

REALITYCARDS SMART CONTRACT AUDIT

March 23, 2021

MixBytes()

CONTENTS

| | |
|---|----|
| 1. INTRODUCTION..... | 1 |
| DISCLAIMER..... | 1 |
| PROJECT OVERVIEW..... | 1 |
| SECURITY ASSESSMENT METHODOLOGY..... | 2 |
| EXECUTIVE SUMMARY..... | 4 |
| PROJECT DASHBOARD..... | 4 |
| 2. FINDINGS REPORT..... | 7 |
| 2.1. CRITICAL..... | 7 |
| 2.2. MAJOR..... | 7 |
| MJR-1 Use msgSender instead of msg.sender in Event param..... | 7 |
| 2.3. WARNING..... | 8 |
| WRN-1 Check that the address is not zero..... | 8 |
| WRN-2 Use general <code>safeTransferFrom</code> | 9 |
| 2.4. COMMENTS..... | 10 |
| CMT-1 Missing the check whether <code>_timestamps</code> has an appropriate length..... | 10 |
| CMT-2 Incorrect function name..... | 11 |
| CMT-3 Difficult calculation of uint max..... | 12 |
| CMT-4 Self-explainable naming..... | 13 |
| CMT-5 Not optimal data type..... | 15 |
| CMT-6 No magic numbers..... | 16 |
| CMT-7 The requirement will never work..... | 17 |
| CMT-8 Save time cache values..... | 18 |
| CMT-9 Explain tricky places..... | 19 |
| CMT-10 One value is always returned..... | 20 |
| CMT-11 Do not hardcode addresses in constructor..... | 21 |
| CMT-12 Use msgSender instead of msg.sender..... | 22 |
| CMT-13 Use SafeMath..... | 23 |
| 3. ABOUT MIXBYTES..... | 24 |

1. INTRODUCTION

1.1 DISCLAIMER

The audit makes no statements or warranties about utility of the code, safety of the code, suitability of the business model, investment advice, endorsement of the platform or its products, regulatory regime for the business model, or any other statements about fitness of the contracts to purpose, or their bug free status. The audit documentation is for discussion purposes only. The information presented in this report is confidential and privileged. If you are reading this report, you agree to keep it confidential, not to copy, disclose or disseminate without the agreement of RealityCards. If you are not the intended recipient(s) of this document, please note that any disclosure, copying or dissemination of its content is strictly forbidden.

1.2 PROJECT OVERVIEW

Reality Cards is the world's first NFT-based prediction market, where instead of betting on an outcome, you own it. Concepts such as shares, bids, asks do not exist- even 'odds' are abstracted away, replaced by a 'daily rental price'.

1.3 SECURITY ASSESSMENT METHODOLOGY

At least 2 auditors are involved in the work on the audit who check the provided source code independently of each other in accordance with the methodology described below:

- 01 "Blind" audit includes:
 - > Manual code study
 - > "Reverse" research and study of the architecture of the code based on the source code only

Stage goal:
Building an independent view of the project's architecture
Finding logical flaws
- 02 Checking the code against the checklist of known vulnerabilities includes:
 - > Manual code check for vulnerabilities from the company's internal checklist
 - > The company's checklist is constantly updated based on the analysis of hacks, research and audit of the clients' code

Stage goal:
Eliminate typical vulnerabilities (e.g. reentrancy, gas limit, flashloan attacks, etc.)
- 03 Checking the logic, architecture of the security model for compliance with the desired model, which includes:
 - > Detailed study of the project documentation
 - > Examining contracts tests
 - > Examining comments in code
 - > Comparison of the desired model obtained during the study with the reversed view obtained during the blind audit

Stage goal:
Detection of inconsistencies with the desired model
- 04 Consolidation of the reports from all auditors into one common interim report document
 - > Cross check: each auditor reviews the reports of the others
 - > Discussion of the found issues by the auditors
 - > Formation of a general (merged) report

Stage goal:
Re-check all the problems for relevance and correctness of the threat level
Provide the client with an interim report
- 05 Bug fixing & re-check.
 - > Client fixes or comments on every issue
 - > Upon completion of the bug fixing, the auditors double-check each fix and set the statuses with a link to the fix

Stage goal:
Preparation of the final code version with all the fixes
- 06 Preparation of the final audit report and delivery to the customer.

