



# Cyberscope

## Audit Report

# Crypto 4 A Cause Fund

August 2022

Type ERC20

Network MATIC

Address 0x8fd0195469b51a935dc3c48617ced6b400e38c9c

Audited by © cyberscope

# Table of Contents

<b>Table of Contents</b>	<b>1</b>
<b>Contract Review</b>	<b>3</b>
<b>Audit Updates</b>	<b>3</b>
<b>Source Files</b>	<b>4</b>
<b>Contract Analysis</b>	<b>9</b>
<b>ST - Stop Transactions</b>	<b>10</b>
Description	10
Recommendation	10
<b>MT - Mint Tokens</b>	<b>11</b>
Description	11
Recommendation	11
<b>Contract Diagnostics</b>	<b>12</b>
<b>MC - Missing Check</b>	<b>13</b>
Description	13
Recommendation	13
<b>SPR - Sale Price Rate</b>	<b>14</b>
Description	14
Recommendation	14
<b>EVS - External Value Sanitization</b>	<b>15</b>
Description	15
Recommendation	15
<b>L04 - Conformance to Solidity Naming Conventions</b>	<b>16</b>
Description	16
Recommendation	16
<b>L07 - Missing Events Arithmetic</b>	<b>17</b>
Description	17

<b>Recommendation</b>	<b>17</b>
<b>L09 - Dead Code Elimination</b>	<b>17</b>
<b>Description</b>	<b>17</b>
<b>Recommendation</b>	<b>17</b>
<b>L15 - Local Scope Variable Shadowing</b>	<b>19</b>
<b>Description</b>	<b>19</b>
<b>Recommendation</b>	<b>19</b>
<b>Contract Functions</b>	<b>20</b>
<b>Contract Flow</b>	<b>32</b>
<b>Domain Info</b>	<b>33</b>
<b>Summary</b>	<b>34</b>
<b>Disclaimer</b>	<b>35</b>
<b>About Cyberscope</b>	<b>36</b>

## Contract Review

<b>Contract Name</b>	TokenERC20
<b>Compiler Version</b>	v0.8.12+commit.f00d7308
<b>Optimization</b>	800 runs
<b>Licence</b>	
<b>Explorer</b>	<a href="https://polygonscan.com/token/0x8fd0195469b51a935dc3c48617ced6b400e38c9c">https://polygonscan.com/token/0x8fd0195469b51a935dc3c48617ced6b400e38c9c</a>
<b>Symbol</b>	C4C
<b>Decimals</b>	18
<b>Total Supply</b>	1,000,000,000
<b>Domain</b>	crypto4ac.com

## Audit Updates

<b>Initial Audit</b>	2nd August 2022
<b>Corrected</b>	

## Source Files

Filename	SHA256
@openzeppelin/contracts-upgradeable/access/AccessControlEnumerableUpgradeable.sol	a55a53b215e2bb9c350bf7b86ee09b0e488522f7d8747877fd9a3a7e474c2c26
@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol	7f221363f6bd6fcf5af3f5e6ae628756c195021c3b366718665427c4f14099cb
@openzeppelin/contracts-upgradeable/access/IAccessControlEnumerableUpgradeable.sol	00e174801c04f08f2840ee1eed6394d06ba2b029c0b6078166255148794c1187
@openzeppelin/contracts-upgradeable/access/IAccessControlUpgradeable.sol	6d3fbd4566bc123db1ee6ba2a1b79544b572df9b9cc9be360ddb3244dd07c86b
@openzeppelin/contracts-upgradeable/governance/utils/IVotesUpgradeable.sol	400936c02700eb4147c65a91a15fb6f90d074d7519f8ebce49dce78a2c917186
@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol	6e058aaee8c641107b209b62c34d484f2f125a44ecb66f7204a701614dfc1d68

