SOLIDProof

Bring trust into your projects

Blockchain Security | Smart Contract Audits | KYC

MADE IN GERMANY

Audit

Security Assessment
29. October, 2021

For

CFS

CRYPTO FOR SPEED

Disclaimer

SolidProof.io reports are not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. These reports are not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team. SolidProof.io do not cover testing or auditing the integration with external contract or services (such as Unicrypt, Uniswap, PancakeSwap etc’...)

SolidProof.io Audits do not provide any warranty or guarantee regarding the absolute bug- free nature of the technology analyzed, nor do they provide any indication of the technology proprietors. SolidProof Audits should not be used in any way to make decisions around investment or involvement with any particular project. These reports in no way provide investment advice, nor should be leveraged as investment advice of any sort.

SolidProof.io Reports represent an extensive auditing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology. Blockchain technology and cryptographic assets present a high level of ongoing risk. SolidProof’s position is that each company and individual are responsible for their own due diligence and continuous security. SolidProof in no way claims any guarantee of security or functionality of the technology we agree to analyze.

Version	Date	Description
1.0	29. October 2021	• Layout project • Automated- /Manual-Security Testing • Summary

Network
Binance Smart Chain (BEP20)

Website
https://cryptoforspeed.com/

Telegram
https://t.me/cryptoForSpeed
https://t.me/cryptoforspeedchannel

Twitter
https://twitter.com/cryptoforspeed

Github
https://github.com/CryptoForSpeed

Discord
https://discord.gg/su3pn62aYE

Reddit
https://www.reddit.com/r/CryptoForSpeed/

Medium
https://medium.com/@cryptoforspeed

TikTok
https://www.tiktok.com/@cryptoforspeed

Description

CryptoForSpeed is a cross platform racing gameFi. Players can obtain CFS tokens as reward through various racing models. Innovative game models, It should be noted, we integrate the real world with the virtual world to open the racing metaverse for players.At the same time, we are trying to establish cooperative relations with top automobile companies so that we can manage the abundant and enjoyable features more efficient.

Project Engagement

During the 25th of October 2021, **CryptoForSpeed Team** engaged Solidproof.io to audit smart contracts that they created. The engagement was technical in nature and focused on identifying security flaws in the design and implementation of the contracts. They provided Solidproof.io with access to their code repository and whitepaper.

Logo

Contract Link

v1.0

TBA

Vulnerability & Risk Level

Risk represents the probability that a certain source-threat will exploit vulnerability, and the impact of that event on the organization or system. Risk Level is computed based on CVSS version 3.0.

Level	Value	Vulnerability	Risk (Required Action)
Critical	9 - 10	A vulnerability that can disrupt the contract functioning in a number of scenarios, or creates a risk that the contract may be broken.	Immediate action to reduce risk level.
High	7 – 8.9	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.	Implementation of corrective actions as soon aspossible.
Medium	4 – 6.9	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.	Implementation of corrective actions in a certain period.
Low	2 – 3.9	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.	Implementation of certain corrective actions or accepting the risk.
Informational	0 – 1.9	A vulnerability that have informational character but is not effecting any of the code.	An observation that does not determine a level of risk

Auditing Strategy and Techniques Applied

Throughout the review process, care was taken to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices. To do so, reviewed line-by-line by our team of expert pentesters and smart contract developers, documenting any issues as there were discovered.

Methodology

The auditing process follows a routine series of steps:

1. Code review that includes the following:
    i) Review of the specifications, sources, and instructions provided to SolidProof to make sure we understand the size, scope, and functionality of the smart contract.
    ii) Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
    iii) Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to SolidProof describe.

2. Testing and automated analysis that includes the following:
    i) Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
    ii) Symbolic execution, which is analysing a program to determine what inputs causes each part of a program to execute.

3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.

