



Three Sigma

Code Audit

M^0 [Labs]

M^0 Labs

Coordination layer for permissioned institutional actors to generate \$M

Disclaimer

The ensuing audit offers no assertions or assurances about the code's security. It cannot be deemed an adequate judgment of the contract's correctness on its own. The authors of this audit present it solely as an informational exercise, reporting the thorough research involved in the secure development of the intended contracts, and make no material claims or guarantees regarding the contract's post-deployment operation. The authors of this report disclaim all liability for all kinds of potential consequences of the contract's deployment or use. Due to the possibility of human error occurring during the code's manual review process, we advise the client team to commission several independent audits in addition to a public bug bounty program.

Table of Contents

Code Audit

M^O Coordination layer for permissioned institutional actors to generate \$M

Table of Contents

Disclaimer	3
Summary	7
Scope	9
Methodology	11
Project Dashboard	13
Code Maturity Evaluation	16
Findings	19
3S-M^0-L01	19
3S-M^0-L02	20
3S-M^0-N01	21
3S-M^0-N02	22

Summary

Three Sigma audited M⁰ in a 3 day engagement. The audit was conducted from 17-07-2024 to 19-07-2024.

Protocol Description

A neutral value transmission framework able to permissionlessly mint currencies under decentralized governance. The purpose of \$M is to become a superior building block for value representation, by combining the convenience of digital money with the risk profile of physical cash.

Scope

Filepath	nSLOC
src/libs/IndexingMath.sol	46
src/Migratable.sol	20
src/Proxy.sol	31
src/WrappedMToken.sol	239
Total	336

00000000 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000020 00 00 00 00 3B A3 ED FD 7A 7B 12 B2 7A C7 2C 3E . . . ;Éíÿz{.
00000030 67 76 8F 61 7F C8 1B C3 88 8A 51 32 3A 9F B8 AA gv.a.È.Ä^ŠQ
00000040 4B 1E 5E 4A 29 AB 5F 49 FF FF 00 1D 1D AC 2B 7C K.^J)«_Iyy.
00000050 01 01 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00
00000060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000070 00 00 00 00 00 00 FF FF FF FF 4D 04 FF FF 00 1D ŸŸŸŸŸ
00000080 01 04 45 54 68 65 20 54 69 6D 65 73 20 30 33 2F ..EThe Tim
00000090 4A 61 6E 2F 32 30 30 39 20 43 68 61 6E 63 65 6C Jan/2009 Ch
000000A0 6C 6F 72 20 6F 6E 20 62 72 69 6E 6B 20 6F 66 20 lor on brin
000000B0 73 65 63 6F 6E 64 20 62 61 69 6C 6F 75 74 20 66 second bail
000000C0 6F 72 20 62 61 6E 6B 73 FF FF FF FF 01 00 F2 05 or banksŸŸŸŸ
000000D0 2A 01 00 00 00 43 41 04 67 8A FD B0 FE 55 48 27 * . . . CA.gŠŸŸ
000000E0 19 67 F1 A6 71 30 B7 10 5C D6 A8 28 E0 39 09 A6 .gñ|q0·.\ŸŸ
000000F0 79 62 E0 EA 1F 61 DE B6 49 F6 BC 3F 4C EF 38 C4 ybâê.ab¶IÖ¼
00000100 F3 55 04 E5 1E C1 12 DE 5C 38 4D F7 BA 0B 8D 57 óU.ã.Á.p\8M
00000110 8A 4C 70 2B 6B F1 1D 5F AC 00 00 00 00 ŠLp+kñ._~.
00000000 30 31 30 30 30 30 30 30 36 66 65 32 38 63 30 61 01000000006f
00000000 **Methodology** 62 33 37 32 63 31 61 36 61 32 34 36 b6f1b3726c1
00000020 61 65 36 33 66 37 34 66 39 33 31 65 38 33 36 35 ae63f7f4639
00000000 **Code Audit** 35 61 30 38 39 63 36 38 64 36 31 39 30 30 e15a089c638
00000000 **M^O** Coordination layer for permissioned institutional actors to generate \$M 30
00000050 31 65 34 62 61 37 34 34 62 62 62 65 36 38 30 65 1e4ba7445bb
00000060 31 66 65 65 31 34 36 37 37 62 61 31 61 33 63 33 1fee1456376
00000070 35 34 30 62 66 37 62 31 63 64 62 36 30 36 65 38 5400bfb1bcd
00000080 35 37 32 33 33 65 30 65 36 31 62 63 36 36 34 39 57233e0e6bc
00000090 66 66 66 66 30 30 31 64 30 31 65 33 36 32 39 39 fff0001d01e
000000A0 30 31 30 31 30 30 30 30 30 30 30 30 31 30 30 30 30 01010000000
000000B0 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 00000000000
000000C0 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 00000000000
000000D0 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 00000000000
000000E0 30 30 30 30 30 30 30 30 30 30 30 30 30 66 66 66 66 00000000000
000000F0 66 66 66 66 30 37 30 34 66 66 66 66 30 30 31 64 ffff074ffff0
00000100 30 31 30 34 66 66 66 66 66 66 66 66 30 31 30 30 0104ffffff
00000110 66 32 30 35 32 61 30 31 30 30 30 30 30 30 30 34 33 f2052a01000
00000120 34 31 30 34 39 36 62 35 33 38 65 38 35 33 35 31 4104946b538
00000130 39 63 37 32 36 61 32 63 39 31 65 36 31 65 63 31 9c726ac91e6
00000140 31 36 30 30 61 65 31 33 39 30 38 31 33 61 36 32 16000ae1390
00000150 37 63 36 36 66 62 38 62 65 37 39 34 37 62 65 36 7c66fb8be79
00000160 33 63 35 32 64 61 37 35 38 39 33 37 39 35 31 35 3c52da73589
00000170 64 34 65 30 61 36 30 34 66 38 31 34 31 37 38 31 d4e0a646381
00000180 65 36 32 32 39 34 37 32 31 31 36 36 62 66 36 32 e6224947321
00000190 31 65 37 33 61 38 32 63 62 66 32 33 34 32 63 38 1e73a823cbf
000001A0 35 38 65 65 61 63 30 30 30 30 30 30 30 30 58eacc00000
00000000 30 31 30 30 30 30 30 30 34 38 36 30 65 62 31 38 01000000486