Findings discovered during the audit are classified as follows:

FINDINGS SEVERITY BREAKDOWN

| Level | Description | Required action |
|----------|--|---|
| Critical | Bugs leading to assets theft, fund access locking, or any other loss funds to be transferred to any party | Immediate action to fix issue |
| Major | Bugs that can trigger a contract failure. Further recovery is possible only by manual modification of the contract state or replacement. | Implement fix as soon as possible |
| Warning | Bugs that can break the intended contract logic or expose it to DoS attacks | Take into consideration and implement fix in certain period |
| Comment | Other issues and recommendations reported to/acknowledged by the team | Take into consideration |

Based on the feedback received from the Customer's team regarding the list of findings discovered by the Contractor, they are assigned the following statuses:

| Status | Description |
|--------------|---|
| Fixed | Recommended fixes have been made to the project code and no longer affect its security. |
| Acknowledged | The project team is aware of this finding. Recommendations for this finding are planned to be resolved in the future. This finding does not affect the overall safety of the project. |
| No issue | Finding does not affect the overall safety of the project and does not violate the logic of its work. |

1.4 EXECUTIVE SUMMARY

The audited scope implements custom predictions market with a feature of renting tokens to claim reward instead of having them.

The project have several logical modules: proxies to xDai and ETH mainnet, NFT hubs to manage NFT, RCMarket to mange prediction market, RCFactory to create new RCMarkets, RCTreasury to store deposits and manage rewards.

Usage of xDai makes gas very cheap.

Such project could be used to create robust predictions markets.

1.5 PROJECT DASHBOARD

| | |
|-------------------------|--|
| Client | RealityCards |
| Audit name | RealityCards |
| Initial version | 8c0b05b25a7deef25f98532ae2f8afd4f9a84360 |
| Final version | a860b714944341eeda9b26a9e3d1f8f0747b6cbd |
| SLOC | 1457 |
| Date | 2021-02-01 - 2021-03-23 |
| Auditors engaged | 2 auditors |

FILES LISTING

| | |
|------------------------------|---------------------------|
| RCFactory.sol | RCFactory.sol |
| RCMarket.sol | RCMarket.sol |
| RCTreasury.sol | RCTreasury.sol |
| RCNftHubXdai.sol | RCNftHubXdai.sol |
| RCNftHubMainnet.sol | RCNftHubMainnet.sol |
| RCProxyMainnet.sol | RCProxyMainnet.sol |
| RCProxyXdai.sol | RCProxyXdai.sol |
| IAlternateReceiverBridge.sol | IAlternateReceiverBri... |
| IRCPProxyMainnet.sol | IRCPProxyMainnet.sol |
| IRCPProxyXdai.sol | IRCPProxyXdai.sol |
| IRCNftHubXdai.sol | IRCNftHubXdai.sol |
| IRRealitio.sol | IRRealitio.sol |
| IERC20Dai.sol | IERC20Dai.sol |
| IERC721.sol | IERC721.sol |
| IRCMarket.sol | IRCMarket.sol |
| IFactory.sol | IFactory.sol |
| ITreasury.sol | ITreasury.sol |
| IBridgeContract.sol | IBridgeContract.sol |
| RCTreasury.sol | RCTreasury.sol |
| EIP712Base.sol | EIP712Base.sol |
| NativeMetaTransaction.sol | NativeMetaTransaction.sol |
| CloneFactory.sol | CloneFactory.sol |

FINDINGS SUMMARY

| Level | Amount |
|----------|--------|
| Critical | 0 |
| Major | 1 |
| Warning | 2 |
| Comment | 13 |

CONCLUSION

Smart contracts have been audited and several suspicious places have been spotted. During the audit no critical issues were spotted. One issue was marked major as it might cause the undesirable behavior. Several warnings and comments were found and discussed with the client. After working on the reported findings some of them were fixed or acknowledged (if the problem was not critical). So, the contracts are assumed as secure to use according to our security criteria. Final commit identifier with all fixes: `a860b714944341eeda9b26a9e3d1f8f0747b6cbd`

2. FINDINGS REPORT

2.1 CRITICAL

Not Found

2.2 MAJOR

| | |
|----------|--|
| MJR-1 | Use msgSender instead of msg.sender in Event param |
| File | RCMarket.sol NativeMetaTransaction.sol |
| Severity | Major |
| Status | Acknowledged |

DESCRIPTION

Since the contract uses metatransactions everywhere (and uses NativeMetaTransaction), you should always use `msgSender()`

- [RCMarket.sol#L584](#)

otherwise the event parameter maybe not correct

look at the logic at [NativeMetaTransaction.sol#L105](#)

But at [RCMarket.sol#L579](#) the `msgSender()`, so it is used what is not consistent.