4. Specific, itemized, actionable recommendations to help you take steps to secure your smart contracts.

Used Code from other Frameworks/Smart Contracts (direct imports)

Imported packages:

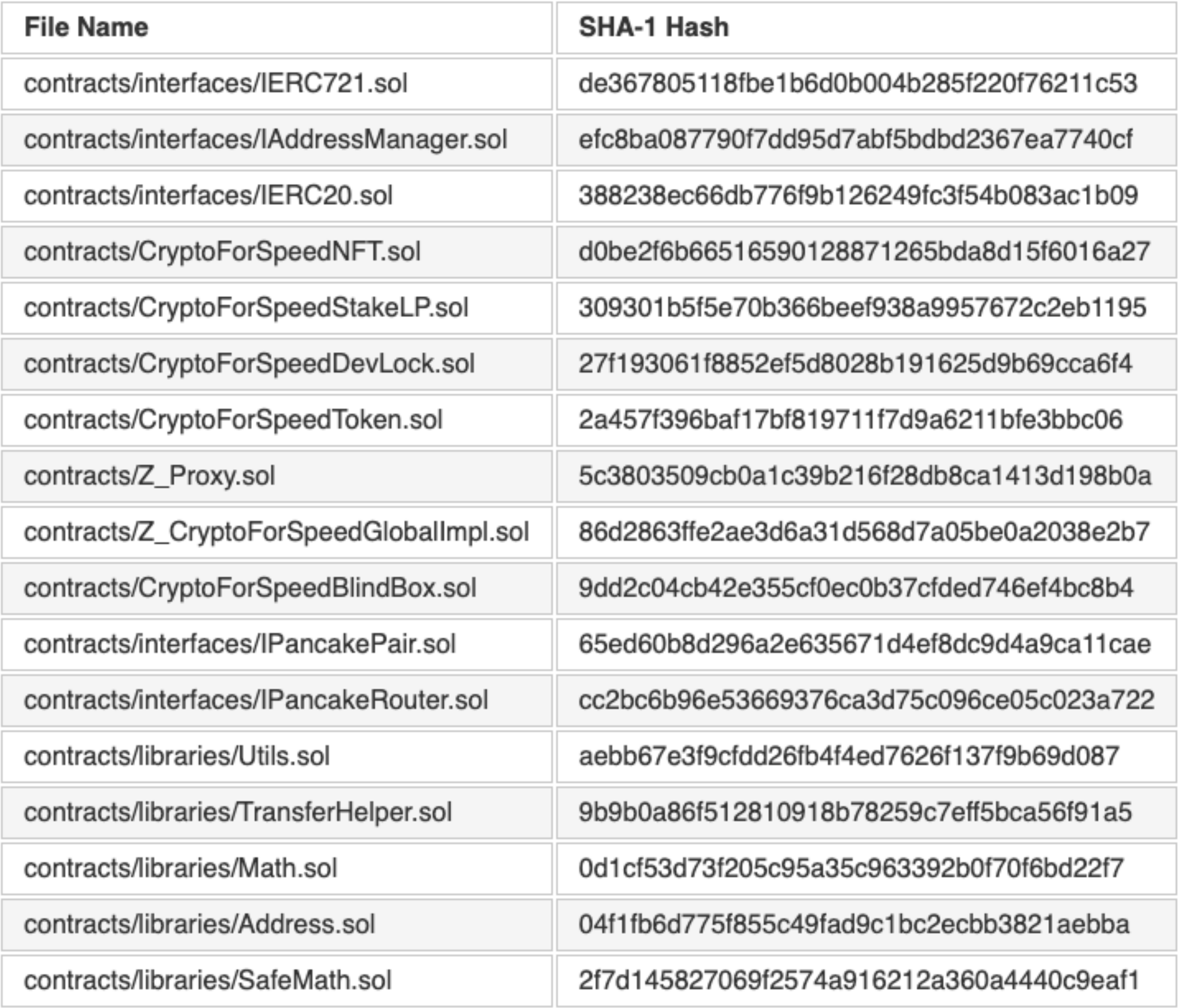
- Look at inheritance graph

Tested Contract Files

This audit covered the following files listed below with a SHA-1 Hash.