Methodology

To begin, we reasoned meticulously about the contract's business logic, checking security-critical features to ensure that there were no gaps in the business logic and/or inconsistencies between the aforementioned logic and the implementation. Second, we thoroughly examined the code for known security flaws and attack vectors. Finally, we discussed the most catastrophic situations with the team and reasoned backwards to ensure they are not reachable in any unintentional form.

Taxonomy

In this audit we report our findings using as a guideline Immunefi's vulnerability taxonomy, which can be found at immunefi.com/severity-updated/. The final classification takes into account the severity, according to the previous link, and likelihood of the exploit. The following table summarizes the general expected classification according to severity and likelihood; however, each issue will be evaluated on a case-by-case basis and may not strictly follow it.

Severity / Likelihood	LOW	MEDIUM	HIGH
NONE	None		
LOW	Low		
MEDIUM	Low	Medium	Medium
HIGH	Medium	High	High
CRITICAL	High	Critical	Critical

Project Dashboard

Application Summary

Name	M^0
Commit	0dd7b4553009e3d87298a2f9e6aec2a89ba308ad
Fix Commit	55896c2f37a13a39fae933d52a2b6687ff4c9408
Language	Solidity
Platform	Ethereum

Engagement Summary

Timeline	17-07-2024 to 19-07-2024
Nº of Auditors	1
Review Time	3 days

Vulnerability Summary

Issue Classification	Found	Addressed	Acknowledged
Critical	0	0	0
High	0	0	0
Medium	0	0	0
Low	2	1	1