<b>@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol</b>	8aecaaba0f09bc906c27867246210adfd19230a3e4a209a1909045c633030476
<b>@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol</b>	b6adbe9bc075b15cfb4b90f1ae020da4c78e3feada056a4c75b875350285c915
<b>@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol</b>	a439a162881f7f36131b1fe307aa2a8dc98ac3f01ac121ff92fbbc25d0d216b5
<b>@openzeppelin/contracts-upgradeable/token/ERC20/extensions/draft-ERC20PermitUpgradeable.sol</b>	6409d907153066d7af6cb38d7a3ec2eaaf57caa7b8b355228a2c7649d7099168
<b>@openzeppelin/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol</b>	b97515a88e75c313eacf0a27c9439ef371d86d4c2730d3b13076640942f813df
<b>@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC20BurnableUpgradeable.sol</b>	ca660e828b0c4be205a9f56f3b87b91c1fa67cfd0f6e9dbd431faea7a6280d36
<b>@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC20PausableUpgradeable.s</b>	4be8fb2dba4cfb6282d9c311185ce1c854175e5f9e8321bde52689dea732a8d9

ol	
@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC20VoteSUpgradeable.sol	d1016ca29e15b3b91c5ccc2d4afdd7064a0c6e2839b2e6160e3a7f2ce95057b7
@openzeppelin/contracts-upgradeable/token/ERC20/extensions/IERC20MetadataUpgradeable.sol	68bcca423fc72ec9625e219c9e36306c726a347e43f3711467c579bd3f6500c8
@openzeppelin/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol	db1d80b38061ba675444e6ad861a621d99666042950278d6cdeae9a108afdd17
@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol	44edc4d7099c781d11421cea2d82a52948e738f5f6191c8ad01dfc0f9858549c
@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol	5fb301961e45cb482fe4e05646d2f529aa449fe0e90c6671475d6a32356fa2d4
@openzeppelin/contracts-upgradeable/utils/CountersUpgradeable.sol	5c1ac829a429b0c2ca9b4c9ed8b78d412320e9175e45f088c4e9056ef95fbf21
@openzeppelin/contracts-upgradeable/utils/cryptography/draft-EIP712Upgradeable.sol	9dd13a59c80288b44db61f9eaca6704fae90e79808c2669ad1bf41aefeef3f29

<b>@openzeppelin/contracts-upgradeable/Utils/cryptography/ECDSAUpgradeable.sol</b>	22ee481b20f289ce83a466bffd66ade2dfb47a23307179b254fed5756b3ee2cf
<b>@openzeppelin/contracts-upgradeable/Utils/introspection/ERC165Upgradeable.sol</b>	fd84e5284eccc479268f0ef36b830019d4f7999ceb7959430d8d8d9e602dd4ef
<b>@openzeppelin/contracts-upgradeable/Utils/introspection/IERC165Upgradeable.sol</b>	a39bc026ad6214e9ecd526bd4a1ddf9862d80bd4a9d0d031d9bafa4c3c147c0b
<b>@openzeppelin/contracts-upgradeable/Utils/math/MathUpgradeable.sol</b>	404840654f775c8dd015de4bb15d2bcabb93974cb4e2729397587a9090df788a
<b>@openzeppelin/contracts-upgradeable/Utils/math/SafeCastUpgradeable.sol</b>	dd20bf714af3411164fad48402c99fc2a0b64c323ff63d5b8f6b72eeb26c9525
<b>@openzeppelin/contracts-upgradeable/Utils/MulticallUpgradeable.sol</b>	33d0a6636b6ed6b75ebf3ab474f79c012ea23f0291dcbae748164fd515bc4e36
<b>@openzeppelin/contracts-upgradeable/Utils/StringsUpgradeable.sol</b>	16a0e36f8dc6a83df3fec4344a11ad166ba99649d1cc52613c7ebe8015bd81a3
<b>@openzeppelin/contracts-upgradeable/Utils/structs/EnumerableSetUpgradeable</b>	80cae696855012fa154908e5641f81c5d94ac3bf5ecd463c62fdafc120b9bc9e



