






Executive Summary

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

Type	Light client, Cross chain protocol	Documentation quality	High	<div><div></div></div>
Timeline	2024-09-23 through 2024-11-11	Test quality	Medium	<div><div></div></div>
Language	Rust, Solidity	Total Findings	7	<div><div></div><div>Fixed: 7</div></div>
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review	High severity findings ⓘ	0	
Specification	Doc Site 	Medium severity findings ⓘ	0	
Source Code	<ul style="list-style-type: none">datachainlab/lcp-solidity  #54d2bac datachainlab/lcp  #4df1e8d 	Low severity findings ⓘ	5	<div><div></div><div>Fixed: 5</div></div>
Auditors	<ul style="list-style-type: none">Andy Lin Senior Auditing EngineerJulio Aguilar Auditing EngineerGereon Mendler Auditing Engineer	Undetermined severity findings ⓘ	0	
		Informational findings ⓘ	2	<div><div></div><div>Fixed: 2</div></div>

Summary of Findings

This project focuses on implementing the Light Client Proxy (LCP), which uses Intel SGX, a Trusted Execution Environment (TEE), to perform light client verification within a secure enclave. This setup ensures a safe verification process by generating cryptographic proofs that other chains can trust through Intel's Remote Attestation mechanism. By using SGX, LCP provides a scalable way for cross-chain communication without needing separate light client implementations for each blockchain pair. The LCP protocol uses a relay to transfer messages between an upstream chain and a downstream chain. Verification results are provided as signed commitments, which help establish trust between chains.

This audit reviewed the LCP node, written in Rust, and the LCP client, implemented as Solidity contracts. The LCP node serves as the main controller of the enclave system by creating an Enclave Key (EK), starting the Enclave Light Client (ELC) process, and signing the ELC results with the EK. The LCP client verifies the LCP node's results by checking the EK signatures to ensure updates are secure and cannot be changed without the enclave's approval. Please note that the ELC is not included in this specific audit, although other teams are reviewing some of their ELC implementations.

The codebase is easy to read and well organized into separate parts. There are documentation websites that clearly explain the main concepts. Both the LCP node and client have tests to verify the code, though there is room to improve overall coverage.

After our review, we believe the team has considered many potential attack methods, and we found very few serious issues. We discussed several potential attack factors, and they all seem to be mitigated or resolved by assumptions about the ELC. For more details, please see the "Operational Considerations" section. We also found some minor issues and have suggestions for improvements. We recommend fixing and enhancing all of these to make the system more secure and reliable.

Fix Review Update: The team has fixed all issues, and also most of the suggestions.

ID	DESCRIPTION	SEVERITY	STATUS
LCP-1	Enclave Key Remains in Memory After Use	<ul style="list-style-type: none">Low ⓘ	Fixed
LCP-2	Usage of Unrecommended <code>parse_overflowing_slice()</code>	<ul style="list-style-type: none">Low ⓘ	Fixed
LCP-3	Not Returning Error or Reverting when Size Overflows	<ul style="list-style-type: none">Low ⓘ	Fixed
LCP-4	Panic Due to Out-of-Boundary Access	<ul style="list-style-type: none">Low ⓘ	Fixed

ID	DESCRIPTION	SEVERITY	STATUS
LCP-5	Dependency Risks	• Low ⓘ	Fixed
LCP-6	Consider Using <code>rand::rng::OsRng</code> for More Robustness	• Informational ⓘ	Fixed
LCP-7	Enclave Leaking Panic Contents	• Informational ⓘ	Fixed

Assessment Breakdown

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

i **Disclaimer**

Only features that are contained within the repositories at the commit hashes specified on the front page of the report are within the scope of the audit and fix review. All features added in future revisions of the code are excluded from consideration in this report.

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

1. Code review that includes the following
 1. Review of the specifications, sources, and instructions provided to Quantstamp to make sure we understand the size, scope, and functionality of the smart contract.
 2. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
 3. 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:
 1. 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.
 2. 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, clarity, 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.

Scope

For the `lcp-solidity` repository, the `contracts/` folder is in scope.

And for the `lcp` repository, only the following files are in scope:

```
./enclave-modules/**
./modules/
├── attestation-report
│   └── src
│       ├── errors.rs
│       ├── lib.rs
│       └── report.rs
├── commitments/*
└── context/*
```

```

├── crypto/*
├── ecall-commands/*
├── light-client
│   └── src
│       ├── client.rs
│       ├── context.rs
│       ├── errors.rs
│       ├── ibc.rs
│       ├── lib.rs
│       ├── path.rs
│       └── registry.rs
├── ocall-commands
│   └── src
│       ├── lib.rs
│       ├── log.rs
│       └── store.rs
├── remote-attestation
│   └── src
│       ├── errors.rs
│       ├── ias.rs
│       ├── ias_utils.rs
│       └── lib.rs
├── store
│   └── src
│       ├── cache.rs
│       ├── errors.rs
│       ├── lib.rs
│       └── store.rs
└── types
    └── src
        ├── any.rs
        ├── errors.rs
        ├── height.rs
        ├── host.rs
        ├── lib.rs
        ├── sgx.rs
        └── time.rs

```

Operational Considerations

1. The LCP architecture assumes trust in Intel's Remote Attestation service (IAS), which acts as a third-party verifier for the integrity of the SGX enclave environment. This reliance on Intel's IAS (Attestation Service) is critical, as it validates the Enclave Key (EK) generated within SGX. It is assumed that the EK, verified through the Attestation Verification Report (AVR), ensures authenticity and prevents malicious actors from forging or substituting the public key used in the verification process.
2. In cases of a chain hard fork, the ELC (Enclave Light Client) is expected to treat unintentional forks that result from consensus failures, such as a fork at a specific block height, as misbehavior. This handling aligns with standards such as ICS-02, where consensus failures (e.g., validators in a PoS system producing multiple blocks at the same height) are treated as errors. When this occurs, the ELC will halt further processing, preventing updates from the forked blocks from being propagated to the LCP client. This behavior helps mitigate risks such as double-spending and malicious withdrawals by ensuring only a single, verified chain state update is applied.
3. The light client registered in the enclave should not include both the original chain and the forked chains (not hard forks for upgrades, but forks like ETH and ETC that create separate chains), as they could have the same state ID, which breaks the uniqueness assumption.
4. We assume that all ELC (Enclave Light Client) implementations must check that the `type_url` of the client state matches the specific ELC that the implementation supports. The current implementation of `ecall-handler/src/light-client/register.rs::get_light_client_by_client_id()` risks accepting tampered data from `ctx.client_state(client_id)` and thus retrieving the wrong light client in the line `ctx.get_light_client(any_client_state.type_url.as_ref())`. Therefore, the ELC itself needs to ensure it is parsing the expected client state data of its own type.
5. We assume that ELCs for PoS chains that rely on validators and the ability to slash to ensure the protocol's robustness must include the "Validation Context" (see: [doc](#)) as part of their light client validation. The `trusting_period` field should not be optional for these ELCs.
6. Any LCP Node can operate multiple enclaves supporting multiple light clients each, but we recommend the creation of distinct enclaves.
7. In this audit, we assume that the ELC (Enclave Light Client) implementation ensures that even with a compromised OCALL result, the ELC must not falsely promote state changes from a canonical state ID to an incorrect one. This requires careful attention in the ELC implementation, as it must withstand the threat model of invalid storage data access.
8. In the `LCPClientBase` contract, the `ibcHandler` is expected to be set as the `IBCCClient` contract (see: [code](#)), which ensures that `initializeClient()` cannot be called with the same client ID multiple times.
9. The `AVRValidator` contract implementation has strong assumptions about the enclave report JSON format. First, the key-value pairs must be encoded in the expected order. Secondly, there cannot be extra spaces between the key, the ":", and the value. These assumptions are not guaranteed by the report structure and are observed in practice. There is a risk that the assumptions can change, causing report parsing to fail.

Key Actors And Their Capabilities

There are not many privileged actors in the LCP node and LCP client. One privileged actor role is the operator. The LCP client can be initiated with operators. If configured, any updates or verifications will require passing the configured threshold of EKs (enclave keys) that are mapped to specific operators. This provides an extra layer of security outside the enclave.

Findings

LCP-1 Enclave Key Remains in Memory After Use

• Low ⓘ Fixed

✓ Update

The team fixed the issue in commit `b8b63b9` with the following changes:

1. Implement the `Drop` trait for `EnclaveKey`.
2. Use `Zeroizing` to wrap on places where unsealed keys will be used. This ensures that those data will be zeroized once dropped.

File(s) affected: `modules/crypto/src/keys.rs`

Description: The enclave key currently remains in memory after it is unsealed and used. Since it represents the critical source of trust in the system, limiting its exposure to hardware level attacks is recommended.

Recommendation: We suggest implementing the following:

1. Override the enclave key memory entry with zeroes after use.
2. Consider implementing the `Drop trait` for the `EnclaveKey` to ensure that the memory data is zeroed out instead of remaining in the memory layout. This ensures that the Rust runtime deconstructs the private key value more safely, avoiding attacks that leak memory from the enclave.

LCP-2 Usage of Unrecommended `parse_overflowing_slice()`

• Low ⓘ Fixed

✓ Update

The team fixed the issue as recommended in commit `5d104a9`.

File(s) affected: `modules/crypto/src/keys.rs`

Description: The `verify_signature()` function uses `Signature::parse_overflowing_slice()`, which is not recommended according to its [code comment](#). The code comment recommends using `parse_standard_slice()` instead.

Recommendation: Consider using the `parse_standard_slice()` function instead, as pointed out in the code comment.

LCP-3 Not Returning Error or Reverting when Size Overflows

• Low ⓘ Fixed

✓ Update

The team fixed the issue as recommended in commit `67dc4a7`.

File(s) affected: `modules/crypto/src/keys.rs`

Description: The `calc_raw_sealed_data_size()` function returns `max` when there is an unexpected size. This is a copy of the implementation from the `sgx_tseal/src/internal.rs`. However, in the Rust SGX SDK, it will error out (see: [code](#)) at the place calling when receiving the `max` output. We suggest that the same should be applied here.

Recommendation: Consider applying an `assert!()` or ensuring the function errors out under the conditions where `add_mac_txt_size > max - encrypt_txt_size` and `payload_size > max - sealed_data_size`.

LCP-4 Panic Due to Out-of-Boundary Access

• Low ⓘ Fixed

✓ Update

The team fixed the issue as recommended in commit `b7c7d33` .

File(s) affected: `LCPUtils.sol`

Description: The `LCPUtils.readBytesUntil()` function calls the `BytesUtils.find()` function. The `BytesUtils.find()` function has the `len` input, which represents the number of bytes to search (see: [code comment](#)). The current implementation passes `src.length` as the `len` input, which will trigger a panic when the needle is not found, as it will attempt to search outside of `src.length` .

There is no real impact feature-wise, as `readBytesUntil()` intends to revert with `LCPUtilsReadBytesUntilNotFound()` when the needle is not found. However, it might make it hard to identify the reason for a failed transaction.

Recommendation: Instead of `BytesUtils.find(src, offset, src.length, needle)` , use `src.length - offset` as the `len` input.

LCP-5 Dependency Risks

• Low ⓘ Fixed

✓ Update

The team fixed the issue with the following commits: `1adaf00` , `6ddc607` , `f5bd737` , `7fe83cc` , `d44f678` , and `8bd0145` .

We re-ran the cargo audit after applying the fixes on the fix review commit `a41fceb6d26` . The code using the dependencies has indeed been upgraded to safe versions, although some risky dependencies are still being pulled in due to being transitive dependencies (dependency of dependency).

Also, the client provided the following explanation:

We consider that the following crates do not need to be fixed because these crates are not used in production:

- `curve25519-dalek`, `ed25519-dalek`, `rustls`, `tungstenite`, `webpki`: only used in testing crate `integration-test` or `nodes-runner` or both
- `rsa`: only used in SW mode(i.e., not production)

File(s) affected: `Cargo.toml`

Description: The [Cargo audit](#) report has identified potential dependency risks. Here is a summary of the risks:

Key Vulnerabilities:

1. `curve25519-dalek` (v3.2.0)

- **Issue:** Timing variability in `Scalar29::sub` and `Scalar52::sub` .
- **Solution:** Upgrade to version `>= 4.1.3`.
- **Link:** [RUSTSEC-2024-0344](#)

2. `ed25519-dalek` (v1.0.1)

- **Issue:** Double Public Key Signing Function Oracle Attack.
- **Solution:** Upgrade to version `>= 2`.
- **Link:** [RUSTSEC-2022-0093](#)

3. `h2` (v0.3.19)

- **Issue:** CONTINUATION Flood vulnerability causing service degradation.
- **Solution:** Upgrade to `^0.3.26` or `>= 0.4.4`.
- **Link:** [RUSTSEC-2024-0332](#)

4. `libgit2-sys` (v0.15.2)

- **Issue:** Memory corruption, denial of service, and arbitrary code execution.
- **Solution:** Upgrade to version `>= 0.16.2`.
- **Link:** [RUSTSEC-2024-0013](#)

5. `rsa` (v0.9.2)

- **Issue:** Timing side-channel vulnerability (Marvin Attack).
- **Solution:** No available fix.
- **Link:** [RUSTSEC-2023-0071](#)

6. `rustls` (v0.19.1 and v0.20.6)

- **Issue:** Infinite loop vulnerability in network input handling.
- **Solution:** Upgrade to specific patched versions.
- **Link:** [RUSTSEC-2024-0336](#)

7. `shlex` (v1.1.0)

- **Issue:** Quote API issues.
- **Solution:** Upgrade to version `>= 1.3.0`.
- **Link:** [RUSTSEC-2024-0006](#)

8. `tungstenite` (v0.17.3)

- **Issue:** Denial of service vulnerability.
- **Solution:** Upgrade to version `>= 0.20.1`.

◦ **Link:** [RUSTSEC-2023-0065](#)

9. **webpki (v0.21.4)**

- **Issue:** CPU exhaustion in certificate path building.
- **Solution:** Upgrade to version `>= 0.22.2`.
- **Link:** [RUSTSEC-2023-0052](#)

Other Warnings:

1. Unmaintained Crates: Several crates, including `atty`, `buf_redux`, `mach`, and `multipart`, are unmaintained and may need replacement.
2. Yanked Crates: Some crates, such as `futures-util` and `hermit-abi`, are marked as yanked.

Additionally, we have manually identified some outdated dependencies that can be upgraded:

1. The codebase is using the `tiny_keccak` library version 1.5, while the latest version is 2.0.2 ([crates.io](#)).
2. It is also using `sha2` version 0.10.6, while the latest is 0.10.8.
3. The API version used for the attestation mechanism is v4, while the latest is v5.

We recommend addressing these vulnerabilities and updating the outdated dependencies to enhance the security and reliability of the project.

Recommendation: Consider updating the dependencies to avoid these risks and run `cargo audit` again to ensure they are mitigated.

LCP-6 Consider Using `rand::rngs::OsRng` for More Robustness

• **Informational** ⓘ **Fixed**

✓ **Update**

The team fixed the issue as recommended in commit `8ec8a17`.

File(s) affected: `modules/remote-attestation/src/ias_utils.rs`

Description: The function `get_quote()` in `ias_utils.rs` uses `rand::thread_rng()` which provides a pseudorandom value in each thread independently which might not be as robust or secure as desired.

