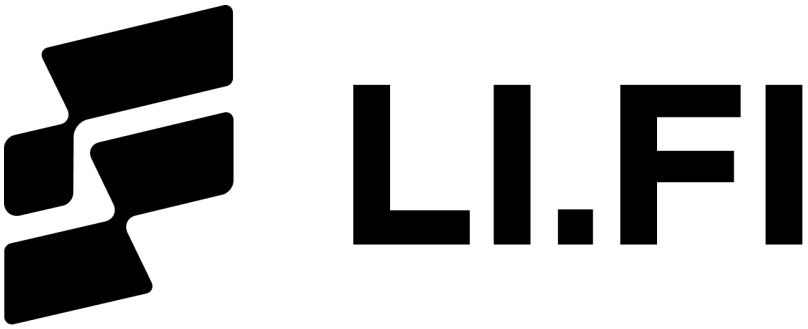

LI.FI Security Review

PioneerFacet.sol(v1.0.0)

Security Researcher

Sujith Somraaj (somraajsujith@gmail.com)

Report prepared by: Sujith Somraaj

June 26, 2025

Contents

1 About Researcher **2**

2 Disclaimer **2**

3 Scope **2**

4 Risk classification **2**
4.1 Impact . . . 2
4.2 Likelihood . . . 3
4.3 Action required for severity levels . . . 3

5 Executive Summary **3**

6 Findings **4**
6.1 Low Risk . . . 4
6.1.1 Sanity check `refundAddress` in bridging functions . . . 4
6.1.2 Sanity check `transactionId` in bridging functions . . . 4
6.2 Informational . . . 4
6.2.1 Rename event `RefundAddressRegistered` to `RefundAddressRegisterForPioneer` . . . 4
6.2.2 Missing inline parameter documentation . . . 5
6.2.3 Validate swap output and bridging asset in `swapAndStartBridgeTokensViaPioneer` . . . 5
6.2.4 Index parameters in `RefundAddressRegistered` event . . . 5

1 About Researcher

Sujith Somraaj is a distinguished security researcher and protocol engineer with over eight years of comprehensive experience in the Web3 ecosystem.

In addition to working as a Security researcher at Spearbit, Sujith is also the security researcher and advisor for leading bridge protocol LI.FI and also is a former founding engineer and current CISO at Superform, a yield aggregator with over $170M in TVL.

Sujith has experience working with protocols / funds including Edge Capital, Berachain, Optimism, Sonic, Monad, Blast, ZkSync, Decent, Drips, SuperSushi Samurai, DistrictOne, Omni-X, Centrifuge, Superform-V2, Tea.xyz, Paintswap, Bitcorn, Sweep n' Flip, Byzantine Finance, Variational Finance, Satsbridge, Earthfast and Angles

Learn more about Sujith on sujithsomraaj.xyz or on cantina.xyz

2 Disclaimer

Note that this security audit is not designed to replace functional tests required before any software release, and does not give any warranties on finding all possible security issues of that given smart contract(s) or blockchain software. i.e., the evaluation result does not guarantee against a hack (or) the non existence of any further findings of security issues. As one audit-based assessment cannot be considered comprehensive, I always recommend proceeding with several audits and a public bug bounty program to ensure the security of smart contract(s). Lastly, the security audit is not an investment advice.

This review is done independently by the reviewer and is not entitled to any of the security agencies the researcher worked / may work with.

3 Scope

- src/Facets/PioneerFacet.sol(v1.0.0)

4 Risk classification

Severity level	Impact: High	Impact: Medium	Impact: Low
Likelihood: high	Critical	High	Medium
Likelihood: medium	High	Medium	Low
Likelihood: low	Medium	Low	Low

4.1 Impact

High leads to a loss of a significant portion (>10%) of assets in the protocol, or significant harm to a majority of users.

Medium global losses <10% or losses to only a subset of users, but still unacceptable.

Low losses will be annoying but bearable — applies to things like griefing attacks that can be easily repaired or even gas inefficiencies.

4.2 Likelihood

High almost certain to happen, easy to perform, or not easy but highly incentivized

Medium only conditionally possible or incentivized, but still relatively likely

Low requires stars to align, or little-to-no incentive

4.3 Action required for severity levels

Critical Must fix as soon as possible (if already deployed)

High Must fix (before deployment if not already deployed)

Medium Should fix

Low Could fix

5 Executive Summary

Over the course of 6 hours in total, LI.FI engaged with the researcher to audit the contracts described in section 3 of this document ("scope").

In this period of time a total of 6 issues were found.