<b>ble.sol</b>	
<b>contracts/interfaces/IThirdwebContract.sol</b>	8fc9d29ddee99b052ccdc521c272ee4df8a7de0e1754bfcba397dc5cdfa18c72
<b>contracts/interfaces/IThirdwebPlatformFee.sol</b>	f3d7fb410d1d7d68e024460fec65ea2199a5684ed171b308696b2e70c41d5c65
<b>contracts/interfaces/IThirdwebPrimarySale.sol</b>	78d189e4e669b38d60c15877ef5f24b0e7bad4be6f0e411ad840336d47c084fe
<b>contracts/interfaces/ITWFee.sol</b>	4c57ef2e5572551ee29ec7ecfcb67932f152f7b0ffd1e5c84e0976f577eb43c5
<b>contracts/interfaces/IWETH.sol</b>	09e1104223d0b83a346c98102eafec96916c44f53c8c3eef13e1806149943bfb
<b>contracts/interfaces/token/ITokenERC20.sol</b>	1aa729594efce9d39beb832784f98172bb3a47959d4b997cb265ce4b56277338
<b>contracts/lib/CurrencyTransferLib.sol</b>	edb795a92aafc22c3154c8fdaa696315b33ec86b68280a73d1b8c9914f6d2638
<b>contracts/lib/FeeType.sol</b>	3d2ede585eb7e37872a0f3566a143f5b2aa586873160966d34c98963015f622d
<b>contracts/openzeppelin-presets/meta-tx/ERC2771ContextUpgradeable.sol</b>	4ef0ce1601048c10a4b0fdc3247062be8f1a9ca0441c862ddfadc16251a31edb
<b>contracts/token/TokenERC20.sol</b>	41f12c3f3665abbc3f9653bd853f1511074cd63eeae859cdd6f14e7619fbb54b

# Contract Analysis

● Critical   ● Medium   ● Minor   ● Pass

Severity	Code	Description
●	ST	Contract Owner is not able to stop or pause transactions
●	OCTD	Contract Owner is not able to transfer tokens from specific address
●	OTUT	Owner Transfer User's Tokens
●	ELFM	Contract Owner is not able to increase fees more than a reasonable percent (25%)
●	ULTW	Contract Owner is not able to increase the amount of liquidity taken by dev wallet more than a reasonable percent
●	MT	Contract Owner is not able to mint new tokens
●	BT	Contract Owner is not able to burn tokens from specific wallet
●	BC	Contract Owner is not able to blacklist wallets from selling

## ST - Stop Transactions

<b>Criticality</b>	medium
<b>Location</b>	contract.sol#L134

### Description

The 'admin' role has the authority to stop the transactions for everyone except the 'transfer' role. The 'admin' role may take advantage of this by setting any address except zero to the 'transfer' role.

```
if (!hasRole(TRANSFER_ROLE, address(0)) && from != address(0) && to !=  
address(0)) {  
    require(hasRole(TRANSFER_ROLE, from) || hasRole(TRANSFER_ROLE, to),  
"transfers restricted.");  
}
```

### Recommendation

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

## MT - Mint Tokens

<b>Criticality</b>	critical
<b>Location</b>	contract.sol#L162,168

### Description

The 'minter' role has the authority to mint tokens. The 'minter' role may take advantage of it by calling the `mintTo` function. If this method is abused, then the contract tokens will be highly inflated.

```
function mintTo(address to, uint256 amount) public virtual {  
    require(hasRole(MINTER_ROLE, _msgSender()), "not minter.");  
    _mintTo(to, amount);  
}
```

The 'minter' role can also mint tokens by using an off-chain signed message. The 'minter' role may take advantage of it by calling the 'mintWithSignature' function providing a signed message. The message contains information like the amount of tokens that will be minted and the recipient address.

```
function mintWithSignature(MintRequest calldata _req, bytes calldata _signature)  
external payable nonReentrant {
```

### Recommendation

The owner should carefully manage the credentials of the owner's account. We advised considering an extra-strong security mechanism that the actions may be quarantined by many users instead of one. The owner could also renounce the contract ownership for a period of time or pass the access to the zero address.

