tez fast withdrawal for Etherlink

Tezos Foundation

Independent security assessment report

inference

Report version: 1.0 / date: 17.04.2025

inference

Table of contents

inference

Version / Date	Description
1.0 / 17.04.2025	Final version

Summary

Inference AG was engaged by the Tezos Foundation to perform an independent security assessment of Etherlink's tez fast withdrawal bridge. The tez fast withdrawal bridge allows users transferring tez from Etherlink to Tezos L1 to receive their tez more quickly in exchange for a fee. These tez are provided by others willing to front the tez in return for the offered fee, prior to the completion of the standard withdrawal process, which requires waiting until the corresponding message is cemented.

Inference AG performed the security assessment based on the agreed scope, following our approach and activities as outlined in the "Project overview" chapter between the 8th of April 2025 and the 14th of April 2025. Feedback from the Etherlink's tez fast withdrawal bridge team was received and Inference performed a reassessment.

Based on our scope and our performed activities, our security assessment revealed a few security issues rated with a low severity. Additionally, different observations were also made. All of these were resolved with appropriate actions, improving the quality and security of the bridging solution.

This report only shows remaining open or partly resolved issues and observations.

Overview on issues and observations

At Inference AG we separate the findings that we identify in our security assessments in two categories:

- Security issues represent risks to either users of the platform, owners of the contract, the environment of the blockchain, or one or more of these. For example, the possibility to steal funds from the contract, or to lock them in the contract, or to set the contract in a state that renders it unusable are all potential security issues;
- Observations represent opportunities to create a better performing contract, saving gas fees, integrating more efficiently into the existing environment, and creating a better user experience overall. For example, code optimizations that save execution time (and thus gas fees),better compliance to existing standards, and following secure coding best practices are all examples of observations.

Details for each reported issue or observation can be obtained from the "Security issues" and "Observations" sections.

	Severity / Status
Security issues	
There are no open, known security issues.	
Observations	
There are no open observations.	

Project overview

Scope

The scope of the security assessment was the following smart contract:

- Tezos L1 side:
  - Fast-Withdrawal:
    i. CameLIGO:
    /tezos/contracts/fast-withdrawal/fast-withdrawal.mligo, including all imported files and code required to compile the smart contract.
    ii. Michelson:
    /tezos/build/fast-withdrawal.tz

All files in scope were made available via a source code repo: https://github.com/baking-bad/etherlink-bridge/ and our initial security assessment considered commit "02e9d7eeff52480ac0001c6d4a9a55f2c047c25a"[1].

Our reassessment considered commit:
"4bd007485be42b6abfeb1865e7fb857084021977"[2].

Scope limitations

There are no specific limitations within the outlined scope.

[1] The sha256sum hash of the repository's ".zip" file is:
ba497bd4384950bd7d85cb697f0fd237d211eb8e3a084d440961fca02e759b45
[2] The sha256sum hash of the repository's ".zip" file is:
36bb3860d08c34be6c7adfdd59ecd15620c4dd955b1f9d214cdc28572c35d3a0

Methodology

Inference's methodology for smart contract security assessments on Tezos is a combination of a source code review of the smart contract source code in the high-level language (e.g. Ligo or SmartPy), and the resulting compiled code in Michelson. This approach provides additional assurance that the compiler producing the Michelson code is correct, and does not introduce any security issues. Furthermore, this approach fosters a better understanding for smart contract users of what has been reviewed and what has been effectively deployed on-chain.

Inference's methodology for smart contract security assessments on Ethereum comprises a source code review of the smart contract source code in the high-level language Solidity.

In order to ensure a high quality in our security assessments, Inference is using subject matter experts having a high adversarial scenario mindset to spot potential issues in smart contracts under review. Additionally, we apply checklists derived from good practices and commonly known issues based on the Tezos smart contract assessment checklist to document our work and to ensure good issue coverage.

Furthermore, Inference maintains regular communications with the smart contract development team to ensure a correct understanding of the smart contract solution and environment, but also to make teams aware of any observations as soon as possible.

Inference's internal quality assurance procedures ensure that results of security assessments are challenged for completeness and appropriateness by a second independent expert.

Objectives

The objectives are the identification of security issues with regards to the assessed smart contracts and their conceptual design and specification. The security assessment also focuses on adversarial scenarios on specific use cases which have been listed in appendix "Adversarial scenarios". These were identified together with the Etherlink's tez fast withdrawal bridge team and checked during our security assessment.

