ Should read bytes at given index
      ✓ Should read bytes at last index (74ms)
      ✓ Should fail read bytes out of bounds (40ms)
      ✓ Should fail read bytes totally out of bounds (64ms)
    128 bytes
      ✓ Should read bytes at index zero (52ms)
      ✓ Should read bytes at given index (60ms)
      ✓ Should read bytes at last index (69ms)
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds
    129 bytes
      ✓ Should read bytes at index zero
      ✓ Should read bytes at given index (105ms)
      ✓ Should read bytes at last index (49ms)
      ✓ Should fail read bytes out of bounds
      ✓ Should fail read bytes totally out of bounds

  Contract: MainModule
    Nested signatures
      ✓ Should accept simple nested signed ERC1271 message (246ms)
      ✓ Should accept simple nested signer (508ms)
      ✓ Should accept two nested signers (874ms)
      ✓ Should accept mixed nested and eoa signers (546ms)
      ✓ Should handle 2 nested sequence wallets (450ms)
      ✓ Should handle 64 nested sequence wallets (17271ms)
      ✓ Should handle 97 nested sequence wallets (31883ms)
      ✓ Should handle binary tree of sequence wallets (6504ms)
```

```
    ✓ Should handle ternary tree of sequence wallets (13211ms)
    ✓ Should handle hexary tree of sequence wallets (14130ms)
    ✓ Should handle random tree of sequence wallets (depth 1) (519ms)
    ✓ Should handle random tree of sequence wallets (depth 2) (685ms)
    ✓ Should handle random tree of sequence wallets (depth 3) (1084ms)
    ✓ Should handle random tree of sequence wallets (depth 4) (3188ms)
    ✓ Should reject invalid nested signature (827ms)
    ✓ Should enforce threshold on nested sigantures (695ms)
    ✓ Should read weight of nested wallets (1409ms)
  Authentication
    ✓ Should accept initial owner signature (163ms)
    ✓ Should reject non-owner signature (268ms)
    ✓ Should reject signature with invalid flag (100ms)
  Network ID
     ✓ Should reject a transaction of another network id (262ms)
  Nonce
    Using non-encoded nonce
       ✓ Should default to space zero (145ms)
       ✓ Should work with zero as initial nonce (195ms)
       ✓ Should emit NonceChange event (372ms)
       ✓ Should fail if nonce did not change (246ms)
       ✓ Should fail if nonce increased by two (352ms)
    using 0x00 space
       ✓ Should work with zero as initial nonce (164ms)
       ✓ Should emit NonceChange event (204ms)
       ✓ Should accept next nonce (210ms)
       ✓ Should fail if nonce did not change (252ms)
       ✓ Should fail if nonce increased by two (761ms)
       ✓ Should use nonces storage keys (82ms)
    using 0x01 space
       ✓ Should work with zero as initial nonce (300ms)
       ✓ Should emit NonceChange event (308ms)
       ✓ Should accept next nonce (177ms)
       ✓ Should fail if nonce did not change (446ms)
       ✓ Should fail if nonce increased by two (342ms)
       ✓ Should use nonces storage keys (137ms)
    using 0x1cae space
       ✓ Should work with zero as initial nonce (103ms)
       ✓ Should emit NonceChange event (712ms)
       ✓ Should accept next nonce (1099ms)
       ✓ Should fail if nonce did not change (659ms)
       ✓ Should fail if nonce increased by two (169ms)
       ✓ Should use nonces storage keys (471ms)
    using 0xb3f342189345e432b29d8d5874f389a4ebd68d40 space
       ✓ Should work with zero as initial nonce (132ms)
       ✓ Should emit NonceChange event (810ms)
       ✓ Should accept next nonce (761ms)
       ✓ Should fail if nonce did not change (802ms)
       ✓ Should fail if nonce increased by two (156ms)
       ✓ Should use nonces storage keys (411ms)
    using 0xffffffffffffffffffffffffffffffffffffffff space
       ✓ Should work with zero as initial nonce (121ms)
       ✓ Should emit NonceChange event (809ms)
       ✓ Should accept next nonce (605ms)
       ✓ Should fail if nonce did not change (194ms)
       ✓ Should fail if nonce increased by two (345ms)
       ✓ Should use nonces storage keys (137ms)
    using two spaces simultaneously
       ✓ Should keep separated nonce counts (996ms)