# Contract Diagnostics

● Critical    ● Medium    ● Minor

Severity	Code	Description
●	MC	Missing Check
●	SPR	Sale Price Rate
●	EVS	External Value Sanitization
●	L04	Conformance to Solidity Naming Conventions
●	L07	Missing Events Arithmetic
●	L09	Dead Code Elimination
●	L15	Local Scope Variable Shadowing

## MC - Missing Check

<b>Criticality</b>	minor
<b>Location</b>	contract.sol#L226

### Description

The contract is processing variables that have not properly sanitized and checked that they form the proper shape. These variables may produce vulnerability issues.

If the sum of platformFees and twFee are greater than the price, the contract will underflow.

```
CurrencyTransferLib.transferCurrency(  
    _currency,  
    _msgSender(),  
    _primarySaleRecipient,  
    _price - platformFees - twFee  
);
```

### Recommendation

The contract should properly check the variables according to the required specifications.

## SPR - Sale Price Rate

<b>Criticality</b>	minor
<b>Location</b>	contract.sol#L175

### Description

According to the `mintWithSignature` method, the 'minter' role can mint tokens according to the signature. The signature contains the price and the address of funds that will be deposited in order to mint tokens. There is no on-chain connection between price in relation to the quantity of tokens that will be minted. The contract assumes that the off-chain mechanism sets the correct price per token.

```
collectPrice(saleRecipient, _req.currency, _req.price);  
  
_mintTo(receiver, _req.quantity);
```

### Recommendation

The contract could incarnate a more transparent layer of on-chain price rate. A suggested implementation could use a price oracle mechanism.

## EVS - External Value Sanitization

<b>Criticality</b>	minor
<b>Location</b>	contract.sol#L217

### Description

During the funds distribution phase in the 'mintWithSignature' method, the contract is using an external source in order to determine the 'thirdweb' fee. Since the 'thirdweb' is operating as an external source, the returned values should be sanitised.

```
(address twFeeRecipient, uint256 twFeeBps) =  
thirdwebFee.getFeeInfo(address(this), FeeType.PRIMARY_SALE);  
uint256 twFee = (_price * twFeeBps) / MAX_BPS;
```

### Recommendation

The contract could embody checks that guarantee the proper execution of the contract. The 'twFeeBps' could be less than 'MAX\_BPS' or limit up to a specific value.



## L04 - Conformance to Solidity Naming Conventions

<b>Criticality</b>	minor
<b>Location</b>	contracts/lib/CurrencyTransferLib.sol#L14,15,16,17,32,33,34,35,36,61,62,63,64,79,80,81,82,109  contracts/token/TokenERC20.sol#L83,84,85,86,87,88,89,90,162,168,183,189,208,209,210,241,256,305

### 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.

```
_uri  
_signature  
_req  
_price  
_currency  
_primarySaleRecipient  
_platformFeeBps  
_platformFeeRecipient  
_saleRecipient  
...
```

### Recommendation

Follow the Solidity naming convention.

<https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions>.

## L07 - Missing Events Arithmetic

**Criticality**

minor

**Location**

contracts/token/TokenERC20.sol#L82

### Description

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
platformFeeBps = uint128(_platformFeeBps)
```

### Recommendation

Emit an event for critical parameter changes.

## L09 - Dead Code Elimination

**Criticality**

minor

**Location**

contracts/lib/CurrencyTransferLib.sol#L78,106,31

### Description

Functions that are not used in the contract, and make the code's size bigger.

```
transferCurrencyWithWrapperAndBalanceCheck  
safeTransferNativeTokenWithWrapper  
safeTransferERC20WithBalanceCheck
```

### Recommendation

Remove unused functions.



## L15 - Local Scope Variable Shadowing

**Criticality**

minor

**Location**

contracts/token/TokenERC20.sol#L84,85

### Description

There are variables that are defined in the local scope containing the same name from an upper scope.