Recommendation: For the usage here, it seems less concerning as it is only for generating a nonce to avoid replay attacks on the quote. If more security is required, consider using `rand::rngs::OsRng` which provides a cryptographically secure random number generator that sources randomness from the operating system's entropy source.

LCP-7 Enclave Leaking Panic Contents

• **Informational** ⓘ **Fixed**

✓ **Update**

The team fixed the issue in commit `e99cbd3`. Now, unless `panic-logging` feature is on, it will only log a static message.

File(s) affected: `enclave-modules/runtime/lib.rs`

Description: The panic handler added to this forked crate makes use of `ocall` to pass panic information outside of the enclave, simply forwarding the info associated with the panic.

Recommendation: Evaluate if this behavior is strictly necessary in production, otherwise consider only passing some static string.

Auditor Suggestions

S1 Misuse of `rsgx_raw_is_outside_enclave` Function

Fixed

✓ **Update**

The team fixed the issue as recommended in the commit `3cacd40`.

File(s) affected: `enclave-modules/utils/src/pointers.rs`

Description: In the `validate_mut_ptr()` function, it checks against `rsgx_raw_is_outside_enclave()` and returns an error if the condition passes. The `rsgx_raw_is_outside_enclave()` function is supposed to be used when `user_check` attributes are specified in the EDL files. For `in` or `out` attributes, the SDK ensures that the memory allocation is within the specified size. Currently, the check is only applied to `output_buf` from the `ecall_execute_command()`, and the `out_buf` uses the `out` attribute in the EDL file. In other words, the `rsgx_raw_is_outside_enclave()` check is redundant.

See the [Intel Guide](#), specifically section 4.1.2, which discusses the `user_check` attribute.

Recommendation: Consider removing the check here and instead check if `ptr.is_null()` or `ptr_len == 0`.

S2 Missing Validations

Mitigated

✓ Update

Some validations are added in the following commits: `aca4366`, `0f3591d`, `a91d2b5`, `1cb681c`, `bf2266a`. We have updated the status in the description of each point and flagged the overall status as mitigated.

The client provided the following explanation for the remaining ones:

Some validations are considered unnecessary as they can be safely assumed: lcp-solidity points 2,3-2,4,7,etc.

For point lcp 2, cannot use `rsgx_verify_report()` in the enclave outside, and our enclave doesn't need to verify the report to trust. (this comment has already posted on slack)

For point solidity 5, an ELC instance does not correspond one-to-one with a specific LCPClient instance. Therefore, it is unnecessary to include the `clientId` in the report.

File(s) affected: `LCPClientBase.sol`, `NodePtr.sol`, `AVRValidator.sol`, `modules/lcp-client/src/client_def.rs`, `modules/remote-attestation/src/ias-utils.rs`

Related Issue(s): [SWC-123](#)

Description: We recommend adding the following validations to tighten the potential attack surface:

lcp-solidity Repository:

1. **Fixed** `NodePtr.getPtr()`: Consider adding validations to ensure that `_ixs`, `_ixf`, and `_ixl` cannot exceed `2^80 - 1`, as they can use at most 80 bits.
2. `AVRValidator.validateAndExtractElements()`: This function only calls `validateAdvisories()` when the status hash `h` is one of the whitelisted values like `HASHED_GROUP_OUT_OF_DATE`, `HASHED_CONFIGURATION_NEEDED`, etc., in the `if` statement. We suggest enforcing a revert in the `else` statement of the condition so that unexpected statuses will not be processed.
3. **Mitigated** In `LCPClientBase.initializeClient()`:
 1. The variables `clientId`, `clientState.allowed_quote_statuses[i]` and `clientState.allowed_advisory_ids[i]` should be validated against the empty string.
 2. Additionally, the function does not check that the provided `clientId` was already initialized and allows the caller to rewrite the client storage.
4. `LCPClientBase.registerEnclaveKey()`: Consider adding validation to ensure that `reElems.mrenclave` cannot be empty bytes.
5. In `LCPClientBase.registerEnclaveKey()`: there is no check against the report and the given `clientId` allowing to link an enclave key with a different, potentially wrong, client id. Consider adding the `clientId` to the report.
6. **Fixed** In `LCPClientBase.updateState()`: there is the possibility of overwriting the `consensusState`. Consider checking that the consensus state at the `postHeight` is empty before setting it.
7. The client id is validated on the LCP node to have a specific format but that is not done on the solidity light client contract. If the client id has the wrong format, a re-registering in the IBC handler would be necessary with the correct `clientId` format.

lcp Repository:

1. **Fixed** `modules/lcp-client/src/client_def.rs::update_client()`: The code comment for this function states that `"// verify_client_message verifies a client message"`. Please update the code comment to match the `update_client()` function.

Recommendation: We recommend adding the relevant checks.

S3 General Suggestions and Best Practices

Mitigated

✓ Update

The team followed some of the suggestions in the commits: `010fbf9d`, `c1958728`. We have updated the status in the description of each point and flagged the overall status as mitigated.

File(s) affected: `modules/remote-attestation/src/ias-utils.rs`, `AVRValidator.sol`, `LCPClientBase.sol`

Description: Here are some general suggestions and best practices we recommend the team follow:

lcp-solidity Repository:

1. **Fixed** `AVRValidator.validateAdvisories()`: The line `allowedAdvisories[string(report[lastStart:lastStart + offset - lastStart - 1])]`... can be simplified to `report[lastStart:offset-1]` instead of the more complex `lastStart + offset - lastStart - 1`.
2. With recent solc versions `require` statements can be used in conjunction with custom errors to improve code readability while retaining gas savings.
3. Consider if the team can benefit from adding index to the `RegisteredEnclaveKey` event in the `LCPClientBase` contract.

Icp Repository:

1. **Fixed** In the function `modules/remote-attestation/src/ias_utils.rs::decode_spid()` : The variable `spid_str` represents a hexadecimal string where each pair of characteres corresponds to a single byte. When `hex::decode` is called, it converts it into a `Vec<u8>` containing 16 bytes. Therefore, it is not necessary to later call `spid.id.copy_from_slice(&decoded_vec[..16])` . Instead, `spid.id.copy_from_slice(&decoded_vec)` would be sufficient. It might also be usefull to check that the length of `decoded_vec` is 16.
2. **Fixed** `modules/remote-attestation/src/ias_utils.rs::get_quote()` : Consider using the `quote_size` variable instead of a hardcoded value of `2048` in the line `let quote = [0u8; 2048];` .

S4 Missing Gap Storage for Upgradeable Base Contract

Fixed

✓ Update

The team fixed the issue as recommended in commit `87eed10a` .

File(s) affected: `LCPClientBase.sol`

Description: The `LCPClientBase` contract can be used as the base contract for an upgradeable contract in the `LCPClientOwnableUpgradeable` contract. However, the `LCPClientBase` implementation does not include gap storage to ensure there are extra storage slots available for additions during an upgrade.

Recommendation: Consider adding a `_gap` storage (see: [example](#)) to ensure smooth upgrades.

S5 Ownership Can Be Renounced

Acknowledged

i Update

The client acknowledged the issue with the following explanation:

This is intended. We consider that each project should decide this behaviour.

File(s) affected: `LCPClientOwnableUpgradeable.sol`

Description: If the owner renounces their ownership, all ownable contracts will be left without an owner. Consequently, any function guarded by the `onlyOwner` modifier will no longer be able to be executed.

Recommendation: Confirm that this is the intended behavior. If not, override and disable the `renounceOwnership()` function in the affected contract.

S6 Critical Role Transfer Not Following Two-Step Pattern

Acknowledged

i Update

The client acknowledged the issue with the following explanation:

We suppose that since LCP is intended for use in various usecase, each project should decide which ownership transfer pattern to use.

File(s) affected: `LCPClientOwnableUpgradeable.sol`

Description: The owner of the contracts can call `transferOwnership()` to transfer the ownership to a new address. If an uncontrollable address is accidentally provided as the new owner address then the contract will no longer have an active owner, and functions with the `onlyOwner` modifier can no longer be executed.

Recommendation: Consider using OpenZeppelin's `Ownable2Step` contract to adopt a two-step ownership pattern in which the new owner must accept their position before the transfer is complete.

S7 Unlocked Pragma

Acknowledged

i Update

The client acknowledged the issue with the following explanation:

We consider that each project should decide specific version.

File(s) affected: `all contracts`

Related Issue(s): [SWC-103](#)

Description: Every Solidity file specifies in the header a version number of the format `pragma solidity (^)0.8.*`. The caret (`^`) before the version number implies an unlocked pragma, meaning that the compiler will use the specified version *and above*, hence the term "unlocked".

Recommendation: For consistency and to prevent unexpected behavior in the future, we recommend removing the caret to lock the file to a specific Solidity version, unless it is an interface or abstract file.

S8 Avoid Unused Code

Fixed

✓ Update

The team fixed the issue as recommended in commit `30a68cb`.

File(s) affected: `ILCPClientErrors.sol`, `LCPCClientBase.sol`

Description: Here is some unused code:

1. The `ILCPClientErrors.LCPClientClientStateEmptyOperators()` error is not used.
2. The `LCPCClientBase.verifyECDSASignature(bytes32 commitment, bytes memory signature, address signer)` function is not used. Note that there are two `verifyECDSASignature()` functions with different signatures.
3. The `LCPCCommitment.parseUpdateStateProxyMessage()` function is only used for testing. Consider moving it to a test or mock contract.

Recommendation: Consider removing unused code.

Definitions

- **High severity** – High-severity issues usually put a large number of users' sensitive information at risk, or are reasonably likely to lead to catastrophic impact for client's reputation or serious financial implications for client and users.
- **Medium severity** – Medium-severity issues tend to put a subset of users' sensitive information at risk, would be detrimental for the client's reputation if exploited, or are reasonably likely to lead to moderate financial impact.
- **Low severity** – 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.
- **Fixed** – Adjusted program implementation, requirements or constraints to eliminate the risk.
- **Mitigated** – Implemented actions to minimize the impact or likelihood of the risk.
- **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).

Toolset

The notes below outline the setup and steps performed in the process of this audit.

Setup

Tool Setup:

- [Cargo Audit](#) [↗](#) 0.21.0
- [Slither](#) [↗](#) v0.10.4

Steps taken to run the tools:

Cargo Audit:

- Installed via `cargo install cargo-audit`
- Ran `cargo audit`

Slither:

1. Install the Slither tool: `pip3 install slither-analyzer`
2. Run Slither from the project directory: `slither`.

3. We temporarily removed the `detectors_to_run` filter in the `slither.config.json` to enable more detectors being run.

Automated Analysis

Cargo Audit

We have summarized the result in the report, here is the raw output:

```
[0m [0m [1m [32m    Fetching [0m advisory database from `https://github.com/RustSec/advisory-db.git`
[0m [0m [1m [32m        Loaded [0m 664 security advisories (from /root/.cargo/advisory-db)
[0m [0m [1m [32m    Updating [0m crates.io index
[0m [0m [1m [32m    Scanning [0m Cargo.lock for vulnerabilities (606 crate dependencies)
[0m [0m [1m [31mCrate:      [0m curve25519-dalek
[0m [0m [1m [31mVersion:    [0m 3.2.0
[0m [0m [1m [31mTitle:      [0m Timing variability in `curve25519-dalek`'s `Scalar29::sub`/`Scalar52::sub`
[0m [0m [1m [31mDate:       [0m 2024-06-18
[0m [0m [1m [31mID:         [0m RUSTSEC-2024-0344
[0m [0m [1m [31mURL:        [0m https://rustsec.org/advisories/RUSTSEC-2024-0344
[0m [0m [1m [31mSolution:   [0m Upgrade to >=4.1.3
[0m [0m [1m [31mDependency tree:
[0mcurve25519-dalek 3.2.0
└─ ed25519-dalek 1.0.1
   └─ tendermint-testgen 0.28.0
      └─ ibc-relayer-types 0.22.0
         └─ integration-test 0.1.0
            └─ ibc-test-framework 0.22.0
               └─ nodes-runner 0.1.0
                  └─ integration-test 0.1.0
                     └─ ibc-telemetry 0.22.0
                        └─ ibc-relayer-cli 1.3.0
                           └─ ibc-test-framework 0.22.0
                              └─ ibc-relayer 0.22.0
                                 └─ nodes-runner 0.1.0
                                    └─ integration-test 0.1.0
                                       └─ ibc-test-framework 0.22.0
                                          └─ ibc-relayer-rest 0.22.0
                                             └─ ibc-relayer-cli 1.3.0
                                                └─ ibc-relayer-cli 1.3.0
           └─ ibc-relayer-rest 0.22.0
              └─ ibc-relayer-cli 1.3.0
                 └─ ibc-relayer 0.22.0
                    └─ ibc-chain-registry 0.22.0
                       └─ ibc-relayer-cli 1.3.0
      └─ tendermint 0.28.0
         └─ tendermint-testgen 0.28.0
            └─ tendermint-rpc 0.28.0
               └─ tendermint-light-client 0.28.0
                  └─ ibc-relayer 0.22.0
                     └─ integration-test 0.1.0
                        └─ ibc-test-framework 0.22.0
                           └─ ibc-relayer-types 0.22.0
                              └─ ibc-relayer-cli 1.3.0
                                 └─ ibc-relayer 0.22.0
                                    └─ ibc-chain-registry 0.22.0
           └─ tendermint-light-client-verifier 0.28.0
              └─ tendermint-light-client 0.28.0
                 └─ integration-test 0.1.0
                    └─ ibc-relayer-types 0.22.0
                       └─ ibc-relayer-cli 1.3.0
                          └─ ibc-relayer 0.22.0
                             └─ tendermint-light-client 0.28.0
                                └─ tendermint-config 0.28.0
                                   └─ tendermint-rpc 0.28.0
                  └─ ibc-telemetry 0.22.0
                     └─ ibc-relayer-types 0.22.0
                        └─ ibc-relayer-cli 1.3.0
                           └─ ibc-relayer 0.22.0
           └─ ibc-relayer 0.22.0
              └─ ed25519-dalek-bip32 0.2.0
                 └─ ibc-relayer 0.22.0
```

```
[0m [0m [1m [31mCrate:      [0m ed25519-dalek
[0m [0m [1m [31mVersion:    [0m 1.0.1
[0m [0m [1m [31mTitle:      [0m Double Public Key Signing Function Oracle Attack on `ed25519-dalek`
[0m [0m [1m [31mDate:       [0m 2022-06-11
[0m [0m [1m [31mID:        [0m RUSTSEC-2022-0093
[0m [0m [1m [31mURL:       [0m https://rustsec.org/advisories/RUSTSEC-2022-0093
[0m [0m [1m [31mSolution:   [0m Upgrade to >=2
[0m [0m [1m [31mDependency tree:
[0med25519-dalek 1.0.1
```

```
└─ tendermint-testgen 0.28.0
   └─ ibc-relayer-types 0.22.0
      └─ integration-test 0.1.0
      └─ ibc-test-framework 0.22.0
         └─ nodes-runner 0.1.0
            └─ integration-test 0.1.0
      └─ ibc-telemetry 0.22.0
         └─ ibc-relayer-cli 1.3.0
            └─ ibc-test-framework 0.22.0
      └─ ibc-relayer 0.22.0
         └─ nodes-runner 0.1.0
            └─ integration-test 0.1.0
            └─ ibc-test-framework 0.22.0
            └─ ibc-relayer-rest 0.22.0
               └─ ibc-relayer-cli 1.3.0
      └─ ibc-relayer-cli 1.3.0
      └─ ibc-relayer-rest 0.22.0
      └─ ibc-relayer-cli 1.3.0
      └─ ibc-relayer 0.22.0
      └─ ibc-chain-registry 0.22.0
         └─ ibc-relayer-cli 1.3.0
└─ tendermint 0.28.0
   └─ tendermint-testgen 0.28.0
   └─ tendermint-rpc 0.28.0
      └─ tendermint-light-client 0.28.0
         └─ ibc-relayer 0.22.0
      └─ integration-test 0.1.0
      └─ ibc-test-framework 0.22.0
      └─ ibc-relayer-types 0.22.0
      └─ ibc-relayer-cli 1.3.0
      └─ ibc-relayer 0.22.0
      └─ ibc-chain-registry 0.22.0
   └─ tendermint-light-client-verifier 0.28.0
      └─ tendermint-light-client 0.28.0
      └─ integration-test 0.1.0
      └─ ibc-relayer-types 0.22.0
      └─ ibc-relayer-cli 1.3.0
      └─ ibc-relayer 0.22.0
   └─ tendermint-light-client 0.28.0
   └─ tendermint-config 0.28.0
      └─ tendermint-rpc 0.28.0
   └─ ibc-telemetry 0.22.0
   └─ ibc-relayer-types 0.22.0
   └─ ibc-relayer-cli 1.3.0
   └─ ibc-relayer 0.22.0
└─ ibc-relayer 0.22.0
└─ ed25519-dalek-bip32 0.2.0
   └─ ibc-relayer 0.22.0
```

```
[0m [0m [1m [31mCrate:      [0m h2
[0m [0m [1m [31mVersion:    [0m 0.3.19
[0m [0m [1m [31mTitle:      [0m Degradation of service in h2 servers with CONTINUATION Flood
[0m [0m [1m [31mDate:       [0m 2024-04-03
[0m [0m [1m [31mID:        [0m RUSTSEC-2024-0332
[0m [0m [1m [31mURL:       [0m https://rustsec.org/advisories/RUSTSEC-2024-0332
[0m [0m [1m [31mSolution:   [0m Upgrade to ^0.3.26 OR >=0.4.4
[0m [0m [1m [31mDependency tree:
[0mh2 0.3.19
```

```
└─ tonic 0.8.1
   └─ tonic-reflection 0.6.0
      └─ service 0.1.0
         └─ lcp 0.0.1
```

```
└─ service 0.1.0
└─ lcp-proto 0.1.0
    └─ tendermint-lc 0.1.0
    └─ service 0.1.0
    └─ lcp-types 0.1.0
        └─ service 0.1.0
        └─ remote-attestation 0.1.0
            └─ lcp 0.0.1
                └─ integration-test 0.1.0
        └─ light-client 0.1.0
            └─ tendermint-lc 0.1.0
            └─ mock-lc 0.1.0
                └─ lcp-client 0.1.0
            └─ lcp-client 0.1.0
                └─ context 0.1.0
                    └─ lcp-client 0.1.0
        └─ lcp 0.0.1
        └─ keymanager 0.1.0
            └─ remote-attestation 0.1.0
            └─ lcp 0.0.1
            └─ integration-test 0.1.0
            └─ enclave-api 0.1.0
                └─ service 0.1.0
                └─ remote-attestation 0.1.0
                └─ lcp 0.0.1
                    └─ integration-test 0.1.0
        └─ integration-test 0.1.0
        └─ enclave-api 0.1.0
        └─ ecall-commands 0.1.0
            └─ lcp 0.0.1
            └─ integration-test 0.1.0
            └─ enclave-api 0.1.0
        └─ context 0.1.0
        └─ commitments 0.1.0
            └─ light-client 0.1.0
            └─ integration-test 0.1.0
            └─ enclave-api 0.1.0
            └─ ecall-commands 0.1.0
        └─ attestation-report 0.1.0
            └─ remote-attestation 0.1.0
            └─ lcp-client 0.1.0
            └─ keymanager 0.1.0
            └─ integration-test 0.1.0
            └─ enclave-api 0.1.0
        └─ integration-test 0.1.0
        └─ enclave-api 0.1.0
└─ ibc-test-framework 0.22.0
    └─ nodes-runner 0.1.0
    └─ integration-test 0.1.0
└─ ibc-relayer 0.22.0
    └─ nodes-runner 0.1.0
    └─ integration-test 0.1.0
    └─ ibc-test-framework 0.22.0
    └─ ibc-relayer-rest 0.22.0
        └─ ibc-relayer-cli 1.3.0
            └─ ibc-test-framework 0.22.0
    └─ ibc-relayer-cli 1.3.0
└─ ibc-proto 0.24.1
    └─ integration-test 0.1.0
    └─ ibc-test-framework 0.22.0
    └─ ibc-relayer-types 0.22.0
        └─ integration-test 0.1.0
        └─ ibc-test-framework 0.22.0
        └─ ibc-telemetry 0.22.0
            └─ ibc-relayer-cli 1.3.0
                └─ ibc-relayer 0.22.0
        └─ ibc-relayer-rest 0.22.0
        └─ ibc-relayer-cli 1.3.0
        └─ ibc-relayer 0.22.0
        └─ ibc-chain-registry 0.22.0
            └─ ibc-relayer-cli 1.3.0
    └─ ibc-relayer 0.22.0
```

```
| └─ ibc-chain-registry 0.22.0
|─ request 0.11.14
| └─ ibc-chain-registry 0.22.0
|─ hyper 0.14.20
|   └─ tonic 0.8.1
|   └─ tendermint-rpc 0.28.0
|     └─ tendermint-light-client 0.28.0
|       └─ ibc-relayer 0.22.0
|     └─ integration-test 0.1.0
|     └─ ibc-test-framework 0.22.0
|     └─ ibc-relayer-types 0.22.0
|     └─ ibc-relayer-cli 1.3.0
|     └─ ibc-relayer 0.22.0
|     └─ ibc-chain-registry 0.22.0
|   └─ request 0.11.14
|   └─ hyper-timeout 0.4.1
|     └─ tonic 0.8.1
|   └─ hyper-rustls 0.23.2
|     └─ request 0.11.14
|   └─ hyper-rustls 0.22.1
|     └─ tendermint-rpc 0.28.0
|     └─ hyper-proxy 0.9.1
|       └─ tendermint-rpc 0.28.0
|   └─ hyper-proxy 0.9.1
|   └─ axum 0.5.13
|     └─ tonic 0.8.1
```

```
[0m [0m [1m [31mCrate:      [0m h2
[0m [0m [1m [31mVersion:    [0m 0.3.19
[0m [0m [1m [31mTitle:      [0m Resource exhaustion vulnerability in h2 may lead to Denial of Service (DoS)
[0m [0m [1m [31mDate:       [0m 2024-01-17
[0m [0m [1m [31mID:        [0m RUSTSEC-2024-0003
[0m [0m [1m [31mURL:       [0m https://rustsec.org/advisories/RUSTSEC-2024-0003
[0m [0m [1m [31mSolution:  [0m Upgrade to ^0.3.24 OR >=0.4.2
```

```
[0m [0m [1m [31mCrate:      [0m libgit2-sys
[0m [0m [1m [31mVersion:    [0m 0.15.2+1.6.4
[0m [0m [1m [31mTitle:      [0m Memory corruption, denial of service, and arbitrary code execution in
libgit2
[0m [0m [1m [31mDate:       [0m 2024-02-06
[0m [0m [1m [31mID:        [0m RUSTSEC-2024-0013
[0m [0m [1m [31mURL:       [0m https://rustsec.org/advisories/RUSTSEC-2024-0013
[0m [0m [1m [31mSeverity:   [0m 8.6 (high)
[0m [0m [1m [31mSolution:  [0m Upgrade to >=0.16.2
[0m [0m [1m [31mDependency tree:
[0mlibgit2-sys 0.15.2+1.6.4
|─ git2 0.17.2
|   └─ lcp 0.0.1
```

```
[0m [0m [1m [31mCrate:      [0m mio
[0m [0m [1m [31mVersion:    [0m 0.8.4
[0m [0m [1m [31mTitle:      [0m Tokens for named pipes may be delivered after deregistration
[0m [0m [1m [31mDate:       [0m 2024-03-04
[0m [0m [1m [31mID:        [0m RUSTSEC-2024-0019
[0m [0m [1m [31mURL:       [0m https://rustsec.org/advisories/RUSTSEC-2024-0019
[0m [0m [1m [31mSolution:  [0m Upgrade to >=0.8.11
[0m [0m [1m [31mDependency tree:
[0mmio 0.8.4
|─ tokio 1.25.1
|   └─ tower 0.4.13
|     └─ tower-http 0.3.4
|       └─ axum 0.5.13
|         └─ tonic 0.8.1
|           └─ tonic-reflection 0.6.0
|             └─ service 0.1.0
|               └─ lcp 0.0.1
|             └─ service 0.1.0
|             └─ lcp-proto 0.1.0
|               └─ tendermint-lc 0.1.0
|                 └─ service 0.1.0
|                 └─ lcp-types 0.1.0
|                   └─ service 0.1.0
```



```
└─ remote-attestation 0.1.0
  └─ lcp 0.0.1
    └─ integration-test 0.1.0
└─ light-client 0.1.0
  └─ tendermint-lc 0.1.0
  └─ mock-lc 0.1.0
    └─ lcp-client 0.1.0
  └─ lcp-client 0.1.0
  └─ context 0.1.0
    └─ lcp-client 0.1.0
└─ lcp 0.0.1
└─ keymanager 0.1.0
  └─ remote-attestation 0.1.0
  └─ lcp 0.0.1
  └─ integration-test 0.1.0
  └─ enclave-api 0.1.0
    └─ service 0.1.0
    └─ remote-attestation 0.1.0
    └─ lcp 0.0.1
    └─ integration-test 0.1.0
└─ integration-test 0.1.0
└─ enclave-api 0.1.0
└─ ecall-commands 0.1.0
  └─ lcp 0.0.1
  └─ integration-test 0.1.0
  └─ enclave-api 0.1.0
└─ context 0.1.0
└─ commitments 0.1.0
  └─ light-client 0.1.0
  └─ integration-test 0.1.0
  └─ enclave-api 0.1.0
  └─ ecall-commands 0.1.0
└─ attestation-report 0.1.0
  └─ remote-attestation 0.1.0
  └─ lcp-client 0.1.0
  └─ keymanager 0.1.0
  └─ integration-test 0.1.0
  └─ enclave-api 0.1.0
└─ integration-test 0.1.0
└─ enclave-api 0.1.0
└─ ibc-test-framework 0.22.0
  └─ nodes-runner 0.1.0
  └─ integration-test 0.1.0
└─ ibc-relayer 0.22.0
  └─ nodes-runner 0.1.0
  └─ integration-test 0.1.0
  └─ ibc-test-framework 0.22.0
  └─ ibc-relayer-rest 0.22.0
    └─ ibc-relayer-cli 1.3.0
      └─ ibc-test-framework 0.22.0
  └─ ibc-relayer-cli 1.3.0
└─ ibc-proto 0.24.1
  └─ integration-test 0.1.0
  └─ ibc-test-framework 0.22.0
  └─ ibc-relayer-types 0.22.0
    └─ integration-test 0.1.0
    └─ ibc-test-framework 0.22.0
    └─ ibc-telemetry 0.22.0
      └─ ibc-relayer-cli 1.3.0
        └─ ibc-relayer 0.22.0
      └─ ibc-relayer-rest 0.22.0
      └─ ibc-relayer-cli 1.3.0
      └─ ibc-relayer 0.22.0
      └─ ibc-chain-registry 0.22.0
        └─ ibc-relayer-cli 1.3.0
  └─ ibc-relayer 0.22.0
  └─ ibc-chain-registry 0.22.0
└─ tonic 0.8.1
└─ axum 0.5.13
└─ tonic-reflection 0.6.0
└─ tonic 0.8.1
└─ tokio-util 0.7.3
```

- └─ tower 0.4.13
- └─ tonic 0.8.1
- └─ h2 0.3.19
 - └─ tonic 0.8.1
 - └─ reqwest 0.11.14
 - └─ ibc-chain-**registry** 0.22.0
 - └─ hyper 0.14.20
 - └─ tonic 0.8.1
 - └─ tendermint-rpc 0.28.0
 - └─ tendermint-light-client 0.28.0
 - └─ ibc-relayer 0.22.0
 - └─ integration-test 0.1.0
 - └─ ibc-test-framework 0.22.0
 - └─ ibc-relayer-types 0.22.0
 - └─ ibc-relayer-cli 1.3.0
 - └─ ibc-relayer 0.22.0
 - └─ ibc-chain-**registry** 0.22.0
 - └─ reqwest 0.11.14
 - └─ hyper-timeout 0.4.1
 - └─ tonic 0.8.1
 - └─ hyper-rustls 0.23.2
 - └─ reqwest 0.11.14
 - └─ hyper-rustls 0.22.1
 - └─ tendermint-rpc 0.28.0
 - └─ hyper-proxy 0.9.1
 - └─ tendermint-rpc 0.28.0
 - └─ hyper-proxy 0.9.1
 - └─ axum 0.5.13
 - └─ tokio-stream 0.1.9
 - └─ tonic-reflection 0.6.0
 - └─ tonic 0.8.1
 - └─ tokio-rustls 0.23.4
 - └─ tonic 0.8.1
 - └─ reqwest 0.11.14
 - └─ hyper-rustls 0.23.2
 - └─ async-tungstenite 0.17.2
 - └─ tendermint-rpc 0.28.0
 - └─ tokio-rustls 0.22.0
 - └─ hyper-rustls 0.22.1
 - └─ hyper-proxy 0.9.1
 - └─ tokio-io-timeout 1.2.0
 - └─ hyper-timeout 0.4.1
 - └─ tendermint-rpc 0.28.0
 - └─ tendermint-light-client 0.28.0
 - └─ service 0.1.0
 - └─ reqwest 0.11.14
 - └─ lcp 0.0.1
 - └─ integration-test 0.1.0
 - └─ ibc-test-framework 0.22.0
 - └─ ibc-relayer-cli 1.3.0
 - └─ ibc-relayer 0.22.0
 - └─ ibc-chain-**registry** 0.22.0
 - └─ hyper-timeout 0.4.1
 - └─ hyper-rustls 0.23.2
 - └─ hyper-rustls 0.22.1
 - └─ hyper-proxy 0.9.1
 - └─ hyper 0.14.20
 - └─ h2 0.3.19
 - └─ axum 0.5.13
 - └─ async-tungstenite 0.17.2

```
[0m [0m [1m [31mCrate:      [0m rsa
[0m [0m [1m [31mVersion:    [0m 0.9.2
[0m [0m [1m [31mTitle:      [0m Marvin Attack: potential key recovery through timing sidechannels
[0m [0m [1m [31mDate:       [0m 2023-11-22
[0m [0m [1m [31mID:        [0m RUSTSEC-2023-0071
[0m [0m [1m [31mURL:       [0m https://rustsec.org/advisories/RUSTSEC-2023-0071
[0m [0m [1m [31mSeverity:   [0m 5.9 (medium)
[0m [0m [1m [31mSolution:   [0m No fixed upgrade is available!
[0m [0m [1m [31mDependency tree:
[0mrsa 0.9.2
└─ remote-attestation 0.1.0
```