       ✓ Should emit different events (1176ms)
       ✓ Should not accept nonce of different space (259ms)
  Upgradeability
    ✓ Should update implementation (428ms)
    ✓ Should fail to set implementation to address 0 (205ms)
    ✓ Should fail to set implementation to non-contract (434ms)
    ✓ Should use implementation storage key (182ms)
  External calls
    ✓ Should perform call to contract (335ms)
    ✓ Should return error message (395ms)
  Batch transactions
     ✓ Should perform multiple calls to contracts in one tx (372ms)
     ✓ Should perform call a contract and transfer eth in one tx (950ms)
     ✓ Should fail if one transaction fails (335ms)
  Delegate calls
    ✓ Should delegate call to module (745ms)
  on delegate call revert
     ✓ Should pass if delegate call is optional (679ms)
     ✓ Should fail if delegate call fails (599ms)
  Handle ETH
    ✓ Should receive ETH (44ms)
    ✓ Should transfer ETH (189ms)
    ✓ Should call payable function (383ms)
  Optional transactions
    ✓ Should skip a skipOnError transaction (770ms)
    ✓ Should skip failing transaction within batch (758ms)
    ✓ Should skip multiple failing transactions within batch (444ms)
    ✓ Should skip all failing transactions within batch (357ms)
    ✓ Should skip skipOnError update implementation action (389ms)
  Hooks
    receive tokens
       ✓ Should implement ERC1155 single transfer hook (45ms)
       ✓ Should implement ERC1155 batch transfer hook (110ms)
       ✓ Should implement ERC721 transfer hook (55ms)
       ✓ Should implement ERC223 transfer hook (90ms)
    ERC1271 Wallet
       ✓ Should validate arbitrary signed data (95ms)
       ✓ Should validate arbitrary signed hash (53ms)
       ✓ Should reject data signed by non-owner (67ms)
       ✓ Should reject hash signed by non-owner (117ms)
    External hooks
       ✓ Should read added hook (151ms)
       ✓ Should return zero if hook is not registered (114ms)
       ✓ Should forward call to external hook (263ms)
       ✓ Should not forward call to deregistered hook (325ms)
       ✓ Should pass calling a non registered hook (255ms)
       ✓ Should use hooks storage key (153ms)
  Require configuration
     ✓ Should require configuration of a non-deployed wallet (482ms)
     ✓ Should require configuration of a non-updated wallet (203ms)
     ✓ Should fail to require configuraiton of a non-deployed wallet (423ms)
     ✓ Should fail to require configuration of a non-updated wallet (513ms)
  Update owners
    After a migration
       ✓ Should implement new upgradable module
       ✓ Should accept new owner signature (103ms)
       ✓ Should reject old owner signature (646ms)
       ✓ Should fail to update to invalid image hash (149ms)
       ✓ Should fail to change image hash from non-self address (341ms)
       ✓ Should use image hash storage key
       ✓ Should fail to execute transactions on moduleUpgradable implementation (83ms)
       ✓ Should update wallet and require configuration (450ms)
       ✓ Should fail to update wallet and require wrong configuration (893ms)
      After updating the image hash
         ✓ Should have updated the image hash
         ✓ Should accept new owners signatures (317ms)
         ✓ Should reject old owner signatures (156ms)
         ✓ Should use image hash storage key
  Multisignature
    Forced dynamic signature encoding
      With 1/2 wallet
         ✓ Should accept signed message by first owner (578ms)
         ✓ Should accept signed message by second owner (121ms)
         ✓ Should accept signed message by both owners (132ms)
         ✓ Should reject message without signatures (131ms)
         ✓ Should reject message signed by non-owner (144ms)
      With 2/2 wallet
         ✓ Should accept signed message by both owners (190ms)
         ✓ Should reject message without signatures (371ms)
         ✓ Should reject message signed only by first owner (184ms)
         ✓ Should reject message signed only by second owner (277ms)
         ✓ Should reject message signed by non-owner (288ms)
      With 2/3 wallet