```
_symbol  
_name
```

### Recommendation

The local variables should have different names from the upper scoped variables.

# Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
<b>AccessControlEnumerableUpgradeable</b>	Implementation	Initializable, IAccessControlEnumerableUpgradeable, AccessControlUpgradeable		
	__AccessControlEnumerable_init	Internal	✓	onlyInitializing
	__AccessControlEnumerable_init_unchained	Internal	✓	onlyInitializing
	supportsInterface	Public		-
	getRoleMember	Public		-
	getRoleMemberCount	Public		-
	_grantRole	Internal	✓	
	_revokeRole	Internal	✓	
<b>AccessControlUpgradeable</b>	Implementation	Initializable, ContextUpgradeable, IAccessControlUpgradeable, ERC165Upgradeable		
	__AccessControl_init	Internal	✓	onlyInitializing
	__AccessControl_init_unchained	Internal	✓	onlyInitializing
	supportsInterface	Public		-
	hasRole	Public		-
	_checkRole	Internal		
	getRoleAdmin	Public		-
	grantRole	Public	✓	onlyRole
	revokeRole	Public	✓	onlyRole
	renounceRole	Public	✓	-

	_setupRole	Internal	✓	
	_setRoleAdmin	Internal	✓	
	_grantRole	Internal	✓	
	_revokeRole	Internal	✓	
<b>IAccessControlEnumerableUpgradeable</b>	Interface	IAccessControlUpgradeable		
	getRoleMember	External		-
	getRoleMemberCount	External		-
<b>IAccessControlUpgradeable</b>	Interface			
	hasRole	External		-
	getRoleAdmin	External		-
	grantRole	External	✓	-
	revokeRole	External	✓	-
	renounceRole	External	✓	-
<b>IVotesUpgradeable</b>	Interface			
	getVotes	External		-
	getPastVotes	External		-
	getPastTotalSupply	External		-
	delegates	External		-
	delegate	External	✓	-
	delegateBySig	External	✓	-
<b>Initializable</b>	Implementation			
	_isConstructor	Private		
<b>PausableUpgradeable</b>	Implementation	Initializable, ContextUpgradeable		
	__Pausable_init	Internal	✓	onlyInitializing
	__Pausable_init_unchained	Internal	✓	onlyInitializing
	paused	Public		-

	_pause	Internal	✓	whenNotPaused
	_unpause	Internal	✓	whenPaused
<b>ReentrancyGuardUpgradeable</b>	Implementation	Initializable		
	__ReentrancyGuard_init	Internal	✓	onlyInitializing
	__ReentrancyGuard_init_unchained	Internal	✓	onlyInitializing
<b>ERC20Upgradeable</b>	Implementation	Initializable, ContextUpgradeable, IERC20Upgradeable, IERC20MetadataUpgradeable		
	__ERC20_init	Internal	✓	onlyInitializing
	__ERC20_init_unchained	Internal	✓	onlyInitializing
	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	✓	

<b>ERC20PermitUpgradable</b>	Implementation	Initializable, ERC20Upgradable, IERC20PermitUpgradable, EIP712Upgradable		
	__ERC20Permit_init	Internal	✓	onlyInitializing
	__ERC20Permit_init_unchained	Internal	✓	onlyInitializing
	permit	Public	✓	-
	nonces	Public		-
	DOMAIN_SEPARATOR	External		-
	_useNonce	Internal	✓	
<b>IERC20PermitUpgradeable</b>	Interface			
	permit	External	✓	-
	nonces	External		-
	DOMAIN_SEPARATOR	External		-
<b>ERC20BurnableUpgradeable</b>	Implementation	Initializable, ContextUpgradeable, ERC20Upgradable		
	__ERC20Burnable_init	Internal	✓	onlyInitializing
	__ERC20Burnable_init_unchained	Internal	✓	onlyInitializing
	burn	Public	✓	-
	burnFrom	Public	✓	-
<b>ERC20PausableUpgradeable</b>	Implementation	Initializable, ERC20Upgradable, PausableUpgradeable		
	__ERC20Pausable_init	Internal	✓	onlyInitializing
	__ERC20Pausable_init_unchained	Internal	✓	onlyInitializing