RECOMMENDATION

It is recommended to use `msgSender()` in all of `msg.sender` usages (see also: <https://medium.com/biconomy/biconomy-supports-native-meta-transactions-243ce52a2a2b>).

CLIENT'S COMMENTARY

The instances of `msg.sender` left are in functions that are only for the market contract (doesn't use meta-Tx), or the sponsor function where the sponsor is expected to have funds and not use meta-Tx. But we've decided that yes we will do a blanket change to `msgSender()` everywhere, so this will be fixed.

2.3 WARNING

| | |
|-----------------|---|
| WRN-1 | Check that the address is not zero |
| File | RCTreasury.sol RCProxyMainnet.sol RCProxyMainnet.sol RCProxyXdai.sol RCNftHubMainnet.sol |
| Severity | Warning |
| Status | Fixed at https://github.com/RealityCards/RealityCards-Contracts/blob/a860b714944341eeda9b26a9e3d1f8f0747b6cbd |

DESCRIPTION

The following lines use address variables. But if the value turns out to be zero, funds will be lost:

- RCTreasury.sol#L163
- RCProxyMainnet.sol#L70
- RCProxyMainnet.sol#L75
- RCProxyMainnet.sol#L80
- RCProxyMainnet.sol#L85
- RCProxyMainnet.sol#L90
- RCProxyMainnet.sol#L100
- RCProxyMainnet.sol#L104
- RCProxyXdai.sol#L95
- RCProxyXdai.sol#L100
- RCProxyXdai.sol#L105
- RCProxyXdai.sol#L110
- RCProxyXdai.sol#L120
- RCProxyXdai.sol#L142-L149
- RCProxyXdai.sol#L182
- RCNftHubMainnet.sol#L29

RECOMMENDATION

It is recommended to add a check that address is valid.

| | |
|-----------------|---|
| WRN-2 | Use general <code>safeTransferFrom</code> |
| File | <code>RCNftHubXdai.sol</code> |
| Severity | Warning |
| Status | No issue |

DESCRIPTION

It is required to check success of transfer. So it is should be handled as in ERC20:

- `RCNftHubXdai.sol#L69`

RECOMMENDATION

It is recommended to use the `safeTransferFrom()` method from the ERC20 safe library.

CLIENT'S COMMENTARY

The intention is for the market to move the NFTs as and when the highest bidder changes, this means forcefully moving the NFTs without prior approval of the owner, which is why we are calling the internal function `_transfer()` and bypassing the usual ownership checks in `transferFrom()`, because of this it doesn't matter if the owner is a contract not implementing ERC721. If a non-implementer owns it during the event the market will forcefully move the NFT anyway, if they end up being the owner after the event has completed then it's assumed that was the intention of the winning bidder (the non-ERC721 contract creator) and the NFT is now locked. This will not be amended.

2.4 COMMENTS

| | |
|-----------------|---|
| CMT-1 | Missing the check whether <code>_timestamps</code> has an appropriate length |
| File | RCFactory.sol |
| Severity | Comment |
| Status | Fixed at https://github.com/RealityCards/RealityCards-Contracts/blob/a860b714944341eeda9b26a9e3d1f8f0747b6cbd |

DESCRIPTION

At the lines

- RCFactory.sol#L312-L313
- RCFactory.sol#L317
- RCFactory.sol#L320
- RCFactory.sol#L352

have operations with elements of the `_timestamps` array. It is possible that the number of transferred elements of the `_timestamps` array will be less than 3. In this case, a reference will be made to a nonexistent array element.

For clean code, it is better to avoid this situation and check the length of the array.