         ✓ Should accept signed message by first and second owner (636ms)
         ✓ Should accept signed message by first and last owner (126ms)
         ✓ Should accept signed message by second and last owner (340ms)
         ✓ Should accept signed message by all owners (188ms)
         ✓ Should reject message signed only by first owner (174ms)
         ✓ Should reject message signed only by second owner (153ms)
         ✓ Should reject message signed only by last owner (302ms)
         ✓ Should reject message not signed (355ms)
         ✓ Should reject message signed by non-owner (194ms)
         ✓ Should reject message if the image lacks an owner (413ms)
      With 255/255 wallet
         ✓ Should accept message signed by all owners (6894ms)
         ✓ Should reject message signed by non-owner (11166ms)
         ✓ Should reject message missing a signature (11419ms)
      With weighted owners
         ✓ Should accept signed message with (3+1)/4 weight (128ms)
         ✓ Should accept signed message with (3+3)/4 weight (125ms)
         ✓ Should accept signed message with (3+3+1+1)/4 weight (152ms)
         ✓ Should accept signed message with (3+3+1+1+1)/4 weight (361ms)
         ✓ Should reject signed message with (1)/4 weight (154ms)
         ✓ Should reject signed message with (1+1)/4 weight (192ms)
         ✓ Should reject signed message with (1+1+1)/4 weight (211ms)
         ✓ Should reject signed message with (3)/4 weight (230ms)
         ✓ Should reject signed message with (0)/4 weight (160ms)
```

```
      ✓ Should reject message signed by non-owner (264ms)
    Reject invalid signatures
      ✓ Should reject invalid signature type (125ms)
      ✓ Should reject invalid s value (107ms)
      ✓ Should reject invalid v value (113ms)
  Default signature encoding
    With 1/2 wallet
      ✓ Should accept signed message by first owner (115ms)
      ✓ Should accept signed message by second owner (118ms)
      ✓ Should accept signed message by both owners (254ms)
      ✓ Should reject message without signatures (157ms)
      ✓ Should reject message signed by non-owner (220ms)
    With 2/2 wallet
      ✓ Should accept signed message by both owners (128ms)
      ✓ Should reject message without signatures (128ms)
      ✓ Should reject message signed only by first owner (179ms)
      ✓ Should reject message signed only by second owner (252ms)
      ✓ Should reject message signed by non-owner (193ms)
    With 2/3 wallet
      ✓ Should accept signed message by first and second owner (142ms)
      ✓ Should accept signed message by first and last owner (171ms)
      ✓ Should accept signed message by second and last owner (152ms)
      ✓ Should accept signed message by all owners (178ms)
      ✓ Should reject message signed only by first owner (263ms)
      ✓ Should reject message signed only by second owner (185ms)
      ✓ Should reject message signed only by last owner (198ms)
      ✓ Should reject message not signed (184ms)
      ✓ Should reject message signed by non-owner (232ms)
      ✓ Should reject message if the image lacks an owner (382ms)
    With 255/255 wallet
      ✓ Should accept message signed by all owners (6453ms)
      ✓ Should reject message signed by non-owner (10887ms)
      ✓ Should reject message missing a signature (9944ms)
    With weighted owners
      ✓ Should accept signed message with (3+1)/4 weight (125ms)
      ✓ Should accept signed message with (3+3)/4 weight (114ms)
      ✓ Should accept signed message with (3+3+1+1)/4 weight (147ms)
      ✓ Should accept signed message with (3+3+1+1+1)/4 weight (152ms)
      ✓ Should reject signed message with (1)/4 weight (509ms)
      ✓ Should reject signed message with (1+1)/4 weight (190ms)
      ✓ Should reject signed message with (1+1+1)/4 weight (550ms)
      ✓ Should reject signed message with (3)/4 weight (192ms)
      ✓ Should reject signed message with (0)/4 weight (190ms)
      ✓ Should reject message signed by non-owner (271ms)
    Reject invalid signatures
      ✓ Should reject invalid signature type (121ms)