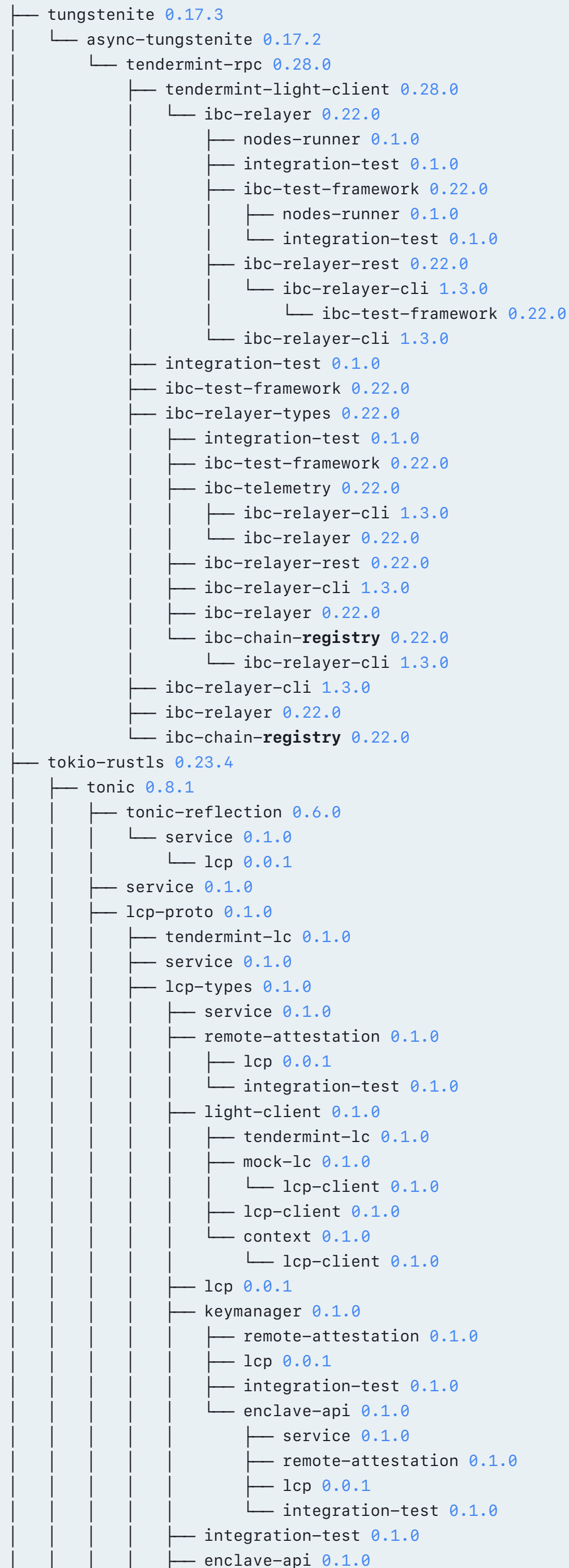
```
└─ lcp 0.0.1
└─ integration-test 0.1.0
```

```
[0m [0m [1m [31mCrate:      [0m rustls
[0m [0m [1m [31mVersion:    [0m 0.19.1
[0m [0m [1m [31mTitle:      [0m `rustls::ConnectionCommon::complete_io` could fall into an infinite loop
based on network input
[0m [0m [1m [31mDate:       [0m 2024-04-19
[0m [0m [1m [31mID:        [0m RUSTSEC-2024-0336
[0m [0m [1m [31mURL:       [0m https://rustsec.org/advisories/RUSTSEC-2024-0336
[0m [0m [1m [31mSeverity:   [0m 7.5 (high)
[0m [0m [1m [31mSolution:  [0m Upgrade to >=0.23.5 OR >=0.22.4, <0.23.0 OR >=0.21.11, <0.22.0
[0m [0m [1m [31mDependency tree:
[0mrustls 0.19.1
```

```
└─ tokio-rustls 0.22.0
└─ hyper-rustls 0.22.1
└─ tendermint-rpc 0.28.0
└─ tendermint-light-client 0.28.0
└─ ibc-relayer 0.22.0
└─ nodes-runner 0.1.0
└─ integration-test 0.1.0
└─ ibc-test-framework 0.22.0
└─ nodes-runner 0.1.0
└─ integration-test 0.1.0
└─ ibc-relayer-rest 0.22.0
└─ ibc-relayer-cli 1.3.0
└─ ibc-test-framework 0.22.0
└─ ibc-relayer-cli 1.3.0
└─ integration-test 0.1.0
└─ ibc-test-framework 0.22.0
└─ ibc-relayer-types 0.22.0
└─ integration-test 0.1.0
└─ ibc-test-framework 0.22.0
└─ ibc-telemetry 0.22.0
└─ ibc-relayer-cli 1.3.0
└─ ibc-relayer 0.22.0
└─ ibc-relayer-rest 0.22.0
└─ ibc-relayer-cli 1.3.0
└─ ibc-relayer 0.22.0
└─ ibc-chain-registry 0.22.0
└─ ibc-relayer-cli 1.3.0
└─ ibc-relayer-cli 1.3.0
└─ ibc-relayer 0.22.0
└─ ibc-chain-registry 0.22.0
└─ hyper-proxy 0.9.1
└─ tendermint-rpc 0.28.0
└─ hyper-proxy 0.9.1
└─ rustls-native-certs 0.5.0
└─ hyper-rustls 0.22.1
└─ hyper-proxy 0.9.1
└─ remote-attestation 0.1.0
└─ lcp 0.0.1
└─ integration-test 0.1.0
└─ hyper-rustls 0.22.1
└─ attestation-report 0.1.0
└─ remote-attestation 0.1.0
└─ lcp-client 0.1.0
└─ keymanager 0.1.0
└─ remote-attestation 0.1.0
└─ lcp 0.0.1
└─ integration-test 0.1.0
└─ enclave-api 0.1.0
└─ service 0.1.0
└─ lcp 0.0.1
└─ remote-attestation 0.1.0
└─ lcp 0.0.1
└─ integration-test 0.1.0
└─ integration-test 0.1.0
└─ enclave-api 0.1.0
```

```
[0m [0m [1m [31mCrate:      [0m rustls
[0m [0m [1m [31mVersion:    [0m 0.20.6
```

```
[0m [0m [1m [31mTitle:      [0m `rustls::ConnectionCommon::complete_io` could fall into an infinite loop
based on network input
[0m [0m [1m [31mDate:      [0m 2024-04-19
[0m [0m [1m [31mID:      [0m RUSTSEC-2024-0336
[0m [0m [1m [31mURL:      [0m https://rustsec.org/advisories/RUSTSEC-2024-0336
[0m [0m [1m [31mSeverity: [0m 7.5 (high)
[0m [0m [1m [31mSolution: [0m Upgrade to >=0.23.5 OR >=0.22.4, <0.23.0 OR >=0.21.11, <0.22.0
[0m [0m [1m [31mDependency tree:
[0mrustls 0.20.6
```



```
└─ ecall-commands 0.1.0
  └─ lcp 0.0.1
  └─ integration-test 0.1.0
  └─ enclave-api 0.1.0
└─ context 0.1.0
└─ commitments 0.1.0
  └─ light-client 0.1.0
  └─ integration-test 0.1.0
  └─ enclave-api 0.1.0
  └─ ecall-commands 0.1.0
└─ attestation-report 0.1.0
  └─ remote-attestation 0.1.0
  └─ lcp-client 0.1.0
  └─ keymanager 0.1.0
  └─ integration-test 0.1.0
  └─ enclave-api 0.1.0
└─ integration-test 0.1.0
└─ enclave-api 0.1.0
└─ ibc-test-framework 0.22.0
└─ ibc-relayer 0.22.0
└─ ibc-proto 0.24.1
  └─ integration-test 0.1.0
  └─ ibc-test-framework 0.22.0
  └─ ibc-relayer-types 0.22.0
  └─ ibc-relayer 0.22.0
  └─ ibc-chain-registry 0.22.0
└─ request 0.11.14
  └─ ibc-chain-registry 0.22.0
└─ hyper-rustls 0.23.2
  └─ request 0.11.14
└─ async-tungstenite 0.17.2
└─ request 0.11.14
└─ hyper-rustls 0.23.2
```

```
[0m [0m [1m [31mCrate:      [0m shlex
[0m [0m [1m [31mVersion:    [0m 1.1.0
[0m [0m [1m [31mTitle:      [0m Multiple issues involving quote API
[0m [0m [1m [31mDate:       [0m 2024-01-21
[0m [0m [1m [31mID:        [0m RUSTSEC-2024-0006
[0m [0m [1m [31mURL:       [0m https://rustsec.org/advisories/RUSTSEC-2024-0006
[0m [0m [1m [31mSolution:   [0m Upgrade to >=1.3.0
[0m [0m [1m [31mDependency tree:
[0mshlex 1.1.0
└─ bindgen 0.65.1
  └─ librocksdb-sys 0.11.0+8.1.1
    └─ rocksdb 0.21.0
      └─ store 0.1.0
        └─ service 0.1.0
          └─ lcp 0.0.1
        └─ remote-attestation 0.1.0
          └─ lcp 0.0.1
          └─ integration-test 0.1.0
        └─ ocall-commands 0.1.0
          └─ ocall-handler 0.1.0
            └─ integration-test 0.1.0
            └─ host 0.1.0
              └─ lcp 0.0.1
              └─ integration-test 0.1.0
              └─ enclave-api 0.1.0
                └─ service 0.1.0
                └─ remote-attestation 0.1.0
                └─ lcp 0.0.1
                └─ integration-test 0.1.0
            └─ host 0.1.0
        └─ light-client 0.1.0
          └─ tendermint-lc 0.1.0
          └─ mock-lc 0.1.0
            └─ lcp-client 0.1.0
          └─ lcp-client 0.1.0
            └─ context 0.1.0
              └─ lcp-client 0.1.0
        └─ lcp-client 0.1.0
```



```
└─ integration-test 0.1.0
└─ host-environment 0.1.0
    └─ ocall-handler 0.1.0
        └─ integration-test 0.1.0
└─ enclave-api 0.1.0
└─ ecall-commands 0.1.0
    └─ lcp 0.0.1
        └─ integration-test 0.1.0
            └─ enclave-api 0.1.0
└─ context 0.1.0
```

```
[0m [0m [1m [31mCrate:      [0m tungstenite
[0m [0m [1m [31mVersion:    [0m 0.17.3
[0m [0m [1m [31mTitle:      [0m Tungstenite allows remote attackers to cause a denial of service
[0m [0m [1m [31mDate:       [0m 2023-09-25
[0m [0m [1m [31mID:        [0m RUSTSEC-2023-0065
[0m [0m [1m [31mURL:       [0m https://rustsec.org/advisories/RUSTSEC-2023-0065
[0m [0m [1m [31mSeverity:   [0m 7.5 (high)
[0m [0m [1m [31mSolution:  [0m Upgrade to >=0.20.1
[0m [0m [1m [31mDependency tree:
```

```
[0mtungstenite 0.17.3
└─ async-tungstenite 0.17.2
    └─ tendermint-rpc 0.28.0
        └─ tendermint-light-client 0.28.0
            └─ ibc-relayer 0.22.0
                └─ nodes-runner 0.1.0
                    └─ integration-test 0.1.0
                └─ ibc-test-framework 0.22.0
                    └─ nodes-runner 0.1.0
                        └─ integration-test 0.1.0
                └─ ibc-relayer-rest 0.22.0
                    └─ ibc-relayer-cli 1.3.0
                        └─ ibc-test-framework 0.22.0
                └─ ibc-relayer-cli 1.3.0
└─ integration-test 0.1.0
└─ ibc-test-framework 0.22.0
└─ ibc-relayer-types 0.22.0
    └─ integration-test 0.1.0
    └─ ibc-test-framework 0.22.0
    └─ ibc-telemetry 0.22.0
        └─ ibc-relayer-cli 1.3.0
            └─ ibc-relayer 0.22.0
└─ ibc-relayer-rest 0.22.0
└─ ibc-relayer-cli 1.3.0
└─ ibc-relayer 0.22.0
└─ ibc-chain-registry 0.22.0
    └─ ibc-relayer-cli 1.3.0
└─ ibc-relayer-cli 1.3.0
└─ ibc-relayer 0.22.0
└─ ibc-chain-registry 0.22.0
```

```
[0m [0m [1m [31mCrate:      [0m webpki
[0m [0m [1m [31mVersion:    [0m 0.21.4
[0m [0m [1m [31mTitle:      [0m webpki: CPU denial of service in certificate path building
[0m [0m [1m [31mDate:       [0m 2023-08-22
[0m [0m [1m [31mID:        [0m RUSTSEC-2023-0052
[0m [0m [1m [31mURL:       [0m https://rustsec.org/advisories/RUSTSEC-2023-0052
[0m [0m [1m [31mSeverity:   [0m 7.5 (high)
[0m [0m [1m [31mSolution:  [0m Upgrade to >=0.22.2
[0m [0m [1m [31mDependency tree:
```

```
[0mwebpki 0.21.4
└─ webpki-roots 0.21.1
    └─ hyper-rustls 0.22.1
        └─ tendermint-rpc 0.28.0
            └─ tendermint-light-client 0.28.0
                └─ ibc-relayer 0.22.0
                    └─ nodes-runner 0.1.0
                        └─ integration-test 0.1.0
                    └─ ibc-test-framework 0.22.0
                        └─ nodes-runner 0.1.0
                            └─ integration-test 0.1.0
                    └─ ibc-relayer-rest 0.22.0
```

```
└── ibc-relayer-cli 1.3.0
    └── ibc-test-framework 0.22.0
        └── ibc-relayer-cli 1.3.0
└── integration-test 0.1.0
└── ibc-test-framework 0.22.0
└── ibc-relayer-types 0.22.0
    ├── integration-test 0.1.0
    ├── ibc-test-framework 0.22.0
    ├── ibc-telemetry 0.22.0
    │   ├── ibc-relayer-cli 1.3.0
    │   └── ibc-relayer 0.22.0
    ├── ibc-relayer-rest 0.22.0
    ├── ibc-relayer-cli 1.3.0
    ├── ibc-relayer 0.22.0
    └── ibc-chain-registry 0.22.0
        └── ibc-relayer-cli 1.3.0
└── ibc-relayer-cli 1.3.0
└── ibc-relayer 0.22.0
└── ibc-chain-registry 0.22.0
└── hyper-proxy 0.9.1
    └── tendermint-rpc 0.28.0
```

```
webpki-roots 0.17.0
└── remote-attestation 0.1.0
    ├── lcp 0.0.1
    └── integration-test 0.1.0
```

```
tokio-rustls 0.22.0
└── hyper-rustls 0.22.1
└── hyper-proxy 0.9.1
```

```
rustls 0.19.1
└── tokio-rustls 0.22.0
└── rustls-native-certs 0.5.0
    ├── hyper-rustls 0.22.1
    └── hyper-proxy 0.9.1
└── remote-attestation 0.1.0
└── hyper-rustls 0.22.1
└── attestation-report 0.1.0
    ├── remote-attestation 0.1.0
    ├── lcp-client 0.1.0
    ├── keymanager 0.1.0
    │   ├── remote-attestation 0.1.0
    │   ├── lcp 0.0.1
    │   ├── integration-test 0.1.0
    │   └── enclave-api 0.1.0
    │       ├── service 0.1.0
    │       │   └── lcp 0.0.1
    │       ├── remote-attestation 0.1.0
    │       ├── lcp 0.0.1
    │       └── integration-test 0.1.0
    ├── integration-test 0.1.0
    └── enclave-api 0.1.0
```

```
remote-attestation 0.1.0
```

```
hyper-rustls 0.22.1
```

```
hyper-proxy 0.9.1
```

```
attestation-report 0.1.0
```

```
[0m [0m [1m [33mCrate:      [0m atty
[0m [0m [1m [33mVersion:    [0m 0.2.14
[0m [0m [1m [33mWarning:     [0m unmaintained
[0m [0m [1m [33mTitle:       [0m `atty` is unmaintained
[0m [0m [1m [33mDate:        [0m 2024-09-25
[0m [0m [1m [33mID:         [0m RUSTSEC-2024-0375
[0m [0m [1m [33mURL:         [0m https://rustsec.org/advisories/RUSTSEC-2024-0375
[0m [0m [1m [33mDependency tree:
[0matty 0.2.14
```

```
└── ibc-relayer-cli 1.3.0
    └── ibc-test-framework 0.22.0
        ├── nodes-runner 0.1.0
        └── integration-test 0.1.0
```

```
env_logger 0.9.0
└── store 0.1.0
    ├── service 0.1.0
    └── lcp 0.0.1
```

```

└─ remote-attestation 0.1.0
  └─ lcp 0.0.1
    └─ integration-test 0.1.0
└─ ocall-commands 0.1.0
  └─ ocall-handler 0.1.0
    └─ integration-test 0.1.0
      └─ host 0.1.0
        └─ lcp 0.0.1
          └─ integration-test 0.1.0
            └─ enclave-api 0.1.0
              └─ service 0.1.0
                └─ remote-attestation 0.1.0
                  └─ lcp 0.0.1
                    └─ integration-test 0.1.0
└─ host 0.1.0
└─ light-client 0.1.0
  └─ tendermint-lc 0.1.0
  └─ mock-lc 0.1.0
    └─ lcp-client 0.1.0
  └─ lcp-client 0.1.0
  └─ context 0.1.0
    └─ lcp-client 0.1.0
└─ lcp-client 0.1.0
└─ integration-test 0.1.0
└─ host-environment 0.1.0
  └─ ocall-handler 0.1.0
    └─ integration-test 0.1.0
└─ enclave-api 0.1.0
└─ ecall-commands 0.1.0
  └─ lcp 0.0.1
    └─ integration-test 0.1.0
      └─ enclave-api 0.1.0
└─ context 0.1.0
└─ lcp 0.0.1
└─ integration-test 0.1.0
└─ clap 3.2.12
  └─ lcp 0.0.1
  └─ ibc-relayer-cli 1.3.0
  └─ clap_complete 3.2.5
    └─ ibc-relayer-cli 1.3.0
└─ abscissa_core 0.6.0
  └─ ibc-relayer-cli 1.3.0