RECOMMENDATION

It is recommended to check the number of array elements:

```
require(_timestamps.length < 3, "Incorrect number of array elements");
```

| | |
|-----------------|---|
| CMT-2 | Incorrect function name |
| File | RCFactory.sol RCFactory.sol RCTreasury.sol RCProxyMainnet.sol |
| Severity | Comment |
| Status | Fixed at https://github.com/RealityCards/RealityCards-Contracts/blob/a860b714944341eeda9b26a9e3d1f8f0747b6cbd |

DESCRIPTION

In some function `setSomeValue()` the value of boolean variable `someValue` is reversed. But by setting it means setting any value and the value may not even change. It would be more correct to call not "Set", but "Change". This can be seen on the following lines:

- [RCFactory.sol#L187](#)
- [RCFactory.sol#L197](#)
- [RCFactory.sol#L216](#)
- [RCFactory.sol#L226](#)
- [RCFactory.sol#L232](#)
- [RCFactory.sol#L237](#)
- [RCFactory.sol#L242](#)
- [RCTreasury.sol#L125](#)
- [RCTreasury.sol#L130](#)
- [RCProxyMainnet.sol#L137](#)

RECOMMENDATION

It is recommended to rename a `setSomeValue()` function to `changeSomeValue()`.

CLIENT'S COMMENTARY

`setMarketCreationGovernorsOnly` to `changeMarketCreationGovernorsOnly`
`setTrapCardsIfUnapproved` to `changeTrapCardsIfUnapproved`
`addOrRemoveGovernor` to `changeGovernorApproval`
`approveOrUnapproveMarket` to `changeMarketApproval`
`addOrRemoveArtist` to `changeArtistApproval`
`addOrRemoveAffiliate` to `changeAffiliateApproval`
`addOrRemoveCardAffiliate` to `changeCardAffiliateApproval`
`setGlobalPause` to `changeGlobalPause`
`setPauseMarket` to `changePauseMarket`
`enableOrDisableDeposits` to `changeDepositsEnabled`

| | |
|-----------------|---|
| CMT-3 | Difficult calculation of uint max |
| File | RCMarket.sol RCProxyMainnet.sol |
| Severity | Comment |
| Status | Fixed at https://github.com/RealityCards/RealityCards-Contracts/blob/a860b714944341eeda9b26a9e3d1f8f0747b6cbd |

DESCRIPTION

See these lines:

- [RCMarket.sol#L29](#)
- [RCMarket.sol#L30](#)
- [RCProxyMainnet.sol#L91](#)

Both of these values are worked out by strange way. E.g. 2^{256} for uint256 will get 0.

But there are simpler ways to calculate the maximum value. For example:

- `uint256 public constant MAX_UINT256 = uint256(-1);`
- `uint256 public constant MAX_UINT256 = type(uint256).max;`

RECOMMENDATION

It is recommended to make it clearer.

CLIENT'S COMMENTARY

The recommended solution was not available in the project solidity version, after update the recommendations were applied.

| | |
|-----------------|---|
| CMT-4 | Self-explainable naming |
| File | RCFactory.sol RCMarket.sol RCTreasury.sol RCProxyXdai.sol |
| Severity | Comment |
| Status | Fixed at https://github.com/RealityCards/RealityCards-Contracts/blob/a860b714944341eeda9b26a9e3d1f8f0747b6cbd |

DESCRIPTION

It's good if the name of the variable is absolutely self-explainable. For primitive types (integers) it's good to know what exactly the variable is. For mappings it's better to add key to the name (e.g. userDeposits not just deposits)

- key and value of struct is unclear
RCFactory.sol#L38 - mappingOfMarkets
RCMarket.sol#L49 - price
- add Percent postfix
RCFactory.sol#L46 - minimumPriceIncrease
RCMarket.sol#L62 - minimumPriceIncrease
- add DayDivisor postfix
RCMarket.sol#L64 - minRentalDivisor
- add weekDivisor postfix
RCMarket.sol#L66 - hotPotatoDivisor
- rentCollected postfix
RCMarket.sol#L51
RCMarket.sol#L53
RCMarket.sol#L55
- deposit - RCTreasury.sol#L23 what is the key?
- RCProxyXdai.sol#L37 what is the key?
- RCProxyXdai.sol#L38 what is the key?
- RCProxyXdai.sol#L39 - must be upper-cased
- RCProxyXdai.sol#L47 - what is the key?

- [RCProxyXdai.sol#L48](#) - what is the key?
- [RCProxyXdai.sol#L118](#) - change amicable to some common word
- [RCProxyXdai.sol#L50](#) - the purpose of the value is not clear from the name

RECOMMENDATION

It is recommended to rename variables.