      ✓ Should reject invalid s value (120ms)
      ✓ Should reject invalid v value (94ms)
  Gas limit
    ✓ Should forward the defined amount of gas (138ms)
    ✓ Should forward different amounts of gas (2116ms)
    ✓ Should fail if forwarded call runs out of gas
    ✓ Should fail without reverting if optional call runs out of gas (178ms)
    ✓ Should continue execution if optional call runs out of gas (789ms)
    ✓ Should fail if transaction is executed with not enough gas (109ms)
  Create contracts
    ✓ Should create a contract (438ms)
    ✓ Should create a contract with value (172ms)
    ✓ Should fail to create a contract from non-self (58ms)
  Transaction events
    ✓ Should emit TxExecuted event (107ms)
    ✓ Should emit multiple TxExecuted events (371ms)
  Internal bundles
    ✓ Should execute internal bundle (436ms)
    ✓ Should execute multiple internal bundles (1550ms)
    ✓ Should execute nested internal bundles (707ms)
    ✓ Should revert bundle without reverting transaction (560ms)

Contract: Multi call utils
  Call multiple contracts
    ✓ Should execute empty call (44ms)
    ✓ Should execute single call (130ms)
    ✓ Should execute two calls (379ms)
    ✓ Should execute calls to multiple contracts (396ms)
    ✓ Return other calls even if single call fails (457ms)
    ✓ Fail if call with revert on error fails (316ms)
    ✓ Fail if batch includes delegate call (200ms)
    ✓ Fail if not enough gas for call (344ms)
    ✓ Should call globals (309ms)

Contract: Require utils
  Require min-nonce
    ✓ Should pass nonce increased from self-wallet (1296ms)
    ✓ Should pass nonce increased from different wallet (1109ms)
    ✓ Should fail if nonce is below required on different wallet (1054ms)
    ✓ Should fail if nonce is below required on self-wallet on a different space (568ms)
    ✓ Should fail if nonce is below required on self-wallet (275ms)
  Expirable transactions
    ✓ Should pass if non expired
    ✓ Should fail if expired (42ms)
    ✓ Should pass bundle if non expired (595ms)
    ✓ Should fail bundle if expired (271ms)


941 passing (6m)
```

Code Coverage

The code features very good test coverage.

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
contracts/	100	100	100	100	
Factory.sol	100	100	100	100	
Wallet.sol	100	100	100	100	
contracts/interfaces/	100	100	100	100	
IERC1271Wallet.sol	100	100	100	100	
contracts/interfaces/receivers/	100	100	100	100	
IERC1155Receiver.sol	100	100	100	100	
IERC223Receiver.sol	100	100	100	100	
IERC721Receiver.sol	100	100	100	100	
contracts/modules/	88.89	100	75	88.89	
GuestModule.sol	87.5	100	60	87.5	101,116
MainModule.sol	100	100	100	100	
MainModuleUpgradable.sol	100	100	100	100	
contracts/modules/commons/	100	95.83	97.67	99.12	
Implementation.sol	100	100	50	50	27
ModuleAuth.sol	100	100	100	100	
ModuleAuthFixed.sol	100	100	100	100	
ModuleAuthUpgradable.sol	100	100	100	100	
ModuleCalls.sol	100	100	100	100	
ModuleCreator.sol	100	100	100	100	
ModuleERC165.sol	100	100	100	100	
ModuleHooks.sol	100	75	100	100	
ModuleSelfAuth.sol	100	100	100	100	
ModuleStorage.sol	100	100	100	100	
ModuleUpdate.sol	100	100	100	100	
contracts/modules/commons/interfaces/	100	100	100	100	
IModuleAuth.sol	100	100	100	100	
IModuleAuthUpgradable.sol	100	100	100	100	
IModuleCalls.sol	100	100	100	100	
IModuleCreator.sol	100	100	100	100	
IModuleHooks.sol	100	100	100	100	
IModuleUpdate.sol	100	100	100	100	
contracts/modules/utils/	100	100	95.24	100	
MultiCallUtils.sol	100	100	100	100	
RequireUtils.sol	100	100	100	100	
SequenceUtils.sol	100	100	0	100	
contracts/utils/	97.14	92.86	100	97.56	
LibAddress.sol	100	100	100	100	
LibBytes.sol	100	100	100	100	
SignatureValidator.sol	95.45	85.71	100	95	75
All files	**98.49**	**95.92**	**95.18**	**98.13**	

Appendix

File Signatures

The following are the SHA-256 hashes of the reviewed files. A file with a different SHA-256 hash has been modified, intentionally or otherwise, after the security review. You are cautioned that a different SHA-256 hash could be (but is not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of the review.