None	2	0	2
------	---	---	---

Category Breakdown

Suggestion	4
Documentation	0
Bug	0
Optimization	0
Good Code Practices	0

00000000	01 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
00000010	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
00000020	00 00 00 00 3B A3 ED FD	7A 7B 12 B2 7A C7 2C 3E;Éíÿz{.
00000030	67 76 8F 61 7F C8 1B C3	88 8A 51 32 3A 9F B8 AA	...	gv.a.È.Ã~SQ
00000040	4B 1E 5E 4A 29 AB 5F 49	FF FF 00 1D 1D AC 2B 7C	...	K.^J)<_Iyy.
00000050	01 01 00 00 00 01 00 00	00 00 00 00 00 00 00 00
00000060	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
00000070	00 00 00 00 00 00 FF FF	FF FF 4D 04 FF FF 00 1Dÿÿÿÿ
00000080	01 04 45 54 68 65 20 54	69 6D 65 73 20 30 33 2FEThe Tim
00000090	4A 61 6E 2F 32 30 30 39	20 43 68 61 6E 63 65 6C	...	Jan/2009 Ch
000000A0	6C 6F 72 20 6F 6E 20 62	72 69 6E 6B 20 6F 66 20	...	lor on brin
000000B0	73 65 63 6F 6E 64 20 62	61 69 6C 6F 75 74 20 66	...	second bail
000000C0	6F 72 20 62 61 6E 6B 73	FF FF FF FF 01 00 F2 05	...	or banksÿÿÿÿ
000000D0	2A 01 00 00 00 43 41 04	67 8A FD B0 FE 55 48 27	...	*....CA.gŠÿ
000000E0	19 67 F1 A6 71 30 B7 10	5C D6 A8 28 E0 39 09 A6gñ q0·.\0"
000000F0	79 62 E0 EA 1F 61 DE B6	49 F6 BC 3F 4C EF 38 C4	...	ybâê.ab¶IÖ¼
00000100	F3 55 04 E5 1E C1 12 DE	5C 38 4D F7 BA 0B 8D 57	...	óU.å.Á.p\8M
00000110	8A 4C 70 2B 6B F1 1D 5F	AC 00 00 00 00	...	ŠLp+kñ._~..
00000000	30 31 30 30 30 30 30 30	36 66 65 32 38 63 30 61	...	01000000006f
00000010	31 61 36 61 32 34 36	b6f1b3726c1
00000020	61 65 36 33 66 37 34 66	39 33 31 65 38 33 36 35	...	ae63f7f4639
00000030	Code Audit	35 61 30 38 39 63	36 38 64 36 31 39 30 30	e15a089c638
00000040	M^O Coordination layer for permissioned institutional actors to generate \$M	30 30 30 30 30 30 30 30	30 30 30 30 30 30 30 30	0000000082c
00000050		31 65 34 62 61 37 34 34	62 62 62 65 36 38 30 65	1e4ba7445bb
00000060		31 66 65 65 31 34 36 37	37 62 61 31 61 33 63 33	1fee1456376
00000070		35 34 30 62 66 37 62 31	63 64 62 36 30 36 65 38	5400bfb1bcd
00000080		35 37 32 33 33 65 30 65	36 31 62 63 36 36 34 39	57233e0e6bc
00000090		66 66 66 66 30 30 31 64	30 31 65 33 36 32 39 39	fff0001d01e
000000A0		30 31 30 31 30 30 30 30	30 30 30 31 30 30 30 30	01010000000
000000B0		30 30 30 30 30 30 30 30	30 30 30 30 30 30 30 30	00000000000
000000C0		30 30 30 30 30 30 30 30	30 30 30 30 30 30 30 30	00000000000
000000D0		30 30 30 30 30 30 30 30	30 30 30 30 30 30 30 30	00000000000
000000E0		30 30 30 30 30 30 30 30	30 30 30 30 66 66 66 66	00000000000
000000F0		66 66 66 66 30 37 30 34	66 66 66 66 30 30 31 64	ffff074ffff
00000100		30 31 30 34 66 66 66 66	66 66 66 66 30 31 30 30	0104ffffffffff
00000110		66 32 30 35 32 61 30 31	30 30 30 30 30 30 34 33	f2052a01000
00000120		34 31 30 34 39 36 62 35	33 38 65 38 35 33 35 31	4104946b538
00000130		39 63 37 32 36 61 32 63	39 31 65 36 31 65 63 31	9c726ac91e6
00000140		31 36 30 30 61 65 31 33	39 30 38 31 33 61 36 32	16000ae1390
00000150		37 63 36 36 66 62 38 62	65 37 39 34 37 62 65 36	7c66fb8be79
00000160		33 63 35 32 64 61 37 35	38 39 33 37 39 35 31 35	3c52da73589
00000170		64 34 65 30 61 36 30 34	66 38 31 34 31 37 38 31	d4e0a646381
00000180		65 36 32 32 39 34 37 32	31 31 36 36 62 66 36 32	e6224947321
00000190		31 65 37 33 61 38 32 63	62 66 32 33 34 32 63 38	1e73a823cbf
000001A0		35 38 65 65 61 63 30 30	30 30 30 30 30 30	58eacc00000
00000000		30 31 30 30 30 30 30 30	34 38 36 30 65 62 31 38	01000000486

Code Maturity Evaluation

Code Maturity Evaluation Guidelines

Category	Evaluation
Access Controls	The use of robust access controls to handle identification and authorization and to ensure safe interactions with the system.
Arithmetic	The proper use of mathematical operations and semantics.
Centralization	The presence of a decentralized governance structure for mitigating insider threats and managing risks posed by contract upgrades
Code Stability	The extent to which the code was altered during the audit.
Upgradeability	The presence of parameterizations of the system that allow modifications after deployment.
Function Composition	The functions are generally small and have clear purposes.
Front-Running	The system's resistance to front-running attacks.
Monitoring	All operations that change the state of the system emit events, making it simple to monitor the state of the system. These events need to be correctly emitted.
Specification	The presence of comprehensive and readable codebase documentation.
Testing and Verification	The presence of robust testing procedures (e.g., unit tests, integration tests, and verification methods) and sufficient test coverage.