A file with a different Hash has been modified, intentionally or otherwise, after the security review. A different Hash could be (but not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of this review.

v1.0

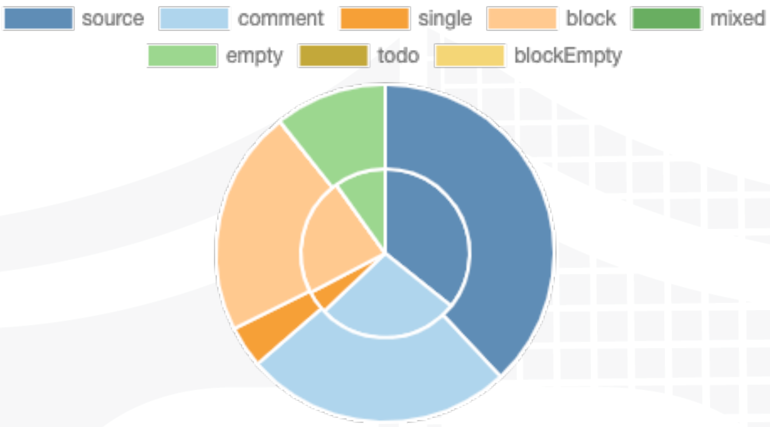


File Name	SHA-1 Hash
contracts/interfaces/IERC721.sol	de367805118fbe1b6d0b004b285f220f76211c53
contracts/interfaces/IAddressManager.sol	efc8ba087790f7dd95d7abf5bdbd2367ea7740cf
contracts/interfaces/IERC20.sol	388238ec66db776f9b126249fc3f54b083ac1b09
contracts/CryptoForSpeedNFT.sol	d0be2f6b66516590128871265bda8d15f6016a27
contracts/CryptoForSpeedStakeLP.sol	309301b5f5e70b366beef938a9957672c2eb1195
contracts/CryptoForSpeedDevLock.sol	27f193061f8852ef5d8028b191625d9b69cca6f4
contracts/CryptoForSpeedToken.sol	2a457f396baf17bf819711f7d9a6211bfe3bbc06
contracts/Z_Proxy.sol	5c3803509cb0a1c39b216f28db8ca1413d198b0a
contracts/Z_CryptoForSpeedGlobalImpl.sol	86d2863ffe2ae3d6a31d568d7a05be0a2038e2b7
contracts/CryptoForSpeedBlindBox.sol	9dd2c04cb42e355cf0ec0b37cfded746ef4bc8b4
contracts/interfaces/IPancakePair.sol	65ed60b8d296a2e635671d4ef8dc9d4a9ca11cae
contracts/interfaces/IPancakeRouter.sol	cc2bc6b96e53669376ca3d75c096ce05c023a722
contracts/libraries/Utils.sol	aebb67e3f9cfdd26fb4f4ed7626f137f9b69d087
contracts/libraries/TransferHelper.sol	9b9b0a86f512810918b78259c7eff5bca56f91a5
contracts/libraries/Math.sol	0d1cf53d73f205c95a35c963392b0f70f6bd22f7
contracts/libraries/Address.sol	04f1fb6d775f855c49fad9c1bc2ecbb3821aebba
contracts/libraries/SafeMath.sol	2f7d145827069f2574a916212a360a4440c9eaf1