Contracts

```
1954f3883db44f020b2e25aaaefc9fe39a9be36cba21162b6615ba5d86639dc2 ./contracts/Factory.sol
46eda621b9c822e2f8df7d695dd7ebe270d425451daff8178ab9dcf34d6f7665 ./contracts/Wallet.sol
76a68bb06d7e554b2933cf403ab19054ff1f3de82e93ecde72970e789fb3a1e7 ./contracts/interfaces/IERC1271Wallet.sol
b82f74ee2e7f986631cbb7aea5e274ddc0c6f01285db11b3bc92d44942cdff5d ./contracts/interfaces/receivers/IERC223Receiver.sol
307b149327c907abd9332cdc83acfc5b4bed690f61b02b024c8f6fa464e31812 ./contracts/interfaces/receivers/IERC1155Receiver.sol
02599760d4c296aea9081bd0ccd1ba2a37208b49a716fce7952272da9ffe2fde ./contracts/interfaces/receivers/IERC721Receiver.sol
4d39090af8432341041cbfda16285aec9b700611ace7c7ed95d9fa15650b7462 ./contracts/modules/GuestModule.sol
935e3ef2b58f7cfd4b95cdf9f06c0c15b7e740d1b916a919e4017b8b468eab41 ./contracts/modules/MainModule.sol
c5fc385d24223b0d286ee8845620bf8a71e3f543eaf639c219ad4b08ffe81ca0 ./contracts/modules/MainModuleUpgradable.sol
37faa4ad3f33ae505b8386ca021e560741cfa4938c30b85b819023b4c0e9ae2f ./contracts/modules/utils/SequenceUtils.sol
b8ae986b20f544b2abfee1b4b829dbd2eeb2f759d8d2e1042baf1ff5b357c833 ./contracts/modules/utils/RequireUtils.sol
c03ca5dc3ac2ae395bb1729753bd7463e3d189d3ad84557237fec06eb27aaec4 ./contracts/modules/utils/MultiCallUtils.sol
7e3ea36359e465eb940ffa1b8641c5530e53343a80470d5bf829c124f1fd0a22 ./contracts/modules/commons/ModuleCalls.sol
0e598ae0f57d4e0f17ac851b2453e1554d483c659289306b3f344145a463a6d8 ./contracts/modules/commons/ModuleCreator.sol
135e921611d048c796f62976e36b855d32e60a26fbe1490bc84c827dcd65bb4a ./contracts/modules/commons/ModuleAuth.sol
0badd65b4d84342d59d43c6cfde70fde4500757ca373c276c0e53a0f4f0dc687 ./contracts/modules/commons/ModuleERC165.sol
b3ee9651eec268697a4d4ae290a288ce7339433e266956ab7593fd1c85e5149d ./contracts/modules/commons/ModuleHooks.sol
9d112cdfe16332755772f3e82c3749a7a4faedd91e960cc0b3c53b8d851524d9 ./contracts/modules/commons/ModuleStorage.sol
47c243c703f71b483e98468053ea7bde7fb976417d84fe546e68f00b58215156 ./contracts/modules/commons/Implementation.sol
51396b9700f882c9e7c5459f1c7bc5e79678fa28ca25282743fd53d751b47d91 ./contracts/modules/commons/ModuleUpdate.sol
9fde9d637946d38ef29b8fe2b6f754618e019dd8fca2c0f3c25d77a61b7941e4 ./contracts/modules/commons/ModuleSelfAuth.sol
831ef918a6db5b3b75c39287189a90f7265bf078a2820aa0c1857b4206762a7a ./contracts/modules/commons/ModuleAuthFixed.sol
1a65f734b3a5d20cfa1cc6ed24f8b859ea088bbedfdb1e99c39e7a3f85971f48 ./contracts/modules/commons/ModuleAuthUpgradable.sol
f864f393563a55894037af7035c3638f1adca6e5744b7e13df7608676add3b6b ./contracts/modules/commons/interfaces/IModuleHooks.sol
b9a5ee32c743de5100f5e7fddbb1e157c1955f92e39b2030fc4eede14c4bcd9a ./contracts/modules/commons/interfaces/IModuleCalls.sol
b5572057bde4e6b561121c31a6a02e33f9ea60af03dfd8c6bf58cb34415b932f ./contracts/modules/commons/interfaces/IModuleCreator.sol
bb494cc39ec2e4d86caad62d03f1ecf7a4c34ff5fcaf9fe631ddae63e921dec7 ./contracts/modules/commons/interfaces/IModuleUpdate.sol
841881ecb4f93f84ae1658f68484e7f5eb91cfd5cf6d228707b2f5c03eb381d7 ./contracts/modules/commons/interfaces/IModuleAuthUpgradable.sol
16fa09ba4dccf2a820424f13342f85ee8524ec99c710fbbd4029bb3a877e8ff0 ./contracts/modules/commons/interfaces/IModuleAuth.sol
80b33f7aa7547fa699d4e970d61666769824a2e59d620b96d9471444eb9701ca ./contracts/utils/LibAddress.sol
