Cyberscope

Audit Report

CRYA Locker

September 2022

Github https://github.com/Eric0718/dao-governance/blob/master/contracts/CryaLock.sol

commit 4b0c52b7d3cc75428b6ed52a999c31b5c3e79098

Audited by © cyberscope

Table of Contents

Contract Review

Github	https://github.com/Eric0718/dao-governance/blob/master/contracts/CryaLock.sol
Commit	4b0c52b7d3cc75428b6ed52a999c31b5c3e79098
Contract Name	CryaLock
Compiler Version	v0.8.11+commit.d7f03943
Optimization	0 runs
Explorer	https://testnet.bscscan.com/token/0xa3bdec85a9b92c1bbe1bd06d863b17ec86781d43

Audit Updates

Initial Audit	19th September 2022
Corrected	

Source Files

Filename	SHA256
@openzeppelin/contracts/governance/utils/IVotes.sol	55fe90680900ea253e4e5b11d9b6ab5c4ff3e85e48ffb94c8b2c29694d01312b
@openzeppelin/contracts/token/ERC20/ERC20.sol	5031430cc2613c32736d598037d3075985a2a09e61592a013dbd09a5bc2041b8
@openzeppelin/contracts/token/ERC20/extensions/draft-ERC20Permit.sol	d070a08919d4a38aa08043c687d1fe1522098b212d2e185aedf2f37275b64087
@openzeppelin/contracts/token/ERC20/extensions/draft-IERC20Permit.sol	3e7aa0e0f69eec8f097ad664d525e7b3f0a3fda8dcdd97de5433ddb131db86ef
@openzeppelin/contracts/token/ERC20/extensions/ERC20Votes.sol	fb449cd9e8ce63e968e8b5c3d39e64f9928a854fcfa4db33d6a853f890e47fd6
@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol	af5c8a77965cc82c33b7ff844deb9826166689e55dc037a7f2f790d057811990
@openzeppelin/contracts/token/ERC20/IERC20.sol	94f23e4af51a18c2269b355b8c7cf4db8003d075c9c541019eb8dcf4122864d5
@openzeppelin/contracts/utils/Context.sol	1458c260d010a08e4c20a4a517882259a23a4baa0b5bd9add9fb6d6a1549814a
@openzeppelin/contracts/utils/Counters.sol	2fdcb1343e5621385b62e57b5c7775607c272122b6f2dc77da8f84828aa40cd0

@openzeppelin/contracts/utils/cryptography/draft-EIP712.sol	fc0e6c5d7184bd03b8deae6ca9a48a1eaaecf9f5e4703611aabfb63401e6d43f
@openzeppelin/contracts/utils/cryptography/ECDSA.sol	4e45d53327d561848fbcf381262ec5c0ac91b2f1f06432210bf76db55279d945
@openzeppelin/contracts/utils/math/Math.sol	929523c09910460ad708c75878d89b9fbed12b65cb5d8b670200c793131072f4
@openzeppelin/contracts/utils/math/SafeCast.sol	e44469cf1affcd59005dc9c69df91af9c7b93e6bc4095148232f86ba9e7f749d
@openzeppelin/contracts/utils/math/SafeMath.sol	0dc33698a1661b22981abad8e5c6f5ebca0dfe5ec14916369a2935d888ff257a
@openzeppelin/contracts/utils/Strings.sol	34127ad0054df5963b0fd694c1b313d17e9114a2f426b85526d6d976210298ab
contracts/CryaLock.sol	54de58e0b6065a4d097850139c13f9e4273c22847309bfadcc2288244f61aa2f
contracts/CryaToken.sol	1ee8c7d2845c0465e4f83fe5fb735f659cd7ba44f7a40b6587b3d3dcad10bbd2

Contract Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	STC	Succeeded Transfer Check	Unresolved
●	BLC	Business Logic Concern	Unresolved
●	ATIB	Address Type Index Boundaries	Unresolved
●	MC	Missing Check	Unresolved
●	L01	Public Function could be Declared External	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L13	Divide before Multiply Operation	Unresolved
●	L14	Uninitialized Variables in Local Scope	Unresolved

STC - Succeeded Transfer Check

Criticality	minor / informative
Location	contract.sol#L170
Status	Unresolved

Description

According to the ERC20 specification, the transfer methods should be checked if the result is successful. Otherwise, the contract may wrongly assume that the transfer has been established.

```
token.transferFrom(admin, to, releaseAmount);
```

Recommendation

The contract should check if the result of the transfer methods is successful.