```

```

[0m [0m [1m [33mCrate:      [0m buf_redux
[0m [0m [1m [33mVersion:    [0m 0.8.4
[0m [0m [1m [33mWarning:    [0m unmaintained
[0m [0m [1m [33mTitle:      [0m buf_redux is Unmaintained
[0m [0m [1m [33mDate:       [0m 2023-01-24
[0m [0m [1m [33mID:        [0m RUSTSEC-2023-0028
[0m [0m [1m [33mURL:       [0m https://rustsec.org/advisories/RUSTSEC-2023-0028
[0m [0m [1m [33mDependency tree:
[0mbuf_redux 0.8.4
└─ multipart 0.18.0
  └─ rouille 3.6.1
    └─ ibc-telemetry 0.22.0
      └─ ibc-relayer-cli 1.3.0
        └─ ibc-test-framework 0.22.0
          └─ nodes-runner 0.1.0
            └─ integration-test 0.1.0
        └─ ibc-relayer 0.22.0
          └─ nodes-runner 0.1.0
            └─ integration-test 0.1.0
          └─ ibc-test-framework 0.22.0
            └─ ibc-relayer-rest 0.22.0
              └─ ibc-relayer-cli 1.3.0
            └─ ibc-relayer-cli 1.3.0
          └─ ibc-relayer-rest 0.22.0

```

```

[0m [0m [1m [33mCrate:      [0m mach
[0m [0m [1m [33mVersion:    [0m 0.3.2
[0m [0m [1m [33mWarning:    [0m unmaintained
[0m [0m [1m [33mTitle:      [0m mach is unmaintained

```

```
[0m [0m [1m [33mDate:      [0m 2020-07-14
[0m [0m [1m [33mID:        [0m RUSTSEC-2020-0168
[0m [0m [1m [33mURL:         [0m https://rustsec.org/advisories/RUSTSEC-2020-0168
[0m [0m [1m [33mDependency tree:
[0mmmach 0.3.2
└─ quanta 0.10.0
   └─ moka 0.9.7
      └─ ibc-telemetry 0.22.0
         └─ ibc-relayer-cli 1.3.0
            └─ ibc-test-framework 0.22.0
               └─ nodes-runner 0.1.0
                  └─ integration-test 0.1.0
            └─ ibc-relayer 0.22.0
               └─ nodes-runner 0.1.0
                  └─ integration-test 0.1.0
            └─ ibc-test-framework 0.22.0
               └─ ibc-relayer-rest 0.22.0
                  └─ ibc-relayer-cli 1.3.0
            └─ ibc-relayer-cli 1.3.0
         └─ ibc-relayer 0.22.0
```

```
[0m [0m [1m [33mCrate:      [0m multipart
[0m [0m [1m [33mVersion:    [0m 0.18.0
[0m [0m [1m [33mWarning:    [0m unmaintained
[0m [0m [1m [33mTitle:      [0m multipart is Unmaintained
[0m [0m [1m [33mDate:      [0m 2023-04-11
[0m [0m [1m [33mID:        [0m RUSTSEC-2023-0050
[0m [0m [1m [33mURL:         [0m https://rustsec.org/advisories/RUSTSEC-2023-0050
[0m [0m [1m [33mDependency tree:
[0mmultipart 0.18.0
└─ rouille 3.6.1
   └─ ibc-telemetry 0.22.0
      └─ ibc-relayer-cli 1.3.0
         └─ ibc-test-framework 0.22.0
            └─ nodes-runner 0.1.0
               └─ integration-test 0.1.0
         └─ ibc-relayer 0.22.0
            └─ nodes-runner 0.1.0
               └─ integration-test 0.1.0
            └─ ibc-test-framework 0.22.0
               └─ ibc-relayer-rest 0.22.0
                  └─ ibc-relayer-cli 1.3.0
            └─ ibc-relayer-cli 1.3.0
      └─ ibc-relayer-rest 0.22.0
```

```
[0m [0m [1m [33mCrate:      [0m proc-macro-error
[0m [0m [1m [33mVersion:    [0m 1.0.4
[0m [0m [1m [33mWarning:    [0m unmaintained
[0m [0m [1m [33mTitle:      [0m proc-macro-error is unmaintained
[0m [0m [1m [33mDate:      [0m 2024-09-01
[0m [0m [1m [33mID:        [0m RUSTSEC-2024-0370
[0m [0m [1m [33mURL:         [0m https://rustsec.org/advisories/RUSTSEC-2024-0370
[0m [0m [1m [33mDependency tree:
[0mproc-macro-error 1.0.4
└─ ouroboros_macro 0.17.0
   └─ ouroboros 0.17.0
      └─ store 0.1.0
         └─ service 0.1.0
            └─ lcp 0.0.1
         └─ remote-attestation 0.1.0
            └─ lcp 0.0.1
               └─ integration-test 0.1.0
         └─ ocall-commands 0.1.0
            └─ ocall-handler 0.1.0
               └─ integration-test 0.1.0
            └─ host 0.1.0
               └─ lcp 0.0.1
                  └─ integration-test 0.1.0
                  └─ enclave-api 0.1.0
                     └─ service 0.1.0
                        └─ remote-attestation 0.1.0
                        └─ lcp 0.0.1
```

```

└── integration-test 0.1.0
    └── host 0.1.0
light-client 0.1.0
├── tendermint-lc 0.1.0
├── mock-lc 0.1.0
│   └── lcp-client 0.1.0
├── lcp-client 0.1.0
├── context 0.1.0
│   └── lcp-client 0.1.0
lcp-client 0.1.0
integration-test 0.1.0
host-environment 0.1.0
├── ocall-handler 0.1.0
│   └── integration-test 0.1.0
enclave-api 0.1.0
ecall-commands 0.1.0
├── lcp 0.0.1
├── integration-test 0.1.0
├── enclave-api 0.1.0
└── context 0.1.0
clap_derive 3.2.7
└── clap 3.2.12
    ├── lcp 0.0.1
    ├── ibc-relayer-cli 1.3.0
    │   ├── ibc-test-framework 0.22.0
    │   │   ├── nodes-runner 0.1.0
    │   │   └── integration-test 0.1.0
    ├── clap_complete 3.2.5
    │   └── ibc-relayer-cli 1.3.0
    └── abscissa_core 0.6.0
        └── ibc-relayer-cli 1.3.0
alloy-sol-macro 0.6.2
└── alloy-sol-types 0.6.2
    ├── lcp-client 0.1.0
    ├── commitments 0.1.0
    │   ├── light-client 0.1.0
    │   ├── integration-test 0.1.0
    │   ├── enclave-api 0.1.0
    │   └── ecall-commands 0.1.0

```

```

[0m [0m [1m [33mCrate:      [0m safemem
[0m [0m [1m [33mVersion:    [0m 0.3.3
[0m [0m [1m [33mWarning:    [0m unmaintained
[0m [0m [1m [33mTitle:      [0m safemem is unmaintained
[0m [0m [1m [33mDate:      [0m 2023-02-14
[0m [0m [1m [33mID:       [0m RUSTSEC-2023-0081
[0m [0m [1m [33mURL:       [0m https://rustsec.org/advisories/RUSTSEC-2023-0081
[0m [0m [1m [33mDependency tree:
[0msafemem 0.3.3

```

```

└── multipart 0.18.0
    └── rouille 3.6.1
        ├── ibc-telemetry 0.22.0
        │   ├── ibc-relayer-cli 1.3.0
        │   │   ├── ibc-test-framework 0.22.0
        │   │   │   ├── nodes-runner 0.1.0
        │   │   │   └── integration-test 0.1.0
        │   └── ibc-relayer 0.22.0
        │       ├── nodes-runner 0.1.0
        │       ├── integration-test 0.1.0
        │       ├── ibc-test-framework 0.22.0
        │       ├── ibc-relayer-rest 0.22.0
        │       │   └── ibc-relayer-cli 1.3.0
        │       └── ibc-relayer-cli 1.3.0
        └── ibc-relayer-rest 0.22.0
buf_redux 0.8.4
└── multipart 0.18.0

```

```

[0m [0m [1m [33mCrate:      [0m serde_cbor
[0m [0m [1m [33mVersion:    [0m 0.11.2
[0m [0m [1m [33mWarning:    [0m unmaintained
[0m [0m [1m [33mTitle:      [0m serde_cbor is unmaintained
[0m [0m [1m [33mDate:      [0m 2021-08-15

```



```
[0m [0m [1m [33mID: [0m RUSTSEC-2021-0127
[0m [0m [1m [33mURL: [0m https://rustsec.org/advisories/RUSTSEC-2021-0127
[0m [0m [1m [33mDependency tree:
[0mserde_cbor 0.11.2
└─ tendermint-light-client 0.28.0
    └─ ibc-relayer 0.22.0
        ├── nodes-runner 0.1.0
        ├── integration-test 0.1.0
        ├── ibc-test-framework 0.22.0
        │   ├── nodes-runner 0.1.0
        │   └─ integration-test 0.1.0
        ├── ibc-relayer-rest 0.22.0
        │   └─ ibc-relayer-cli 1.3.0
        │       └─ ibc-test-framework 0.22.0
        └─ ibc-relayer-cli 1.3.0
```

```
[0m [0m [1m [33mCrate: [0m twoway
[0m [0m [1m [33mVersion: [0m 0.1.8
[0m [0m [1m [33mWarning: [0m unmaintained
[0m [0m [1m [33mTitle: [0m Crate `twoway` deprecated by the author
[0m [0m [1m [33mDate: [0m 2021-05-20
[0m [0m [1m [33mID: [0m RUSTSEC-2021-0146
[0m [0m [1m [33mURL: [0m https://rustsec.org/advisories/RUSTSEC-2021-0146
[0m [0m [1m [33mDependency tree:
[0mtwoway 0.1.8
└─ multipart 0.18.0
    └─ rouille 3.6.1
        ├── ibc-telemetry 0.22.0
        │   ├── ibc-relayer-cli 1.3.0
        │   │   └─ ibc-test-framework 0.22.0
        │   │       ├── nodes-runner 0.1.0
        │   │       └─ integration-test 0.1.0
        │   └─ ibc-relayer 0.22.0
        │       ├── nodes-runner 0.1.0
        │       ├── integration-test 0.1.0
        │       ├── ibc-test-framework 0.22.0
        │       ├── ibc-relayer-rest 0.22.0
        │       │   └─ ibc-relayer-cli 1.3.0
        │       └─ ibc-relayer-cli 1.3.0
        └─ ibc-relayer-rest 0.22.0
```

```
[0m [0m [1m [33mCrate: [0m atty
[0m [0m [1m [33mVersion: [0m 0.2.14
[0m [0m [1m [33mWarning: [0m unsound
[0m [0m [1m [33mTitle: [0m Potential unaligned read
[0m [0m [1m [33mDate: [0m 2021-07-04
[0m [0m [1m [33mID: [0m RUSTSEC-2021-0145
[0m [0m [1m [33mURL: [0m https://rustsec.org/advisories/RUSTSEC-2021-0145
```

```
[0m [0m [1m [33mCrate: [0m borsh
[0m [0m [1m [33mVersion: [0m 0.10.2
[0m [0m [1m [33mWarning: [0m unsound
[0m [0m [1m [33mTitle: [0m Parsing borsh messages with ZST which are not-copy/clone is unsound
[0m [0m [1m [33mDate: [0m 2023-04-12
[0m [0m [1m [33mID: [0m RUSTSEC-2023-0033
[0m [0m [1m [33mURL: [0m https://rustsec.org/advisories/RUSTSEC-2023-0033
[0m [0m [1m [33mDependency tree:
[0mborsh 0.10.2
└─ ibc-proto 0.26.0
    ├── lcp-proto 0.1.0
    │   ├── tendermint-lc 0.1.0
    │   ├── service 0.1.0
    │   │   └─ lcp 0.0.1
    │   └─ lcp-types 0.1.0
    │       ├── service 0.1.0
    │       ├── remote-attestation 0.1.0
    │       │   ├── lcp 0.0.1
    │       │   └─ integration-test 0.1.0
    │       └─ light-client 0.1.0
    │           ├── tendermint-lc 0.1.0
    │           ├── mock-lc 0.1.0
    │           └─ lcp-client 0.1.0
```

```

├── lcp-client 0.1.0
├── context 0.1.0
│   └── lcp-client 0.1.0
├── lcp 0.0.1
├── keymanager 0.1.0
│   ├── remote-attestation 0.1.0
│   ├── lcp 0.0.1
│   ├── integration-test 0.1.0
│   └── enclave-api 0.1.0
│       ├── service 0.1.0
│       ├── remote-attestation 0.1.0
│       ├── lcp 0.0.1
│       └── integration-test 0.1.0
├── integration-test 0.1.0
├── enclave-api 0.1.0
├── ecall-commands 0.1.0
│   ├── lcp 0.0.1
│   ├── integration-test 0.1.0
│   └── enclave-api 0.1.0
├── context 0.1.0
├── commitments 0.1.0
│   ├── light-client 0.1.0
│   ├── integration-test 0.1.0
│   ├── enclave-api 0.1.0
│   └── ecall-commands 0.1.0
├── attestation-report 0.1.0
│   ├── remote-attestation 0.1.0
│   ├── lcp-client 0.1.0
│   ├── keymanager 0.1.0
│   ├── integration-test 0.1.0
│   └── enclave-api 0.1.0
├── integration-test 0.1.0
├── enclave-api 0.1.0
└── ibc 0.29.0
    ├── tendermint-lc 0.1.0
    ├── mock-lc 0.1.0
    ├── light-client 0.1.0
    ├── lcp-types 0.1.0
    ├── lcp-client 0.1.0
    └── integration-test 0.1.0