c8fd3ca7e12098bdc425d685f0489b256f1059909bde81387811f35531923735 ./contracts/utils/SignatureValidator.sol
70bc383cc63cdaee1704722020689b1512d9512a73ec573125efe664c157ff2e ./contracts/utils/LibBytes.sol
b62789b5726966c39197d502ae45938fcd05134c704aa3416e5111a1a91e22d5 ./contracts/migrations/Migrations.sol
9c98620dd5fb7f83a511a535550991e604044520728d434adb2031228027acf5 ./contracts/mocks/CallReceiverMock.sol
39029e409f5cca6e4281e939250c4b7c87010b1e67c7bab041c24a38418be40b ./contracts/mocks/LibBytesImpl.sol
8033085e2b64f7139e29c0167e373b2b4b379462ae24ef6216a15797111f41c1 ./contracts/mocks/HookCallerMock.sol
419a7bed478b7981500ccafd30db7928f4f5b7199a12b3ee3ecbb4c6a3f6d14f ./contracts/mocks/GasBurnerMock.sol
0a9a89997e39560593cd2661c69fa33c458550bd93453e9e2a6331913af9f59d ./contracts/mocks/DelegateCallMock.sol
3cf61dbb607d0dc109edf01b60f421982fc4df91dc62739d6f30431f82ae8ab9 ./contracts/mocks/HookMock.sol
d6acd66a6e06800f7e9844076449077c15777da5f79c2238c471fd62edf3bcb0 ./contracts/mocks/ModuleMock.sol
c098a83ad7ad211c73116f6bd92c109dc8385e432cb0b6b0086e9b45ffe138d1 ./contracts/mocks/ERC165CheckerMock.sol
```

Tests

```
ad220c634ce4dc91b0e9929de802d976d277b2eb94158d1b98fb6896c2b86262 ./tests/LibBytes.spec.ts
dbc7e47e3f3d938594c81806dab10f6100bd9270a01aec6b06ab994cecf24971 ./tests/RequireUtils.spec.ts
2f2daf6c4d57efc18fadaa8b763dfdc805dfc2e5859038c308aa740a7f55c4a1 ./tests/MainModule.spec.ts
97e79c97fab2543a2a4e7732833df48df0c09678a969ef345e740d08b0988ca0 ./tests/MultiCallUtils.spec.ts
184d58a1b193765d9038ec0885fca4e67dd4934e32d34746e738ac3c49a25576 ./tests/MainModule.bench.ts
961a9ba5fb12c95e7d2a96d9a3aa17242bc9fdf275e9cb626a4817114beb1d52 ./tests/ERC165.spec.ts
64d8081d3e14142456c623d216f990f83c8cf885d12c5fc697119c489841dfe9 ./tests/GuestModule.spec.ts
fea38e1e42ea98d288a4eb68b881346b234fb6dbd7d26cbcfd5056c0ebe773d1 ./tests/Factory.spec.ts
6d5c37e4ea1cf43408122b6196ddb650c76d73a648b0204547b78d370b57b361 ./tests/utils/helpers.ts
56cb327ecd10477a223ab6b60318d2cf077c64e422cd2aff87c7d9c11d603e49 ./tests/utils/contract.ts
ad6f227a7d5546055690378e7233b3ddb5b2144c9aff16aad100b3fa919f845c ./tests/utils/index.ts
```

Changelog

- 2021-02-08 - Initial report
- 2021-02-18 - Report finalization after acknowledgement of issues

About Quantstamp

Quantstamp is a Y Combinator-backed company that helps to secure blockchain platforms at scale using computer-aided reasoning tools, with a mission to help boost the adoption of this exponentially growing technology.

With over 1000 Google scholar citations and numerous published papers, Quantstamp's team has decades of combined experience in formal verification, static analysis, and software verification. Quantstamp has also developed a protocol to help smart contract developers and projects worldwide to perform cost-effective smart contract security scans.

To date, Quantstamp has protected $5B in digital asset risk from hackers and assisted dozens of blockchain projects globally through its white glove security assessment services. As an evangelist of the blockchain ecosystem, Quantstamp assists core infrastructure projects and leading community initiatives such as the Ethereum Community Fund to expedite the adoption of blockchain technology.