CLIENT'S COMMENTARY

mappingOfMarkets not changed, variable unused for now, a better name will be chosen if it's used otherwise it will be removed.

price to tokenPrice

minimumPriceIncrease to minimumPriceIncreasePercent

minRentalDivisor to minRentalDayDivisor

hotPotatoDivisor to hotPotatoWeekDivisor

collectedPerUser to rentCollectedPerUser

collectedPerToken to rentCollectedPerToken

totalCollected to totalRentCollected

deposit to userDeposit

isMarket not changed, where used it offers a simple readable name, also used in external bot

upgradedNfts to upgradedNftId

nft to NFT

deposits not changed, used in external bot

hasConfirmedDeposit not changed, used in external bot

setAmicableResolution not changed

floatSize, not changed, it is the size of the float.

| | |
|-----------------|---|
| CMT-5 | Not optimal data type |
| File | RCFactory.sol RCMarket.sol RCNftHubXdai.sol |
| Severity | Comment |
| Status | Fixed at https://github.com/RealityCards/RealityCards-Contracts/blob/a860b714944341eeda9b26a9e3d1f8f0747b6cbd |

DESCRIPTION

At the line `RCFactory.sol#L42` uses an array. But with the use of structure, the code will become clearer.

The following lines use variables and numbers:

- `RCMarket.sol#L34`
- `RCNftHubXdai.sol#L80`
- `RCNftHubXdai.sol#L87`

This makes the code hard to read.

RECOMMENDATION

It is recommended to make a structure.

It is recommended to create constants or enum:

```
MODE_CLASSIC
MODE_WINNER_TAKES_ALL
MODE_HOT_POTATO
```

CLIENT'S COMMENTARY

Not changed, using an array offers easier integration with certain external services that already have an array as a data type.

6. Named constants or enum

Mode changed to enum

Market state checks have been updated.

| | |
|-----------------|---|
| CMT-6 | No magic numbers |
| File | RCMarket.sol RCProxyMainnet.sol RCProxyXdai.sol |
| Severity | Comment |
| Status | Fixed at https://github.com/RealityCards/RealityCards-Contracts/blob/a860b714944341eeda9b26a9e3d1f8f0747b6cbd |

DESCRIPTION

- RCMarket.sol#L487 (fixed)
- RCMarket.sol#L675
- RCMarket.sol#L687
- RCMarket.sol#L721 (fixed)
- RCProxyMainnet.sol#L117 (what is 2 and 0)
- RCProxyMainnet.sol#L149 (what is 2 and 0)
- RCProxyMainnet.sol#L169 (what is 400000) (fixed)
- RCProxyXdai.sol#L173 (what is 200000) (fixed)
- RCProxyXdai.sol#L206 (what is 200000) (fixed)

RECOMMENDATION

It is recommended to create named constants with required explanation about choicing the value

CLIENT'S COMMENTARY

Changed to MIN_RENTAL_VALUE

100 is not considered a magic number here because alternatives such as "HUNDRED", "PERCENT" or "CENTUM" wouldn't clarify the basic arithmetic formula being performed.

As above

Issue derived from 6.a.

Updated to REALITIO_TEMPLATE_ID and REALITIO_NONCE

As above

Updated to XDAI_BRIDGE_GAS_COST

Updated to MAINNET_BRIDGE_GAS_COST

As above

| | |
|-----------------|--|
| CMT-7 | The requirement will never work |
| File | RCMarket.sol |
| Severity | Comment |
| Status | Fixed at https://github.com/RealityCards/RealityCards-Contracts/blob/a860b714944341eeda9b26a9e3d1f8f0747b6cbd |

DESCRIPTION

At the line `RCMarket.sol#L684` has the value for the requirement always False.

RECOMMENDATION

It is recommended to make a variable or condition instead of False. Or remove this requirement.

CLIENT'S COMMENTARY

Require statement amended and moved to a more appropriate place

| | |
|-----------------|--------------------------------|
| CMT-8 | Save time cache values |
| File | RCMarket.sol RCTreasury.sol |
| Severity | Comment |
| Status | No issue |

DESCRIPTION

At the lines:

- RCMarket.sol#L774-L777
- RCTreasury.sol#L176-L186

`msgSender()` is used in a lot of places. It is better to cache it to avoid multi calls.