	_beforeTokenTransfer	Internal	✓	



<b>ERC20VotesUp gradeable</b>	Implementation	Initializable, IVotesUpgr adeable, ERC20Perm itUpgradeab le		
	__ERC20Votes_init	Internal	✓	onlyInitializing
	__ERC20Votes_init_unchained	Internal	✓	onlyInitializing
	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		
<b>IERC20Metada taUpgradeable</b>	Interface	IERC20Upgr adeable		
	name	External		-
	symbol	External		-
	decimals	External		-
<b>IERC20Upgrad eable</b>	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-

	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
<b>AddressUpgradable</b>	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	verifyCallResult	Internal		
<b>ContextUpgradable</b>	Implementation	Initializable		
	__Context_init	Internal	✓	onlyInitializing
	__Context_init_unchained	Internal	✓	onlyInitializing
	_msgSender	Internal		
	_msgData	Internal		
<b>CountersUpgradable</b>	Library			
	current	Internal		
	increment	Internal	✓	
	decrement	Internal	✓	
	reset	Internal	✓	
<b>EIP712Upgradable</b>	Implementation	Initializable		
	__EIP712_init	Internal	✓	onlyInitializing
	__EIP712_init_unchained	Internal	✓	onlyInitializing
	_domainSeparatorV4	Internal		
	_buildDomainSeparator	Private		
	_hashTypedDataV4	Internal		

	_EIP712NameHash	Internal		
	_EIP712VersionHash	Internal		
<b>ECDSAUpgradeable</b>	Library			
	_throwError	Private		
	tryRecover	Internal		
	recover	Internal		
	tryRecover	Internal		
	recover	Internal		
	tryRecover	Internal		
	recover	Internal		
	toEthSignedMessageHash	Internal		
	toEthSignedMessageHash	Internal		
	toTypedDataHash	Internal		
<b>ERC165Upgradeable</b>	Implementation	Initializable, IERC165Upgradeable		
	__ERC165_init	Internal	✓	onlyInitializing
	__ERC165_init_unchained	Internal	✓	onlyInitializing
	supportsInterface	Public		-
<b>IERC165Upgradeable</b>	Interface			
	supportsInterface	External		-
<b>MathUpgradeable</b>	Library			
	max	Internal		
	min	Internal		
	average	Internal		
	ceilDiv	Internal		
<b>SafeCastUpgradeable</b>	Library			
	toUint224	Internal		

	toUint128	Internal		
	toUint96	Internal		
	toUint64	Internal		
	toUint32	Internal		
	toUint16	Internal		
	toUint8	Internal		
	toUint256	Internal		
	toInt128	Internal		
	toInt64	Internal		
	toInt32	Internal		
	toInt16	Internal		
	toInt8	Internal		
	toInt256	Internal		
<b>MulticallUpgradable</b>	Implementation	Initializable		
	__Multicall_init	Internal	✓	onlyInitializing
	__Multicall_init_unchained	Internal	✓	onlyInitializing
	multicall	External	✓	-
	_functionDelegateCall	Private	✓	
<b>StringsUpgradable</b>	Library			
	toString	Internal		
	toHexString	Internal		
	toHexString	Internal		
<b>EnumerableSetUpgradeable</b>	Library			
	_add	Private	✓	
	_remove	Private	✓	
	_contains	Private		
	_length	Private		
	_at	Private		
	_values	Private		
	add	Internal	✓	

	remove	Internal	✓	
	contains	Internal		
	length	Internal		
	at	Internal		
	values	Internal		
	add	Internal	✓	
	remove	Internal	✓	
	contains	Internal		
	length	Internal		
	at	Internal		
	values	Internal		
	add	Internal	✓	
	remove	Internal	✓	
	contains	Internal		
	length	Internal		
	at	Internal		
	values	Internal		
<b>IThirdwebContract</b>	Interface			
	contractType	External		-
	contractVersion	External		-
	contractURI	External		-
	setContractURI	External	✓	-