```

```

[0m [0m [1m [33mCrate:      [0m crossbeam-utils
[0m [0m [1m [33mVersion:    [0m 0.7.2
[0m [0m [1m [33mWarning:    [0m unsound
[0m [0m [1m [33mTitle:      [0m Unsoundness of AtomicCell<*64> arithmetics on 32-bit targets that support
Atomic*64

```

```

[0m [0m [1m [33mDate:      [0m 2022-02-05
[0m [0m [1m [33mID:        [0m RUSTSEC-2022-0041
[0m [0m [1m [33mURL:       [0m https://rustsec.org/advisories/RUSTSEC-2022-0041
[0m [0m [1m [33mDependency tree:
[0m crossbeam-utils 0.7.2
└── crossbeam-channel 0.4.4
    ├── tendermint-light-client 0.28.0
    │   └── ibc-relayer 0.22.0
    │       ├── nodes-runner 0.1.0
    │       ├── integration-test 0.1.0
    │       ├── ibc-test-framework 0.22.0
    │       │   ├── nodes-runner 0.1.0
    │       │   └── integration-test 0.1.0
    │       ├── ibc-relayer-rest 0.22.0
    │       │   └── ibc-relayer-cli 1.3.0
    │       │       └── ibc-test-framework 0.22.0
    │       └── ibc-relayer-cli 1.3.0

```

```

[0m [0m [1m [33mCrate:      [0m unsafe-libyaml
[0m [0m [1m [33mVersion:    [0m 0.2.5
[0m [0m [1m [33mWarning:    [0m unsound
[0m [0m [1m [33mTitle:      [0m Unaligned write of u64 on 32-bit and 16-bit platforms
[0m [0m [1m [33mDate:      [0m 2023-12-20
[0m [0m [1m [33mID:        [0m RUSTSEC-2023-0075
[0m [0m [1m [33mURL:       [0m https://rustsec.org/advisories/RUSTSEC-2023-0075
[0m [0m [1m [33mDependency tree:

```

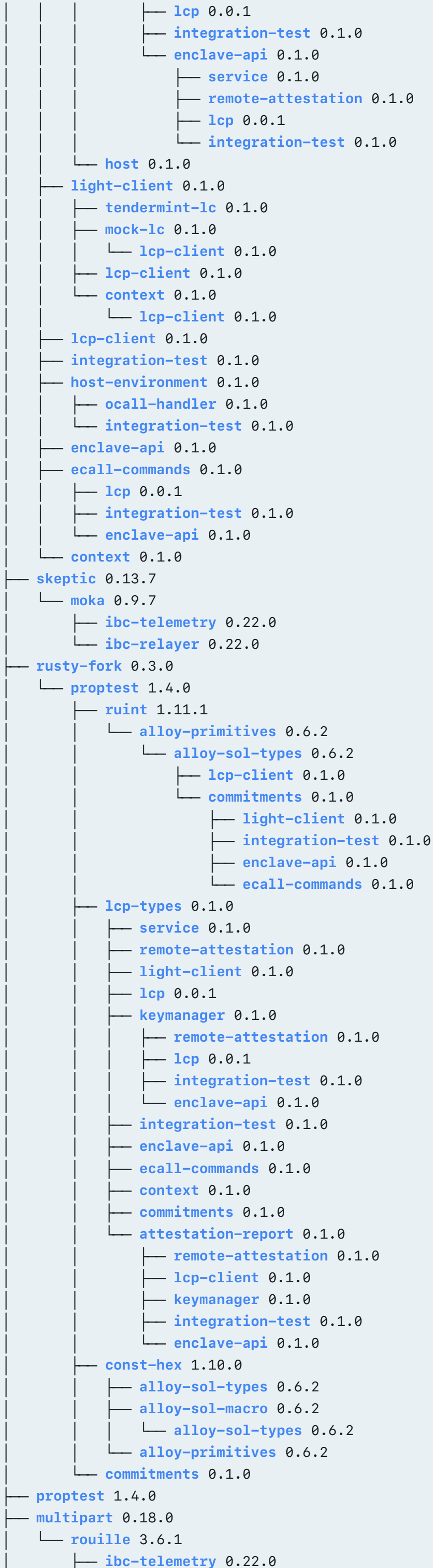
```
[0munsafe-libyaml 0.2.5
└─ serde_yaml 0.9.17
    └─ ibc-test-framework 0.22.0
        ├── nodes-runner 0.1.0
        └─ integration-test 0.1.0

[0m [0m [1m [33mCrate:      [0m futures-util
[0m [0m [1m [33mVersion:    [0m 0.3.26
[0m [0m [1m [33mWarning:    [0m yanked
[0m [0m [1m [33mDependency tree:
[0mfutures-util 0.3.26
└─ tower-http 0.3.4
    └─ axum 0.5.13
        └─ tonic 0.8.1
            ├── tonic-reflection 0.6.0
            │   └─ service 0.1.0
            │       └─ lcp 0.0.1
            ├── service 0.1.0
            ├── lcp-proto 0.1.0
            │   ├── tendermint-lc 0.1.0
            │   ├── service 0.1.0
            │   ├── lcp-types 0.1.0
            │   │   ├── service 0.1.0
            │   │   ├── remote-attestation 0.1.0
            │   │   │   ├── lcp 0.0.1
            │   │   │   └─ integration-test 0.1.0
            │   ├── light-client 0.1.0
            │   │   ├── tendermint-lc 0.1.0
            │   │   ├── mock-lc 0.1.0
            │   │   │   └─ lcp-client 0.1.0
            │   │   ├── lcp-client 0.1.0
            │   │   └─ context 0.1.0
            │   │       └─ lcp-client 0.1.0
            │   ├── lcp 0.0.1
            │   ├── keymanager 0.1.0
            │   │   ├── remote-attestation 0.1.0
            │   │   ├── lcp 0.0.1
            │   │   ├── integration-test 0.1.0
            │   │   └─ enclave-api 0.1.0
            │   │       ├── service 0.1.0
            │   │       ├── remote-attestation 0.1.0
            │   │       ├── lcp 0.0.1
            │   │       └─ integration-test 0.1.0
            │   ├── integration-test 0.1.0
            │   ├── enclave-api 0.1.0
            │   ├── ecall-commands 0.1.0
            │   │   ├── lcp 0.0.1
            │   │   ├── integration-test 0.1.0
            │   │   └─ enclave-api 0.1.0
            │   ├── context 0.1.0
            │   ├── commitments 0.1.0
            │   │   ├── light-client 0.1.0
            │   │   ├── integration-test 0.1.0
            │   │   ├── enclave-api 0.1.0
            │   │   └─ ecall-commands 0.1.0
            │   └─ attestation-report 0.1.0
            │       ├── remote-attestation 0.1.0
            │       ├── lcp-client 0.1.0
            │       ├── keymanager 0.1.0
            │       ├── integration-test 0.1.0
            │       └─ enclave-api 0.1.0
            ├── integration-test 0.1.0
            ├── enclave-api 0.1.0
            ├── ibc-test-framework 0.22.0
            │   ├── nodes-runner 0.1.0
            │   └─ integration-test 0.1.0
            ├── ibc-relayer 0.22.0
            │   ├── nodes-runner 0.1.0
            │   ├── integration-test 0.1.0
            │   ├── ibc-test-framework 0.22.0
            │   ├── ibc-relayer-rest 0.22.0
            │   └─ ibc-relayer-cli 1.3.0
```



```
└─ lcp-types 0.1.0
└─ lcp-client 0.1.0
└─ integration-test 0.1.0
└─ lcp-types 0.1.0
└─ ibc 0.29.0
└─ tendermint 0.28.0
└─ tendermint-testgen 0.28.0
└─ ibc-relayer-types 0.22.0
└─ tendermint-rpc 0.28.0
└─ tendermint-light-client-verifier 0.28.0
└─ tendermint-light-client 0.28.0
└─ integration-test 0.1.0
└─ ibc-relayer-types 0.22.0
└─ ibc-relayer-cli 1.3.0
└─ ibc-relayer 0.22.0
└─ tendermint-light-client 0.28.0
└─ tendermint-config 0.28.0
└─ tendermint-rpc 0.28.0
└─ ibc-telemetry 0.22.0
└─ ibc-relayer-types 0.22.0
└─ ibc-relayer-cli 1.3.0
└─ ibc-relayer 0.22.0
└─ ibc-relayer-cli 1.3.0
└─ ibc-relayer 0.22.0
└─ ibc-chain-registry 0.22.0
└─ hyper-proxy 0.9.1
└─ futures 0.3.26
└─ axum-core 0.2.9
└─ axum 0.5.13
└─ axum 0.5.13
└─ async-tungstenite 0.17.2
└─ tendermint-rpc 0.28.0
```

```
[0m [0m [1m [33mCrate:      [0m hermit-abi
[0m [0m [1m [33mVersion:    [0m 0.3.1
[0m [0m [1m [33mWarning:     [0m yanked
[0m [0m [1m [33mDependency tree:
[0mhermit-abi 0.3.1
└─ io-lifetimes 1.0.11
└─ rustix 0.37.20
└─ tempfile 3.6.0
└─ tendermint-testgen 0.28.0
└─ ibc-relayer-types 0.22.0
└─ integration-test 0.1.0
└─ ibc-test-framework 0.22.0
└─ nodes-runner 0.1.0
└─ integration-test 0.1.0
└─ ibc-telemetry 0.22.0
└─ ibc-relayer-cli 1.3.0
└─ ibc-test-framework 0.22.0
└─ ibc-relayer 0.22.0
└─ nodes-runner 0.1.0
└─ integration-test 0.1.0
└─ ibc-test-framework 0.22.0
└─ ibc-relayer-rest 0.22.0
└─ ibc-relayer-cli 1.3.0
└─ ibc-relayer-cli 1.3.0
└─ ibc-relayer-rest 0.22.0
└─ ibc-relayer-cli 1.3.0
└─ ibc-relayer 0.22.0
└─ ibc-chain-registry 0.22.0
└─ ibc-relayer-cli 1.3.0
└─ store 0.1.0
└─ service 0.1.0
└─ lcp 0.0.1
└─ remote-attestation 0.1.0
└─ lcp 0.0.1
└─ integration-test 0.1.0
└─ ocall-commands 0.1.0
└─ ocall-handler 0.1.0
└─ integration-test 0.1.0
└─ host 0.1.0
```

```
└─ ibc-relayer-rest 0.22.0
└─ integration-test 0.1.0
└─ dialoguer 0.10.3
   └─ ibc-relayer-cli 1.3.0
```

```
[0m [0m [1m [31merror: [0m 12 vulnerabilities found!
[0m [0m [1m [33mwarning: [0m 14 allowed warnings found
```

Slither

Slither analyzed 36 contracts with 115 detectors and found 156 results. Most of these are false positives. We have included the valid ones in the report.

Test Suite Results

To run the test in lcp repository:

1. curl -LO https://download.01.org/intel-sgx/sgx-linux/2.19/distro/ubuntu22.04-server/sgx_linux_x64_sdk_2.19.100.3.bin
2. chmod +x ./sgx_linux_x64_sdk_2.19.100.3.bin
3. echo -e 'no\n/opt' | ./sgx_linux_x64_sdk_2.19.100.3.bin
4. cargo install cargo-llvm-cov
5. source /opt/sgxsdk/environment && make test

To run test in the lcp-solidity repository:

1. npm install
2. forge test

```
/bin/sh: line 1: file: command not found
info: cargo-llvm-cov currently setting cfg(coverage) and cfg(coverage_nightly); you can opt-out it by
passing --no-cfg-coverage and --no-cfg-coverage-nightly
  Compiling rustversion v1.0.18
  Compiling lz4-sys v1.11.1+lz4-1.10.0
  Compiling bzip2-sys v0.1.11+1.0.8
  Compiling libsecp256k1 v0.7.1 (https://github.com/paritytech/libsecp256k1?
rev=48dabd8821852c5fe00b846f6c37e1f6b05c3d8c#48dabd88)
  Compiling axum-core v0.3.4
  Compiling parity-scale-codec v3.7.0
  Compiling axum v0.6.20
  Compiling scale-info v2.11.6
  Compiling crypto v0.1.0 (/root/workspace/projects/LCP-weaudit/qs_workspace-LCP-weaudit-main-
github/lcp/modules/crypto)
  Compiling librocksdb-sys v0.11.0+8.1.1
  Compiling tonic v0.9.2
  Compiling ibc-proto v0.26.0
  Compiling ring v0.17.8
  Compiling lcp-proto v0.1.0 (/root/workspace/projects/LCP-weaudit/qs_workspace-LCP-weaudit-main-
github/lcp/proto)
  Compiling ibc v0.29.0
  Compiling host v0.1.0 (/root/workspace/projects/LCP-weaudit/qs_workspace-LCP-weaudit-main-
github/lcp/modules/host)
  Compiling enclave-api v0.1.0 (/root/workspace/projects/LCP-weaudit/qs_workspace-LCP-weaudit-main-
github/lcp/modules/enclave-api)
  Compiling remote-attestation v0.1.0 (/root/workspace/projects/LCP-weaudit/qs_workspace-LCP-weaudit-
main-github/lcp/modules/remote-attestation)
  Compiling tonic-reflection v0.9.2
  Compiling lcp-types v0.1.0 (/root/workspace/projects/LCP-weaudit/qs_workspace-LCP-weaudit-main-
github/lcp/modules/types)
  Compiling commitments v0.1.0 (/root/workspace/projects/LCP-weaudit/qs_workspace-LCP-weaudit-main-
github/lcp/modules/commitments)
  Compiling webpki v0.22.4
  Compiling attestation-report v0.1.0 (/root/workspace/projects/LCP-weaudit/qs_workspace-LCP-weaudit-
main-github/lcp/modules/attestation-report)
  Compiling keymanager v0.1.0 (/root/workspace/projects/LCP-weaudit/qs_workspace-LCP-weaudit-main-
github/lcp/modules/keymanager)
  Compiling rocksdb v0.21.0
  Compiling store v0.1.0 (/root/workspace/projects/LCP-weaudit/qs_workspace-LCP-weaudit-main-
github/lcp/modules/store)
```

```
Compiling host-environment v0.1.0 (/root/workspace/projects/LCP-weaudit/qs_workspace-LCP-weaudit-main-github/lcp/modules/host-environment)
Compiling ocall-commands v0.1.0 (/root/workspace/projects/LCP-weaudit/qs_workspace-LCP-weaudit-main-github/lcp/modules/ocall-commands)
Compiling light-client v0.1.0 (/root/workspace/projects/LCP-weaudit/qs_workspace-LCP-weaudit-main-github/lcp/modules/light-client)
Compiling ocall-handler v0.1.0 (/root/workspace/projects/LCP-weaudit/qs_workspace-LCP-weaudit-main-github/lcp/modules/ocall-handler)
Compiling ecall-commands v0.1.0 (/root/workspace/projects/LCP-weaudit/qs_workspace-LCP-weaudit-main-github/lcp/modules/ecall-commands)
Compiling context v0.1.0 (/root/workspace/projects/LCP-weaudit/qs_workspace-LCP-weaudit-main-github/lcp/modules/context)
Compiling mock-lc v0.1.0 (/root/workspace/projects/LCP-weaudit/qs_workspace-LCP-weaudit-main-github/lcp/modules/mock-lc)
Compiling service v0.1.0 (/root/workspace/projects/LCP-weaudit/qs_workspace-LCP-weaudit-main-github/lcp/modules/service)
Compiling lcp-client v0.1.0 (/root/workspace/projects/LCP-weaudit/qs_workspace-LCP-weaudit-main-github/lcp/modules/lcp-client)
Compiling tendermint-lc v0.1.0 (/root/workspace/projects/LCP-weaudit/qs_workspace-LCP-weaudit-main-github/lcp/modules/tendermint-lc)
Finished `release` profile [optimized] target(s) in 6m 29s
Running unittests src/lib.rs (target/llvm-cov-target/release/deps/attestation_report-28f6caabb208e6dd)

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

Running unittests src/lib.rs (target/llvm-cov-target/release/deps/commitments-7ae0375f145aacc4)

running 14 tests
test context::tests::test_empty_context_aggregation ... ok
test context::tests::test_context_header ... ok
test context::tests::test_empty_context_serialization ... ok
test context::tests::test_trusting_period_context_serialization ... ok
test context::tests::test_validation_context_and_empty_aggregation ... ok
test context::tests::test_validation_context_aggregation ... ok
test context::tests::test_trusting_period_context ... ok
test context::tests::pt_trusting_period_context ... ok
test message::tests::pt_misbehaviour_with_validation_context ... ok
test message::tests::pt_misbehaviour_with_empty_context ... ok
test message::update_state::tests::test_update_client_message_aggregation ... ok
test message::tests::pt_verify_membership ... ok
test message::tests::pt_update_client_message_with_trusting_period_context ... ok
test message::tests::pt_update_client_message_with_empty_context ... ok

test result: ok. 14 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 2.70s

Running unittests src/lib.rs (target/llvm-cov-target/release/deps/context-09e7ee5a2bae4fe3)

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

Running unittests src/lib.rs (target/llvm-cov-target/release/deps/crypto-614236d203900e56)

running 1 test
test key::tests::test_zeroize_enclave_key ... ok

test result: ok. 1 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.01s

Running unittests src/lib.rs (target/llvm-cov-target/release/deps/ecall_commands-3ec26c945744fdac)

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

Running unittests src/lib.rs (target/llvm-cov-target/release/deps/enclave_api-f9b096b61cc8cc3a)

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s
```

```
Running unittests src/lib.rs (target/llvm-cov-target/release/deps/host-208ccc1cad3534fe)

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

Running unittests src/lib.rs (target/llvm-cov-target/release/deps/host_environment-7f739a30f700d580)

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

Running unittests src/lib.rs (target/llvm-cov-target/release/deps/keymanager-47697b446afe6ac3)

running 1 test
test tests::test_keys ... ok

test result: ok. 1 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.05s

Running unittests src/lib.rs (target/llvm-cov-target/release/deps/lcp_client-ab021f7934ee41a2)

running 3 tests
test client_def::tests::test_compute_eip712_register_enclave_key ... ok
test client_def::tests::test_compute_eip712_update_operators ... ok
test client_def::tests::test_client ... ok

test result: ok. 3 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.04s

Running unittests src/lib.rs (target/llvm-cov-target/release/deps/lcp_proto-141e630736c5a888)

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

Running unittests src/lib.rs (target/llvm-cov-target/release/deps/lcp_types-4dd5fee456b27a1b)

running 10 tests
test height::tests::test_height_ordering ... ok
test height::tests::test_height_add_sub ... ok
test height::tests::test_height ... ok
test host::tests::test_valid_client_id ... ok
test host::tests::test_invalid_client_id ... ok
test time::tests::test_time_from_unix_timestamp_nanos ... ok
test time::tests::test_time_from_unix_timestamp_nanos_overflow ... ok
test height::tests::test_height_str_conversion ... ok
test time::tests::test_time_range ... ok
test any::tests::test_encoding_compatibility_with_proto_any ... ok

test result: ok. 10 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.09s

Running unittests src/lib.rs (target/llvm-cov-target/release/deps/light_client-c2d62967ee8276c8)

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

Running unittests src/lib.rs (target/llvm-cov-target/release/deps/mock_lc-fb665b67a49a36c7)

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

Running unittests src/lib.rs (target/llvm-cov-target/release/deps/ocall_commands-7e98c6ca52b9bdf0)

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

Running unittests src/lib.rs (target/llvm-cov-target/release/deps/ocall_handler-4e230bd18100f4e7)

running 0 tests
```

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

Running unittests src/lib.rs (target/llvm-cov-target/release/deps/remote_attestation-6041aadd6ea58def)

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

Running unittests src/lib.rs (target/llvm-cov-target/release/deps/service-5ad2922059fcae61)

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

Running unittests src/lib.rs (target/llvm-cov-target/release/deps/store-9787660af8fc9571)

running 8 tests

test cache::tests::test_cache_kvs ... ok
test rocksdb::tests::test_concurrent_write_tx_with_same_update_key_1 ... ok
test rocksdb::tests::test_concurrent_include_rollback ... ok
test rocksdb::tests::test_store ... ok
test rocksdb::tests::test_concurrent_write_tx_with_same_update_key_2 ... ok
test rocksdb::tests::test_write_and_snapshot ... ok
test rocksdb::tests::test_concurrent_read_tx ... ok
test rocksdb::tests::test_concurrent_write_different_update_keys ... ok

test result: ok. 8 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 1.45s

Running unittests src/lib.rs (target/llvm-cov-target/release/deps/tendermint_lc-74215dbc883135e1)

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

warning: 87 functions have mismatched data

Filename					Regions	Missed Regions	Cover	Functions
Missed Functions	Executed	Lines	Missed Lines	Cover	Branches	Missed Branches		
Cover								

--								
modules/attestation-report/src/errors.rs					3	3	0.00%	3
3	0.00%	9	9	0.00%	0	0	-	
modules/attestation-report/src/report.rs					59	17	71.19%	23
3	86.96%	120	27	77.50%	0	0	-	
modules/attestation-report/src/verification.rs					30	30	0.00%	5
5	0.00%	37	37	0.00%	0	0	-	
modules/commitments/src/context.rs					156	34	78.21%	31
6	80.65%	495	48	90.30%	0	0	-	
modules/commitments/src/encoder.rs					15	2	86.67%	8
1	87.50%	42	11	73.81%	0	0	-	
modules/commitments/src/errors.rs					6	6	0.00%	6
6	0.00%	18	18	0.00%	0	0	-	
modules/commitments/src/message.rs					77	31	59.74%	28
6	78.57%	157	52	66.88%	0	0	-	
modules/commitments/src/message/misbehaviour.rs					31	16	48.39%	14
7	50.00%	81	38	53.09%	0	0	-	
modules/commitments/src/message/update_state.rs					61	23	62.30%	16
8	50.00%	328	38	88.41%	0	0	-	
modules/commitments/src/message/verify_membership.rs					25	17	32.00%	9
5	44.44%	71	48	32.39%	0	0	-	
modules/commitments/src/proof.rs					15	7	53.33%	10
4	60.00%	40	16	60.00%	0	0	-	
modules/commitments/src/prover.rs					8	2	75.00%	1
0	100.00%	9	0	100.00%	0	0	-	
modules/commitments/src/state.rs					13	5	61.54%	8
2	75.00%	39	5	87.18%	0	0	-	
modules/context/src/context.rs					7	2	71.43%	7
2	71.43%	29	9	68.97%	0	0	-	
modules/crypto/src/errors.rs					2	2	0.00%	2

2	0.00%	6	0.00%	0	0	-	
modules/crypto/src/key.rs				97	46	52.58%	38
19	50.00%	183	80	56.28%	0	-	
modules/crypto/src/traits.rs				2	1	50.00%	2
1	50.00%	10	3	70.00%	0	-	
modules/ecall-commands/src/commands.rs				18	18	0.00%	7
7	0.00%	19	19	0.00%	0	-	
modules/ecall-commands/src/enclave_manage.rs				14	14	0.00%	8
8	0.00%	8	8	0.00%	0	-	
modules/ecall-commands/src/errors.rs				2	2	0.00%	2
2	0.00%	6	6	0.00%	0	-	
modules/ecall-commands/src/light_client.rs				60	60	0.00%	18
18	0.00%	27	27	0.00%	0	-	
modules/ecall-commands/src/messages.rs				70	70	0.00%	17
17	0.00%	111	111	0.00%	0	-	
modules/enclave-api/src/api/command.rs				40	40	0.00%	7
7	0.00%	82	82	0.00%	0	-	
modules/enclave-api/src/api/primitive.rs				44	44	0.00%	2
2	0.00%	58	58	0.00%	0	-	
modules/enclave-api/src/api/proto.rs				55	55	0.00%	6
6	0.00%	53	53	0.00%	0	-	
modules/enclave-api/src/enclave.rs				36	36	0.00%	14
14	0.00%	57	57	0.00%	0	-	
modules/enclave-api/src/errors.rs				6	6	0.00%	6
6	0.00%	18	18	0.00%	0	-	
modules/host-environment/src/lib.rs				3	3	0.00%	3
3	0.00%	9	9	0.00%	0	-	
modules/host/src/enclave.rs				5	5	0.00%	2
2	0.00%	27	27	0.00%	0	-	
modules/host/src/ocalls.rs				40	40	0.00%	6
6	0.00%	78	78	0.00%	0	-	
modules/keymanager/src/errors.rs				6	6	0.00%	6
6	0.00%	18	18	0.00%	0	-	
modules/keymanager/src/lib.rs				264	106	59.85%	45
27	40.00%	414	137	66.91%	0	-	
modules/lcp-client/src/client_def.rs				245	116	52.65%	32
8	75.00%	662	155	76.59%	0	-	
modules/lcp-client/src/client_state.rs				25	12	52.00%	10
4	60.00%	70	17	75.71%	0	-	
modules/lcp-client/src/consensus_state.rs				15	5	66.67%	6
1	83.33%	30	2	93.33%	0	-	
modules/lcp-client/src/errors.rs				6	6	0.00%	6
6	0.00%	18	18	0.00%	0	-	
modules/lcp-client/src/message.rs				55	55	0.00%	21
21	0.00%	113	113	0.00%	0	-	
modules/light-client/src/client.rs				2	0	100.00%	2
0	100.00%	6	0	100.00%	0	-	
modules/light-client/src/context.rs				23	9	60.87%	7
2	71.43%	62	12	80.65%	0	-	
modules/light-client/src/errors.rs				1	1	0.00%	1
1	0.00%	3	3	0.00%	0	-	
modules/light-client/src/ibc.rs				48	47	2.08%	32
31	3.12%	200	194	3.00%	0	-	
modules/light-client/src/path.rs				3	1	66.67%	3
1	66.67%	13	3	76.92%	0	-	
modules/light-client/src/registry.rs				13	12	7.69%	6
5	16.67%	31	28	9.68%	0	-	
modules/mock-lc/src/client.rs				94	36	61.70%	12
7	41.67%	200	62	69.00%	0	-	
modules/mock-lc/src/errors.rs				1	1	0.00%	1
1	0.00%	3	3	0.00%	0	-	
modules/mock-lc/src/message.rs				33	17	48.48%	12
4	66.67%	40	9	77.50%	0	-	
modules/mock-lc/src/state.rs				25	11	56.00%	11
2	81.82%	41	4	90.24%	0	-	
modules/ocall-commands/src/lib.rs				11	11	0.00%	5
5	0.00%	9	9	0.00%	0	-	
modules/ocall-commands/src/log.rs				4	4	0.00%	2
2	0.00%	4	4	0.00%	0	-	
modules/ocall-commands/src/store.rs				7	7	0.00%	3
3	0.00%	3	3	0.00%	0	-	
modules/ocall-handler/src/errors.rs				2	2	0.00%	2