BLC - Business Logic Concern

Criticality	medium
Location	contract.sol#L127
Status	Unresolved

Description

If the `Ecology` address type is claimed 9 months after the 'releaseStartTime', then the user will not be able to claim the entire amount. For instance, if the user claim the amount 10 years after the initial vest, then the `if(calTime < (startTime + 10 * baseTimeInterval)){` condition will never fulfil and the user will never be able to claim the entire amount.

If the `Team` address type is claimed 59 months after the 'releaseStartTime', then the user will not be able to claim the last proportion.

```
}else if (userType == uint8(AddressType.Ecology)){
    //25% locked release in 9 months
    if(calTime < (startTime + 10 * baseTimeInterval)){
        uint256 lockedBalance = addressInfos[user].totalLocked.mul(25).div(100);
        releaseAmount = lockedBalance.div(9);
    }else{
        //75% locked release in 48  months
        uint256 lockedBalance = addressInfos[user].totalLocked.mul(75).div(100);
        releaseAmount = lockedBalance.div(48);
    }
}
...
}else if (userType == uint8(AddressType.Team)){
    //20% release in a year
    if(addressInfos[user].totalLocked == addressInfos[user].lockedLeft){
        return addressInfos[user].totalLocked.mul(20).div(100);
    }else if (calTime > (updateTime + baseTimeInterval)){
        //80% release in 48 months
        uint256 lockedBalance = addressInfos[user].totalLocked.mul(80).div(100);
        releaseAmount = lockedBalance.div(48);
    }
```

Recommendation

The contract should allow the user to claim the entire amount.

ATIB - Address Type Index Boundaries

Criticality	minor / informative
Location	contract.sol#L77
Status	Unresolved

Description

The parameter _addressTypes is representing an enum AddressType index. Hence, it should be properly checked if it is between the boundaries.

```
function addAddressesBeforeTge(address[] calldata _accounts,uint8[] calldata
_addressTypes,uint256[] calldata _lockBalances)public onlyAdmin{
...
}
```

Recommendation

The contract should check if the ‘_addressTypes’ is between the ‘enum boundaries’ .

MC - Missing Check

Criticality	minor / informative
Location	contract.sol#L45
Status	Unresolved

Description

The method ‘addAddressesBeforeTge’ requires that the ‘tgeTime’ should be greater than ‘block.timestamp’. That means that the constructor should not allow the ‘tgeTime’ to be less than the current ‘block.timestamp’.

```
constructor(uint256 _tgeTime,CryaToken _token){
    tgeTime = _tgeTime;
    admin = msg.sender;
    token = _token;
    tokenTotalSupply = token.totalSupply();
    initDistributionRatio();
}
```

Recommendation

The contract could check if the parameters fulifies the business logic requirements.

L01 - Public Function could be Declared External

Criticality	minor / informative
Location	contracts/CryaLock.sol#L77,95,174,203
Status	Unresolved

Description

Public functions that are never called by the contract should be declared external to save gas.

```
addAddressesBeforeTge
releaseLockedBalance
getLockedBalance
IDOTransfer
```

Recommendation

Use the external attribute for functions never called from the contract.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contracts/CryaLock.sol#L20,77,178,203,36,38
Status	Unresolved

Description

Solidity defines a naming convention that should be followed. Rule exceptions:

- Allow constant variable name/symbol/decimals to be lowercase.
- Allow _ at the beginning of the mixed_case match for private variables and unused parameters.

```
addressInfo
_accounts
_addressTypes
_lockBalances
_addrType
IDOTransfer
baseTimeInterval
IDOSupply
```

Recommendation

Follow the Solidity naming convention.
https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions.

L13 - Divide before Multiply Operation

Criticality	minor / informative
Location	contracts/CryaLock.sol#L106
Status	Unresolved

Description

Performing divisions before multiplications may cause lose of prediction.

```
numbs = (calTime - updateTime).div(baseTimeInterval)
releaseAmount = lockedBalance_scope_2.div(48)
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor / informative
Location	contracts/CryaLock.sol#L112
Status	Unresolved

Description

The are variables that are defined in the local scope and are not initialized.