<b>IThirdwebPlatformFee</b>	Interface			
	getPlatformFeeInfo	External		-
	setPlatformFeeInfo	External	✓	-
<b>IThirdwebPrimarySale</b>	Interface			
	primarySaleRecipient	External		-
	setPrimarySaleRecipient	External	✓	-
<b>ITWFee</b>	Interface			

	getFeelInfo	External		-
<b>IWETH</b>	Interface			
	deposit	External	Payable	-
	withdraw	External	✓	-
	transfer	External	✓	-
<b>ITokenERC20</b>	Interface	IThirdwebC ontract, IThirdwebPri marySale, IThirdwebPI atformFee, IERC20Upgr adeable		
	verify	External		-
	mintTo	External	✓	-
	mintWithSignature	External	Payable	-
<b>CurrencyTrans ferLib</b>	Library			
	transferCurrency	Internal	✓	
	transferCurrencyWithWrapperAndBala nceCheck	Internal	✓	
	safeTransferERC20	Internal	✓	
	safeTransferERC20WithBalanceCheck	Internal	✓	
	safeTransferNativeToken	Internal	✓	
	safeTransferNativeTokenWithWrapper	Internal	✓	
<b>FeeType</b>	Library			
<b>ERC2771Conte xtUpgradeable</b>	Implementation	Initializable, ContextUpg radeable		
	__ERC2771Context_init	Internal	✓	onlyInitializing
	__ERC2771Context_init_unchained	Internal	✓	onlyInitializing
	isTrustedForwarder	Public		-
	_msgSender	Internal		
	_msgData	Internal		

<b>TokenERC20</b>	Implementation	Initializable, Reentrancy GuardUpgr adeable, ERC2771Co ntextUpgrad eable, MulticallUpg radeable, ERC20Burn ableUpgrad eable, ERC20Paus ableUpgrad eable, ERC20Votes Upgradeabl e, ITokenERC2 0, AccessCont rolEnumerab leUpgradea ble		
	<Constructor>	Public	✓	initializer
	initialize	External	✓	initializer
	contractType	External		-
	contractVersion	External		-
	_afterTokenTransfer	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	mintTo	Public	✓	-
	verify	Public		-
	mintWithSignature	External	Payable	nonReentrant
	setPrimarySaleRecipient	External	✓	onlyRole
	setPlatformFeeInfo	External	✓	onlyRole
	getPlatformFeeInfo	External		-
	collectPrice	Internal	✓	
	_mintTo	Internal	✓	
	verifyRequest	Internal	✓	
	recoverAddress	Internal		

	_encodeRequest	Internal		
	pause	Public	✓	-
	unpause	Public	✓	-
	setContractURI	External	✓	onlyRole
	_msgSender	Internal		
	_msgData	Internal		



# Contract Flow



## Domain Info

<b>Domain Name</b>	crypto4ac.com
<b>Registry Domain ID</b>	2700508308_DOMAIN_COM-VRSN
<b>Creation Date</b>	2022-06-01T04:49:11Z
<b>Updated Date</b>	2022-06-02T03:17:48Z
<b>Registry Expiry Date</b>	2023-06-01T04:49:11Z
<b>Registrar WHOIS Server</b>	whois.google.com
<b>Registrar URL</b>	<a href="https://domains.google.com">https://domains.google.com</a>
<b>Registrar</b>	Google LLC
<b>Registrar IANA ID</b>	895

The domain was created 10 months before the creation of the audit.

There is no public billing information, the creator is protected by the privacy settings.

## Summary

There are some functions that could be abused by the 'admin' and 'minter' roles like stopping transactions and minting tokens. If the mint functionality is abused, then the contract will be highly inflated. The contract contains an off-chain mechanism for signing mint messages. Additionally, it uses an external contract to determine some of the mint fees. We state that the owner privileges are necessary and required for proper protocol operations. Thus, we emphasise the contract owner to be extra careful with the credentials.

# 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>