Activities

Our security assessment activities for the defined scope were:

- Source code review of smart contract code written in CameLIGO
- Source code review of the smart contract in Michelson

We applied the checklist for smart contract security assessments on Tezos, version 2.0 obtained from https://github.com/InferenceAG/TezosSecurityAssessmentChecklist. We applied the following security checklist tables:

- System / Platform
- Storage
- Gas issues and efficiency
- Code issues
- Transactions
- Entrypoint
- On-chain views
- Admin / Operator functions
- Other topics & test cases

Our activities for the reassessment were:

- Source code review of changes in the smart contract code in CameLIGO
- Source code review of changes in the smart contracts in Michelson
- Reassessing security issues and observations from the initial assessment in case they are claimed to be resolved

Security issues

There are no open known security issues.

Observations

There are no open observations.

Disclaimer

This security assessment report (“Report”) by Inference AG (“Inference”) is solely intended for Tezos Foundation (“Client”) with respect to the Report’s purpose as agreed by the Client. The Report may not be relied upon by any other party than the Client and may only be distributed to a third party or published with the Client’s consent. If the Report is published or distributed by the Client or Inference (with the Client’s approval) then it is for information purposes only and Inference does not accept or assume any responsibility or liability for any other purpose or to any other party.

Security assessments of a software or technology cannot uncover all existing vulnerabilities. Even an assessment in which no weaknesses are found is not a guarantee of a secure system. Generally, code assessments enable the discovery of vulnerabilities that were overlooked during development and show areas where additional security measures are necessary. Within the Client’s defined time frame and engagement, Inference has performed an assessment in order to discover as many vulnerabilities of the technology or software analysed as possible. The focus of the Report’s security assessment was limited to the general items and code parts defined by the Client. The assessment shall reduce risks for the Client but in no way claims any guarantee of security or functionality of the technology or software that Inference agreed to assess. As a result, the Report does not provide any warranty or guarantee regarding the defect-free or vulnerability-free nature of the technology or software analysed.

In addition, the Report only addresses the issues of the system and software at the time the Report was produced. The Client should be aware that blockchain technology and cryptographic assets present a high level of ongoing risk. Given the fact that inherent limitations, errors or failures in any software development process and software product exist, it is possible that even major failures or malfunctions remain undetected by the Report. Inference did not assess the underlying third party infrastructure which adds further risks. Inference relied on the correct performance and execution of the included third party technology itself.

Appendix

Adversarial scenarios

The following adversarial scenarios have been identified and checked during our security assessment.

Scenario	Assessment result
Permanently locked assets in the smart contract.	**Ok** Nothing identified.
Users fronting tez for fast withdrawal do not get the bridged tez.	**NOTE** Nothing identified. However, if the user initiating the payout_withdrawal entrypoint provides incorrect input values (e.g., an invalid timestamp), the user will not receive any tez later.
Fronted or bridged tez can be redirected to unauthorized parties.	**Ok** Nothing identified.
Users using the bridge can get the fronted and bridged tez, even if the users who fronted the tez acted correctly—within the allowed time and with all parameters set properly.	**Ok** Nothing identified.

Risk rating definition for smart contracts

Severities are quantified with two dimensions, roughly defined as follows, whereas the examples have to be regarded as indication only:

Probability of occurrence / materialisation of an issue
(bullets for a category are linked with each other with “and/or” condition.)

- Low:
  - A trusted / privileged role is required.
  - Contract may end up in the issue if other conditions, which are also unlikely to happen, are required.
- Medium:
  - A specific role or contract state is required to trigger the issue.
  - Contract may end up in the issue if another condition is fulfilled as well.
- High:
  - Anybody can trigger the issue.
  - Contract’s state will over the short or long term end up in the issue.

Impact:
(bullets for a category are linked with each other with “and/or” condition.)

- Low:
  - Non-compliance with TZIP standards
  - Unclear error messages
  - Confusing structures
- Medium:
  - A minor amount of assets can be withdrawn or destroyed.
- High:
  - Not inline with the specification
  - A non-minor amount of assets can be withdrawn or destroyed.
  - Entire or part of the contract becomes unusable.

Severity:

	Low impact	Medium impact	High impact
High probability	**High**	**Critical**	**Critical**
Medium probability	**Medium**	**High**	**Critical**
Low probability	**Low**	**Medium**	**High**

Glossary

Term	Description
Bridge	Solution to transfer tokens from a chain or a layer to a different chain or layer
Etherlink	Smart optimistic rollup
Ligo	High level smart contract language. Website: https://ligolang.org/