Code Maturity Evaluation Results

Category	Evaluation
Access Controls	Satisfactory . All access control is correctly implemented.
Arithmetic	Moderate . Some rounding errors were found.
Centralization	Satisfactory . No significant points of centralization were found.
Code Stability	Satisfactory . The code was stable throughout the audit.
Upgradeability	Satisfactory . The contracts are upgradeable.
Function Composition	Satisfactory . The code was correctly split into helper functions.
Front-Running	Satisfactory . No front-running issues were found.
Monitoring	Satisfactory . Most events are emitted.
Specification	Satisfactory . The code follows the specifications.
Testing and Verification	Satisfactory . The codebase implements unit and fuzz tests.

Findings

3S-M^0-L01

Anyone can stop earning \$M directly through `MToken::stopEarning(address account_)` increasing the `WrappedMToken` balance without backing

Id	3S-M^0-L01
Classification	Low
Severity	Medium
Likelihood	Low
Category	Suggestion
Status	Acknowledged

Description

`MToken` allows an account to stop earning if it is not approved anymore in the `Registrar`. For this reason, when the `WrappedMToken` stops being approved, it must call stop earning in the `MToken` and store the last `index`, such that the correct balances can be calculated. However, `MToken::stopEarning(address account_)` is permissionless and enables sending the `WrappedMToken` address, stopping `MToken` from being accrued. However, as it is not called through `WrappedMToken::disableEarning()`, the `_lastDisableEarningIndex()` will not be immediately recorded, accruing artificial `WrappedMToken` balance that is not actually backed. Thus, users will be able to withdraw more `MToken` than what `WrappedMToken` holds, potentially leading to the last users not being able to withdraw, depending on the index mismatch.

Recommendation

The governance call to stop the `WrappedMToken` from being approved in the `Registrar` must be called atomically with a call to `WrappedMToken::disableEarning()`, such that the last index is correctly recorded.

3S-M^0-L02

`_subtractTotalEarningSupply()` could not be in an unchecked block to prevent rounding errors from wrapping the supply

Id	3S-M^0-L02
Classification	Low
Severity	Low
Likelihood	Low
Category	Suggestion
Status	Addressed in #d6f4404 .

Description

On `WrappedMToken::_addEarningAmount()`, the principal of an account is set as `IndexingMath.getPrincipalAmountRoundedDown(balance_ + amount_, index_)`, but the added principal to `_principalOfTotalEarningSupply` is `IndexingMath.getPrincipalAmountRoundedDown(amount_, currentIndex_)`, indicating that there will be a mismatch between the sum of account principal and `_principalOfTotalEarningSupply`. Additionally, the added amount to `totalEarningSupply()` can further round down in `WrappedMToken::_setTotalEarningSupply()` when calculating the index. In `WrappedMToken::_subtractEarningAmount()`, the same happens as the balance of the account is updated with `balance_ - amount` but the subtract principal is calculated on `amount` only, and the index calculation can again round down. Thus, we can expect the sum of the individual accounts principal and balance to deviate slightly (can be considered dust and dealt with using `excess()`, as long as the balance of `MToken` is bigger).

Recommendation

In `WrappedMToken::_subtractTotalEarningSupply()`, remove the `unchecked` block as the mismatch can cause this operation to wrap around, leading to inflated total supplies.

3S-M^0-N01

Re-enabling, although not possible at the moment, will make a jump in the index, not accompanied by the \$M balance increase

Id	3S-M^0-N01
Classification	None
Category	Suggestion
Status	Acknowledged

Description

Firstly this bug is not possible at the moment because re-enabling is disabled, see **WrappedMToken::enableEarning()**. However, if it is allowed, when it is enabled again, it will use the current index from the **MToken**, which will be bigger than the past stored index in **WrappedMToken** while it was disabled. This will make a jump in everyone's **WrappedMToken** balance, but the actual **MToken** holdings will not match it. Thus, users would get instant profit while the last ones would not be able to withdraw.

Recommendation

A simple fix is not trivial. Perhaps when the new index is fetched, a multiplier can be applied that discounts the previous stored index.

3S-M^0-N02

WrappedMToken::_lastDisableEarningIndex() assumes the index will be in position 1 when disabled, which may not be true in the future

Id	3S-M^0-N02
Classification	None
Category	Suggestion
Status	Acknowledged

Description

WrappedMToken::_lastDisableEarningIndex() fetches index 1 from **_enableDisableEarningIndices** to get the last stored index before disabling. However, this is only true if **WrappedMToken** is disabled once, but if the current code to stop re-enabling earning is removed, it will lead to a wrong index.

Recommendation

The correct index is **_enableDisableEarningIndices.length - 1**. If for gas saving purposes 1 is kept, a comment should be added to ensure it is changed later in case the re-enabling is allowed in the future.