Quantstamp's collaborations with leading academic institutions such as the National University of Singapore and MIT (Massachusetts Institute of Technology) reflect our commitment to research, development, and enabling world-class blockchain security.

Timeliness of content

The content contained in the report is current as of the date appearing on the report and is subject to change without notice, unless indicated otherwise by Quantstamp; however, Quantstamp does not guarantee or warrant the accuracy, timeliness, or completeness of any report you access using the internet or other means, and assumes no obligation to update any information following publication.

Notice of confidentiality

This report, including the content, data, and underlying methodologies, are subject to the confidentiality and feedback provisions in your agreement with Quantstamp. These materials are not to be disclosed, extracted, copied, or distributed except to the extent expressly authorized by Quantstamp.

Links to other websites

You may, through hypertext or other computer links, gain access to web sites operated by persons other than Quantstamp, Inc. (Quantstamp). Such hyperlinks are provided for your reference and convenience only, and are the exclusive responsibility of such web sites' owners. You agree that Quantstamp are not responsible for the content or operation of such web sites, and that Quantstamp shall have no liability to you or any other person or entity for the use of third-party web sites. Except as described below, a hyperlink from this web site to another web site does not imply or mean that Quantstamp endorses the content on that web site or the operator or operations of that site. You are solely responsible for determining the extent to which you may use any content at any other web sites to which you link from the report. Quantstamp assumes no responsibility for the use of third-party software on the website and shall have no liability whatsoever to any person or entity for the accuracy or completeness of any outcome generated by such software.

Disclaimer

This report is based on the scope of materials and documentation provided for a limited review at the time provided. Results may not be complete nor inclusive of all vulnerabilities. The review and this report are provided on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. Blockchain technology remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond the programming language, or other programming aspects that could present security risks. A report does not indicate the endorsement of any particular project or team, nor guarantee its security. No third party should rely on the reports in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. To the fullest extent permitted by law, we disclaim all warranties, expressed or implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. We do not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party through the product, any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, and the related services and products, any hyperlinked websites, any websites or mobile applications appearing on any advertising, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third-party providers of products or services. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate. FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.