```
releaseAmount
```

Recommendation

All the local scoped variables should be initialized.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IVotes	Interface			
	getVotes	External		-
	getPastVotes	External		-
	getPastTotalSupply	External		-
	delegates	External		-
	delegate	External	✓	-
	delegateBySig	External	✓	-
ERC20	Implementation	Context, IERC20, IERC20Metadata		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_spendAllowance	Internal	✓	
	_beforeTokenTransfer	Internal	✓	

	_afterTokenTransfer	Internal	✓	
ERC20Permit	Implementation	ERC20, IERC20Permit, EIP712		
	<Constructor>	Public	✓	EIP712
	permit	Public	✓	-
	nonces	Public		-
	DOMAIN_SEPARATOR	External		-
	_useNonce	Internal	✓	
IERC20Permit	Interface			
	permit	External	✓	-
	nonces	External		-
	DOMAIN_SEPARATOR	External		-
ERC20Votes	Implementation	IVotes, ERC20Permit		
	checkpoints	Public		-
	numCheckpoints	Public		-
	delegates	Public		-
	getVotes	Public		-
	getPastVotes	Public		-
	getPastTotalSupply	Public		-
	_checkpointsLookup	Private		
	delegate	Public	✓	-
	delegateBySig	Public	✓	-
	_maxSupply	Internal		
	_mint	Internal	✓	
	_burn	Internal	✓	
	_afterTokenTransfer	Internal	✓	
	_delegate	Internal	✓	
	_moveVotingPower	Private	✓	
	_writeCheckpoint	Private	✓	
	_add	Private		
	_subtract	Private		

IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Counters	Library			
	current	Internal		
	increment	Internal	✓	
	decrement	Internal	✓	
	reset	Internal	✓	
EIP712	Implementation			
	<Constructor>	Public	✓	-
	_domainSeparatorV4	Internal		
	_buildDomainSeparator	Private		
	_hashTypedDataV4	Internal		
ECDSA	Library			
	_throwError	Private		
	tryRecover	Internal		
	recover	Internal		
	tryRecover	Internal		

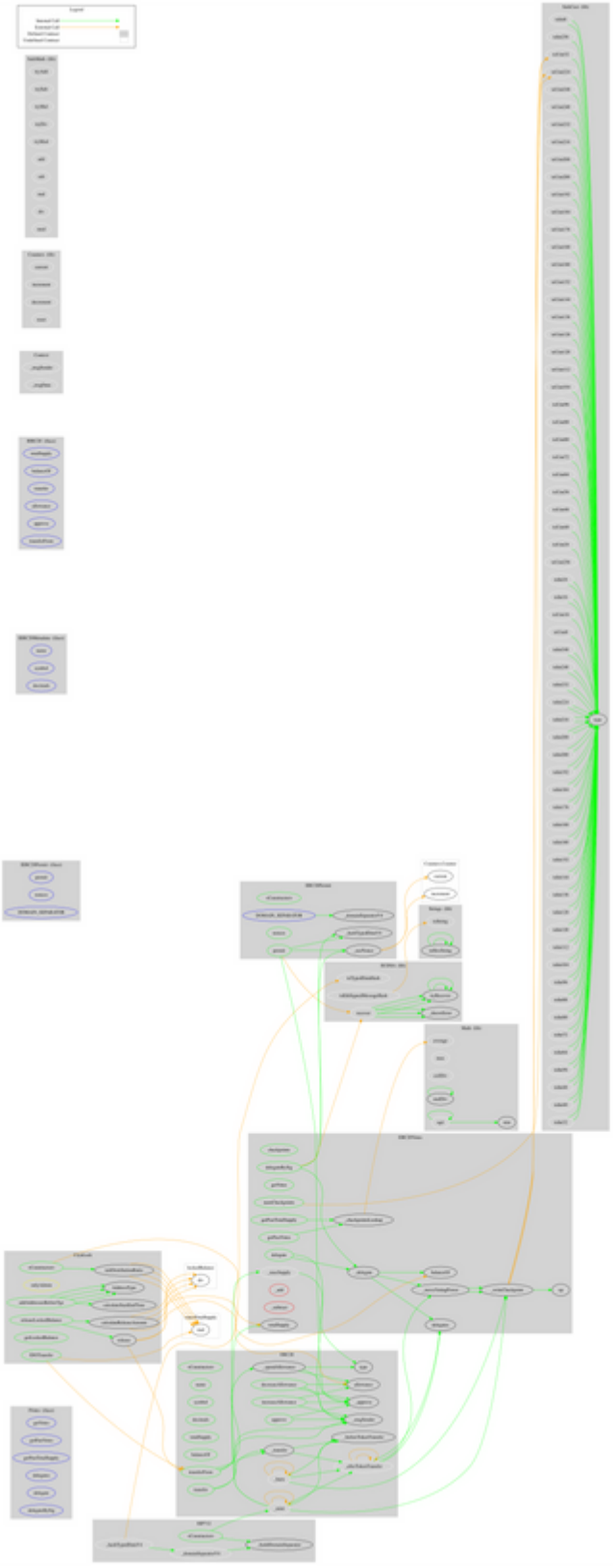
	recover	Internal		
	tryRecover	Internal		
	recover	Internal		
	toEthSignedMessageHash	Internal		
	toEthSignedMessageHash	Internal		
	toTypedDataHash	Internal		
Math	Library			
	max	Internal		
	min	Internal		
	average	Internal		
	ceilDiv	Internal		
	mulDiv	Internal		
	mulDiv	Internal		
	sqrt	Internal		
	sqrt	Internal		
SafeCast	Library			
	toUint248	Internal		
	toUint240	Internal		
	toUint232	Internal		
	toUint224	Internal		
	toUint216	Internal		
	toUint208	Internal		
	toUint200	Internal		
	toUint192	Internal		
	toUint184	Internal		
	toUint176	Internal		
	toUint168	Internal		
	toUint160	Internal		
	toUint152	Internal		
	toUint144	Internal		
	toUint136	Internal		
	toUint128	Internal		
	toUint120	Internal		

	toUint112	Internal		
	toUint104	Internal		
	toUint96	Internal		
	toUint88	Internal		
	toUint80	Internal		
	toUint72	Internal		
	toUint64	Internal		
	toUint56	Internal		