Project Summary	
Project Name	LI.FI
Repository	lifinance/contracts
Commit	af8cfaf1
Audit Timeline	June 25, 2025
Methods	Manual Review
Documentation	Medium
Test Coverage	Medium

Issues Found	
Critical Risk	0
High Risk	0
Medium Risk	0
Low Risk	2
Gas Optimizations	0
Informational	4
Total Issues	**6**

6 Findings

6.1 Low Risk

6.1.1 Sanity check `refundAddress` in bridging functions

Context: PioneerFacet.sol#L100

Description: The `refundAddress` parameter is used by off-chain actors to issue a refund to the user if the bridging process cannot be completed for various reasons. Hence, providing an invalid refund address can result in permanent loss of user funds. Therefore, it's better to sanity-check those values.

Recommendation: Consider adding validations to ensure `refundAddress` is set:

```
+ if( _pioneerData.refundAddress == address(0)) revert("Invalid refund address");
  emit RefundAddressRegistered(_pioneerData.refundAddress);
```

LI.FI: Fixed in cd009186daeef2049e8555e4e0d5ffd9c024ddda

Researcher: Verified fix.

6.1.2 Sanity check `transactionId` in bridging functions

Context: PioneerFacet.sol#L42, PioneerFacet.sol#L64

Description: Bridging through the pioneer facet occurs by sending the necessary source chain tokens to the Pioneer EOA. A solver will then retrieve the transaction ID from the events emitted by LiFiDiamond to continue processing the bridging.

However, there is no sanity check for this input parameter. In other facets, this value is often overlooked, but since it's of value, sanity checking it can protect users against unexpected behavior.

Recommendation: Consider adding basic sanity checks on transaction IDs as follows:

```
function startBridgeTokensViaPioneer( ILiFi.BridgeData memory _bridgeData)
  external payable nonReentrant refundExcessNative(payable(msg.sender))
  validateBridgeData(_bridgeData) doesNotContainSourceSwaps(_bridgeData)
  doesNotContainDestinationCalls(_bridgeData) {
+ if (_bridgeData.transactionId == bytes32(0)) revert InvalidTransactionId();
  ....
}
```

LI.FI: Fixed in 3d0ab5e2ba6571b22380e2434f775ddd871ae736

Researcher: Verified fix

6.2 Informational

6.2.1 Rename event `RefundAddressRegistered` to `RefundAddressRegisterForPioneer`

Context: PioneerFacet.sol#L19

Description: The `RefundAddressRegistered` event is used by the off-chain component to send refunds. However, the address that emits this event will always be the LiFi diamond contract. Therefore, naming the event specifically for Pioneer will improve off-chain tracking, especially if there is a need to add similar events for other bridges.

Recommendation: Consider renaming the event to pioneer-specific. Additionally, it would be great if versioning could be implemented.

LI.FI: Fixed in ef9d46f30af53f8bb930b236caef73c9cbd910ac

Researcher: Verified fix.

6.2.2 Missing inline parameter documentation

Context: PioneerFacet.sol#L42, PioneerFacet.sol#L64, PioneerFacet.sol#L90

Description: The functions `swapAndStartBridgeTokensViaPioneer`, `_startBridge` and `startBridgeTokensViaPioneer` lack inline-documentation for the **_pioneerData** parameter.

Recommendation: Consider adding documentation for all the function parameters.

LI.FI: Fixed in d56a11ba78029db106bd56c5e1a27893d6c970da

Researcher: Verified fix.

6.2.3 Validate swap output and bridging asset in `swapAndStartBridgeTokensViaPioneer`

Context: PioneerFacet.sol#L64

Description: The `swapAndStartBridgeTokensViaPioneer()` function is used to swap assets to the bridging asset before initiating bridging.

However, this function does not validate if the swap output equals the bridging token, leading to unexpected behavior. While this change can consume slightly more gas, it will protect the user against unexpected behavior.

Recommendation: Consider fixing the issue as follows:

```
if(_bridgeData.sendingAssetId != _swapData[_swapData.length - 1].receivingAssetId) revert("Invalid Swap
↪  Data");
```

LI.FI: If the swap outputs a wrong token, then the contract won't have enough tokens to make the transfer to the EOA and revert.

Researcher: Acknowledged.

6.2.4 Index parameters in `RefundAddressRegistered` event

Context: PioneerFacet.sol#L19

Description: The event `RefundAddressRegistered` is used to track refund addresses off-chain. It has one parameter called `refundTo`, which represents the address provided by the user to receive refunds in the event of a transaction failure. This parameter could be indexed for better off-chain tracking.

Recommendation: Consider adding the indexed keyword to the event parameter as follows:

```
-  event RefundAddressRegistered(address refundTo);
+  event RefundAddressRegistered(address indexed refundTo);
```

LI.FI: Fixed in d18f6f0b6533453bae28c5d9e6db79c3b15203e2

Researcher: Verified fix.