2	0.00%	6	0.00%	0	0	—	
modules/ocall-handler/src/log.rs				6	6	0.00%	1
1	0.00%	5	0.00%	0	0	—	
modules/ocall-handler/src/router.rs				7	7	0.00%	1
1	0.00%	7	0.00%	0	0	—	
modules/ocall-handler/src/store.rs				19	19	0.00%	1
1	0.00%	27	0.00%	0	0	—	
modules/remote-attestation/src/ias.rs				22	22	0.00%	2
2	0.00%	25	0.00%	0	0	—	
modules/remote-attestation/src/ias_utils.rs				260	260	0.00%	30
30	0.00%	380	0.00%	0	0	—	
modules/service/src/elc.rs				36	36	0.00%	12
12	0.00%	38	0.00%	0	0	—	
modules/service/src/enclave.rs				39	39	0.00%	10
10	0.00%	33	0.00%	0	0	—	
modules/service/src/service.rs				7	7	0.00%	4
4	0.00%	40	0.00%	0	0	—	
modules/store/src/cache.rs				30	0	100.00%	10
0	100.00%	72	0	0	0	—	
modules/store/src/host.rs				16	16	0.00%	4
4	0.00%	24	0.00%	0	0	—	
modules/store/src/memory.rs				47	37	21.28%	22
17	22.73%	101	84	0	0	—	
modules/store/src/rocksdb.rs				304	44	85.53%	106
12	88.68%	754	39	0	0	—	
modules/store/src/store.rs				15	9	40.00%	10
6	40.00%	32	17	0	0	—	
modules/store/src/transaction.rs				9	0	100.00%	6
0	100.00%	12	0	0	0	—	
modules/tendermint-lc/src/client.rs				194	194	0.00%	22
22	0.00%	403	403	0	0	—	
modules/tendermint-lc/src/errors.rs				1	1	0.00%	1
1	0.00%	3	3	0	0	—	
modules/tendermint-lc/src/message.rs				39	39	0.00%	15
15	0.00%	51	51	0	0	—	
modules/tendermint-lc/src/state.rs				24	24	0.00%	10
10	0.00%	59	59	0	0	—	
modules/tendermint-lc/src/verifier.rs				50	50	0.00%	8
8	0.00%	97	97	0	0	—	
modules/types/src/any.rs				20	5	75.00%	14
2	85.71%	51	6	0	0	—	
modules/types/src/errors.rs				3	2	33.33%	3
2	33.33%	9	6	0	0	—	
modules/types/src/height.rs				76	16	78.95%	23
8	65.22%	135	28	0	0	—	
modules/types/src/host.rs				64	12	81.25%	18
4	77.78%	126	14	0	0	—	
modules/types/src/sgx.rs				17	12	29.41%	9
6	33.33%	33	21	0	0	—	
modules/types/src/time.rs				81	23	71.60%	21
8	61.90%	117	31	0	0	—	
modules/types/src/transmuter.rs				14	10	28.57%	5
3	40.00%	35	18	0	0	—	
proto/src/prost/ibc.lightclients.lcp.v1.rs				15	9	40.00%	10
6	40.00%	10	6	0	0	—	
proto/src/prost/lcp.service.elc.v1.rs				272	272	0.00%	108
108	0.00%	574	574	0	0	—	
proto/src/prost/lcp.service.enclave.v1.rs				106	106	0.00%	45
45	0.00%	232	232	0	0	—	

--							
TOTAL				3676	2379	35.28%	1065
665	37.56%	7658	4066	46.91%	0	0	—

Compiling enclave-environment v0.1.0 (/root/workspace/projects/LCP-weaudit/qs_workspace-LCP-weaudit-main-github/lcp/enclave-modules/environment)

Compiling ecall-handler v0.1.0 (/root/workspace/projects/LCP-weaudit/qs_workspace-LCP-weaudit-main-github/lcp/enclave-modules/ecall-handler)

Finished `release` profile[optimized] target(s) in 3.13s

Running unittests src/lib.rs (target/release/deps/ecall_handler-be7126cfd32bf332)

running 0 tests

```
test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s
```

```
Doc-tests ecall_handler
```

```
running 0 tests
```

```
test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s
```

```
Finished `release` profile [optimized] target(s) in 0.36s
```

```
Running unittests src/lib.rs (target/release/deps/enclave_environment-7b3bc5ebe6f8ec26)
```

```
running 0 tests
```

```
test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s
```

```
Doc-tests enclave_environment
```

```
running 0 tests
```

```
test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s
```

```
Finished `release` profile [optimized] target(s) in 0.47s
```

```
Running unittests src/lib.rs (target/release/deps/host_api-e740a7b9dda2ee4c)
```

```
running 1 test
```

```
TESTtest api::tests::test_execute_command ... ok
```

```
test result: ok. 1 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.01s
```

```
Doc-tests host_api
```

```
running 0 tests
```

```
test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s
```

```
Finished `release` profile [optimized] target(s) in 0.36s
```

```
Running unittests src/lib.rs (target/release/deps/enclave_runtime-7eaf76511711ce0c)
```

```
running 0 tests
```

```
test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s
```

```
Doc-tests enclave_runtime
```

```
running 0 tests
```

```
test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s
```

```
Finished `release` profile [optimized] target(s) in 0.26s
```

```
Running unittests src/lib.rs (target/release/deps/enclave_utils-e8d3fb096b194e35)
```

```
running 0 tests
```

```
test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s
```

```
Doc-tests enclave_utils
```

```
running 0 tests
```

```
test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s
```

```
root@e050f6605a3f:~/workspace/projects/LCP-weaudit/qs_workspace-LCP-weaudit-main-github/lcp-solidity#
```

```
forge test
```

```
[..] Compiling...
```

```
[..] Compiling 86 files with Solc 0.8.27
```

```
[.] Solc 0.8.27 finished in 8.07s
```

```
Compiler run successful with warnings:
```

```
Warning (2018): Function state mutability can be restricted to pure
```

```
--> test/LCPCommitmentTest.t.sol:111:5:
```

```

111 |     function testParseUpdateStateProxyMessageEmptyValidationContext() public {
    |     ^ (Relevant source part starts here and spans across multiple lines).

Warning (2018): Function state mutability can be restricted to pure
--> test/LCPCCommitmentTest.t.sol:205:5:
    |
205 |     function testParseUpdateStateProxyMessage() public {
    |     ^ (Relevant source part starts here and spans across multiple lines).

Warning (2018): Function state mutability can be restricted to pure
--> test/LCPCCommitmentTest.t.sol:311:5:
    |
311 |     function testVerifyMembershipProxyMessage() public {
    |     ^ (Relevant source part starts here and spans across multiple lines).

Warning (2018): Function state mutability can be restricted to pure
--> test/LCPCCommitmentTest.t.sol:357:5:
    |
357 |     function testMisbehaviourProxyMessage() public {
    |     ^ (Relevant source part starts here and spans across multiple lines).

Warning (2018): Function state mutability can be restricted to pure
--> test/LCPUtilsTest.t.sol:11:5:
    |
11 |     function testAttestationTimestampToSeconds(uint40 secs, string memory nonce) public {
    |     ^ (Relevant source part starts here and spans across multiple lines).

Warning (2018): Function state mutability can be restricted to pure
--> test/LCPUtilsTest.t.sol:36:5:
    |
36 |     function testRfc5280TimeToSecondsGeneralizedTime(uint40 secs) public {
    |     ^ (Relevant source part starts here and spans across multiple lines).

Warning (2018): Function state mutability can be restricted to pure
--> test/LCPUtilsTest.t.sol:55:5:
    |
55 |     function testRfc5280TimeToSecondsUTCTime(uint40 secs) public {
    |     ^ (Relevant source part starts here and spans across multiple lines).

Ran 1 test for test/ContractUpgrade.t.sol:ContractUpgrade
[PASS] testUpgrade() (gas: 3143)
Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 98.27ms (20.28ms CPU time)

Ran 1 test for test/LCPCClientBenchmark.t.sol:CachedEnclaveRegistrationBenchmark
[PASS] testRegisterEnclaveKey() (gas: 345263)
Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 178.59ms (19.97ms CPU time)

Ran 1 test for test/LCPCClientBenchmark.t.sol:UpdateClientBenchmark
[PASS] testUpdateClient() (gas: 211430)
Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 186.22ms (2.36ms CPU time)

Ran 1 test for test/LCPCClientBenchmark.t.sol>CreateClientBenchmark
[PASS] testCreateClient() (gas: 393985)
Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 11.34ms (6.96ms CPU time)

Ran 1 test for test/LCPCClientBenchmark.t.sol:NoCacheEnclaveRegistrationBenchmark
[PASS] testRegisterEnclaveKey() (gas: 673379)
Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 21.65ms (11.71ms CPU time)

Ran 5 tests for test/LCPCCommitmentTest.t.sol:LCPCCommitmentTest
[PASS] testMisbehaviourProxyMessage() (gas: 60993)
[PASS] testParseUpdateStateProxyMessage() (gas: 83236)
[PASS] testParseUpdateStateProxyMessageEmptyValidationContext() (gas: 91792)
[PASS] testTrustingPeriodContext() (gas: 17179)
[PASS] testVerifyMembershipProxyMessage() (gas: 27127)
Suite result: ok. 5 passed; 0 failed; 0 skipped; finished in 2.02ms (1.52ms CPU time)

Ran 3 tests for test/LCPUtilsTest.t.sol:LCPUtilsTest
[PASS] testAttestationTimestampToSeconds(uint40,string) (runs: 256, μ: 22657, ~: 22644)
[PASS] testRfc5280TimeToSecondsGeneralizedTime(uint40) (runs: 256, μ: 22906, ~: 22859)

```

```
[PASS] testRfc5280TimeToSecondsUTCTime(uint40) (runs: 256, μ: 22567, ~: 22466)
Suite result: ok. 3 passed; 0 failed; 0 skipped; finished in 124.96ms (124.51ms CPU time)

Ran 3 tests for test/ReportTest.t.sol:ReportTest
[PASS] testAvrForDebugEnclave() (gas: 207215)
[PASS] testValidateAdvisories() (gas: 72214)
[PASS] testVerify() (gas: 1740008)
Suite result: ok. 3 passed; 0 failed; 0 skipped; finished in 20.79ms (20.28ms CPU time)

Ran 5 tests for test/LCPCClientOperator.t.sol:LCPCClientOperatorTest
[PASS] testPreComputationValues() (gas: 4269)
[PASS] testRegisterEnclaveKeyMultiOperators() (gas: 2331523)
[PASS] testRegisterEnclaveKeyOperatorDedicatedAVR() (gas: 2422216)
[PASS] testUpdateClientMultiOperators() (gas: 5637389)
[PASS] testUpdateOperators() (gas: 920760)
Suite result: ok. 5 passed; 0 failed; 0 skipped; finished in 189.83ms (186.49ms CPU time)

Ran 4 tests for test/LCPCClientTest.t.sol:LCPCClientTest
[PASS] testIASClientPermissioned() (gas: 3324188)
[PASS] testIASClientPermissionless() (gas: 3228330)
[PASS] testSimulationClientPermissioned() (gas: 2615528)
[PASS] testSimulationClientPermissionless() (gas: 2519422)
Suite result: ok. 4 passed; 0 failed; 0 skipped; finished in 207.36ms (281.24ms CPU time)

Ran 5 tests for test/CertificateTest.t.sol:CertificateTest
[PASS] testIASCertVerification() (gas: 308873)
[PASS] testInvalidRootCerts() (gas: 30197)
[PASS] testInvalidSigningCerts() (gas: 182024)
[PASS] testSimulationCertVerification() (gas: 294879)
[PASS] testValidSigningCerts() (gas: 798758)
Suite result: ok. 5 passed; 0 failed; 0 skipped; finished in 7.03s (8.61s CPU time)

Ran 11 test suites in 7.06s (8.07s CPU time): 30 tests passed, 0 failed, 0 skipped (30 total tests)
```

Code Coverage

For `lcp-solidity` repository, run `forge coverage` . The branch coverage is a bit low and we suggest to improve the test suite to be > 90% branch coverage.

File	% Lines	% Statements	% Branches	% Funcs
contracts/AVRValidator.sol	93.43% (128/137)	90.66% (165/182)	60.29% (41/68)	100.00% (15/15)
contracts/Asn1Decode.sol	61.11% (33/54)	51.25% (41/80)	21.88% (7/32)	63.16% (12/19)
contracts/LCPCClient.sol	100.00% (1/1)	100.00% (1/1)	100.00% (0/0)	100.00% (1/1)
contracts/LCPCClientBase.sol	64.37% (168/261)	68.71% (213/310)	25.00% (20/80)	69.23% (18/26)
contracts/LCPCClientOwnable Upgradeable.sol	0.00% (0/3)	0.00% (0/3)	100.00% (0/0)	0.00% (0/3)
contracts/LCPCommitment.s ol	82.14% (23/28)	86.84% (33/38)	66.67% (6/9)	100.00% (7/7)
contracts/LCPOperator.sol	100.00% (6/6)	100.00% (12/12)	100.00% (0/0)	100.00% (6/6)
contracts/LCPProtoMarshaler .sol	36.84% (14/38)	39.22% (20/51)	20.00% (2/10)	50.00% (4/8)

File	% Lines	% Statements	% Branches	% Funcs
contracts/LCPUtils.sol	86.36% (19/22)	88.46% (23/26)	66.67% (8/12)	100.00% (4/4)
contracts/proto/ibc/lightclients/lcp/v1/LCP.sol	28.50% (165/579)	28.15% (199/707)	25.00% (33/132)	24.71% (21/85)
test/LCPClientBenchmark.t.sol	96.30% (26/27)	96.77% (30/31)	50.00% (1/2)	87.50% (7/8)
test/TestHelper.t.sol	95.00% (38/40)	97.01% (65/67)	33.33% (1/3)	100.00% (17/17)
Total	51.92% (621/1196)	53.18% (802/1508)	34.20% (119/348)	56.28% (112/199)

Changelog

- 2024-11-12 - Initial report
- 2024-12-09 - Fix review report

About Quantstamp

Quantstamp is a global leader in blockchain security. Founded in 2017, Quantstamp’s mission is to securely onboard the next billion users to Web3 through its best-in-class Web3 security products and services.

Quantstamp’s team consists of cybersecurity experts hailing from globally recognized organizations including Microsoft, AWS, BMW, Meta, and the Ethereum Foundation. Quantstamp engineers hold PhDs or advanced computer science degrees, with decades of combined experience in formal verification, static analysis, blockchain audits, penetration testing, and original leading-edge research.

To date, Quantstamp has performed more than 500 audits and secured over \$200 billion in digital asset risk from hackers. Quantstamp has worked with a diverse range of customers, including startups, category leaders and financial institutions. Brands that Quantstamp has worked with include Ethereum 2.0, Binance, Visa, PayPal, Polygon, Avalanche, Curve, Solana, Compound, Lido, MakerDAO, Arbitrum, OpenSea and the World Economic Forum.

Quantstamp’s collaborations and partnerships showcase our commitment to world-class research, development and security. We’re honored to work with some of the top names in the industry and proud to secure the future of web3.

Notable Collaborations & Customers:

- Blockchains: Ethereum 2.0, Near, Flow, Avalanche, Solana, Cardano, Binance Smart Chain, Hedera Hashgraph, Tezos
- DeFi: Curve, Compound, Maker, Lido, Polygon, Arbitrum, SushiSwap
- NFT: OpenSea, Parallel, Dapper Labs, Decentraland, Sandbox, Axie Infinity, Illuvium, NBA Top Shot, Zora
- Academic institutions: National University of Singapore, MIT

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 or other making available of the report to you by Quantstamp.

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. 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 any website and shall have no liability whatsoever to any person or entity for the accuracy or completeness of any output generated by such software.

Disclaimer

The review and this report are provided on an as-is, where-is, and as-available basis. To the fullest extent permitted by law, Quantstamp disclaims all warranties, expressed 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. You agree that access and/or use of the report and other results of the review, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. 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. This report is based on the scope of materials and documentation provided for a limited review at the time provided. You acknowledge that Blockchain technology remains under development and is subject to unknown risks and flaws and, as such, the report may not be complete or inclusive of all vulnerabilities. The review is limited to the materials identified in the report and does not extend to the compiler layer, or any other areas beyond the programming language, or programming aspects that could present security risks. The report does not indicate the endorsement by Quantstamp of any particular project or team, nor guarantee its security, and and may not be represented as such. No third party is entitled to rely on the report in any any way, including for the purpose of making any decisions to buy or sell a product, product, service or any other asset. Quantstamp does not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party, or or any open source or third-party software, code, libraries, materials, or information to, to, called by, referenced by or accessible through the report, its content, or any related related services and products, any hyperlinked websites, or any other websites or mobile applications, and we will not be a party to or in any way be responsible for monitoring any any transaction between you and any third party. 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.