	toUint48	Internal		
	toUint40	Internal		
	toUint32	Internal		
	toUint24	Internal		
	toUint16	Internal		
	toUint8	Internal		
	toUint256	Internal		
	toInt248	Internal		
	toInt240	Internal		
	toInt232	Internal		
	toInt224	Internal		
	toInt216	Internal		
	toInt208	Internal		
	toInt200	Internal		
	toInt192	Internal		
	toInt184	Internal		
	toInt176	Internal		
	toInt168	Internal		
	toInt160	Internal		
	toInt152	Internal		
	toInt144	Internal		
	toInt136	Internal		
	toInt128	Internal		
	toInt120	Internal		
	toInt112	Internal		
	toInt104	Internal		
	toInt96	Internal		

	toInt88	Internal		
	toInt80	Internal		
	toInt72	Internal		
	toInt64	Internal		
	toInt56	Internal		
	toInt48	Internal		
	toInt40	Internal		
	toInt32	Internal		
	toInt24	Internal		
	toInt16	Internal		
	toInt8	Internal		
	toInt256	Internal		
SafeMath	Library			
	tryAdd	Internal		
	trySub	Internal		
	tryMul	Internal		
	tryDiv	Internal		
	tryMod	Internal		
	add	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	mod	Internal		
	sub	Internal		
	div	Internal		
	mod	Internal		
Strings	Library			
	toString	Internal		
	toHexString	Internal		
	toHexString	Internal		
	toHexString	Internal		
CryaLock	Implementation			

	<Constructor>	Public	✓	-
	initDistributionRatio	Private	✓	
	addAddressesBeforeTge	Public	✓	onlyAdmin
	releaseLockedBalance	Public	✓	onlyAdmin
	calculateReleaseAmount	Private	✓	
	release	Private	✓	
	getLockedBalance	Public		-
	calculateStartEndTime	Private		
	IDOTransfer	Public	✓	onlyAdmin
CryaToken	Implementation	ERC20Votes		
	<Constructor>	Public	✓	ERC20 ERC20Permit
	_afterTokenTransfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	

Contract Flow



Summary

The CRYA locker implements a locker for 4 types. Every type has a different set of vesting options. This audit focuses on the business logic concerns and performance optimizations.

Disclaimer

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment.

Cyberscope team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed.

The Cyberscope team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Cyberscope receive a payment to manipulate those results or change the awarding badge that we will be adding in our website.

Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token.

The Cyberscope team disclaims any liability for the resulting losses.

About Cyberscope

Coinscope audit and K.Y.C. service has been rebranded to Cyberscope.

Coinscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Cyberscope and Coinscope are aiming to make crypto discoverable and efficient globally. They provide all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io