RECOMMENDATION

It is recommended to cache the value.

CLIENT'S COMMENTARY

`msgSender()` is now cached in several functions, although minimal benefit to be had when compiling with the optimizer.

| | |
|-----------------|---|
| CMT-9 | Explain tricky places |
| File | RCTreasury.sol RCProxyXdai.sol |
| Severity | Comment |
| Status | Fixed at https://github.com/RealityCards/RealityCards-Contracts/blob/a860b714944341eeda9b26a9e3d1f8f0747b6cbd |

DESCRIPTION

Let's take a look at the following lines:

- [RCTreasury.sol#L73](#)
- [RCProxyXdai.sol#L215](#)

It is not clear why $24 \times 6 = 10$ minutes.

It is not clear floatSize.

RECOMMENDATION

It is recommended to add explanations as comments.

CLIENT'S COMMENTARY

Added explanation about min rental divisor
floatSize, not changed, it is the size of the float

| | |
|-----------------|---|
| CMT-10 | One value is always returned |
| File | RCTreasury.sol RCProxyXdai.sol |
| Severity | Comment |
| Status | Fixed at https://github.com/RealityCards/RealityCards-Contracts/blob/a860b714944341eeda9b26a9e3d1f8f0747b6cbd |

DESCRIPTION

Here the function always returns True and never returns False:

- [RCTreasury.sol#L100](#)
- [RCProxyXdai.sol#L85](#)

RECOMMENDATION

It is recommended to remove the return statement.

CLIENT'S COMMENTARY

Removed the return value

| | |
|-----------------|--|
| CMT-11 | Do not hardcode addresses in constructor |
| File | RCProxyMainnet.sol |
| Severity | Comment |
| Status | Fixed at https://github.com/RealityCards/RealityCards-Contracts/blob/a860b714944341eeda9b26a9e3d1f8f0747b6cbd |

DESCRIPTION

The address could be changed if deployed to testnet for example [RCProxyMainnet.sol#L52](#)

RECOMMENDATION

It is recommended to set the address as an argument.

CLIENT'S COMMENTARY

Address is now passed into the constructor

| | |
|-----------------|--|
| CMT-12 | Use msgSender instead of msg.sender |
| File | RCMarket.sol RCTreasury.sol NativeMetaTransaction.sol |
| Severity | Comment |
| Status | Fixed at https://github.com/RealityCards/RealityCards-Contracts/blob/a860b714944341eeda9b26a9e3d1f8f0747b6cbd |

DESCRIPTION

Since the contract uses metatransactions everywhere (and uses NativeMetaTransaction), you should always use msgSender

- [RCMarket.sol#L332](#)
- [RCTreasury.sol#L200](#)
- [RCTreasury.sol#L211](#)

look at the logic at [NativeMetaTransaction.sol#L105](#)

In above examples the code is correct but not robust, the method cannot be called via metaTransaction by the market contract. If the part of logic will be inaccurate copy-pasted to some other place it's easy to make a mistake forgetting about switching to `msgSender()`.

RECOMMENDATION

It is recommended to use `msgSender()` in all of `msg.sender` usages (see also: <https://medium.com/biconomy/biconomy-supports-native-meta-transactions-243ce52a2a2b>).

CLIENT'S COMMENTARY

Changed all occurrences of msg.sender to msgSender.

| | |
|-----------------|--|
| CMT-13 | Use SafeMath |
| File | RCTreasury.sol |
| Severity | Comment |
| Status | Fixed at https://github.com/RealityCards/RealityCards-Contracts/blob/a860b714944341eeda9b26a9e3d1f8f0747b6cbd |

DESCRIPTION

The uint256 overflow is possible at [RCTreasury.sol#L84](#)

RECOMMENDATION

It is recommended to use SafeMath.

3. ABOUT MIXBYTES

MixBytes is a team of blockchain developers, auditors and analysts keen on decentralized systems. We build open-source solutions, smart contracts and blockchain protocols, perform security audits, work on benchmarking and software testing solutions, do research and tech consultancy.

BLOCKCHAINS



Ethereum



Cosmos



EOS



Substrate

TECH STACK



Python



Solidity



Rust



C++

CONTACTS



https://github.com/mixbytes/audits_public



<https://mixbytes.io/>



hello@mixbytes.io



<https://t.me/MixBytes>



<https://twitter.com/mixbytes>