Metrics

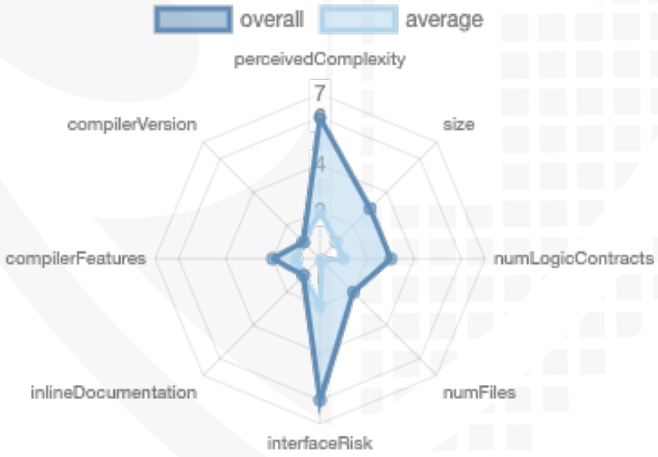
Source Lines

v1.0



Risk Level

v1.0



Capabilities

Components

Version	Contracts	Libraries	Interfaces	Abstract
1.0	10	6	10	8

Exposed Functions

This section lists functions that are explicitly declared public or payable. Please note that getter methods for public stateVars are not included.

Version	Public	Payable
1.0	182	8

Version	External	Internal	Private	Pure	View
1.0	103	183	7	28	64

State Variables

Version	Total	Public
1.0	60	34

Capabilities

Version	Solidity Versions observed	Experimental Features	Can Receive Funds	Uses Assembly	Has Destroyable Contracts
1.0	`=0.6.6`		`yes`	`yes` (3 asm blocks)	

Version	Transfers ETH	Low-Level Calls	DelegateCall	Uses Hash Functions	ECRecover	New/Create/Create2
1.0	`yes`		`yes`	`yes`		

Scope of Work

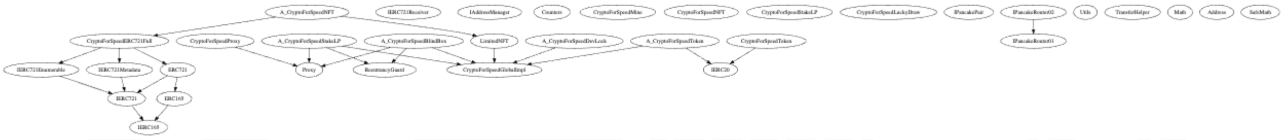
The above token Team provided us with the files that needs to be tested (Github, Bscscan, Etherscan, files, etc.). The scope of the audit is the main contract (usual the same name as team appended with .sol).

We will verify the following claims:

1. Correct implementation of Token standard
2. Deployer cannot mint any new tokens
3. Deployer cannot burn or lock user funds
4. Deployer cannot pause the contract
5. Overall checkup (Smart Contract Security)

Inheritance Graph

v1.0



Verify Claims

Correct implementation of Token standard

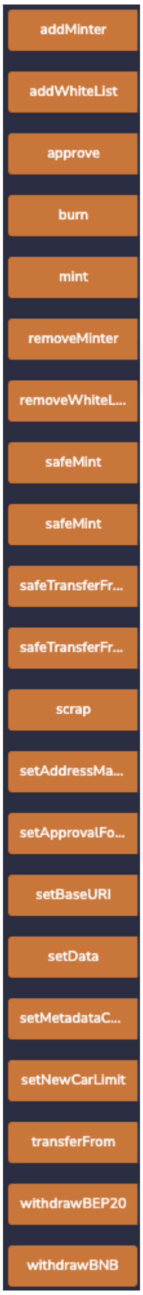
Tested	Verified
✓	✓

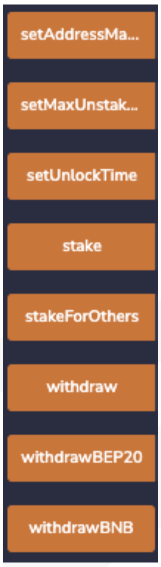
Function	Description	Exist	Tested	Verified
TotalSupply	provides information about the total token supply	✓	✓	✓
BalanceOf	provides account balance of the owner's account	✓	✓	✓
Transfer	executes transfers of a specified number of tokens to a specified address	✓	✓	✓
TransferFrom	executes transfers of a specified number of tokens from a specified address	✓	✓	✓
Approve	allow a spender to withdraw a set number of tokens from a specified account	✓	✓	✓
Allowance	returns a set number of tokens from a spender to the owner	✓	✓	✓

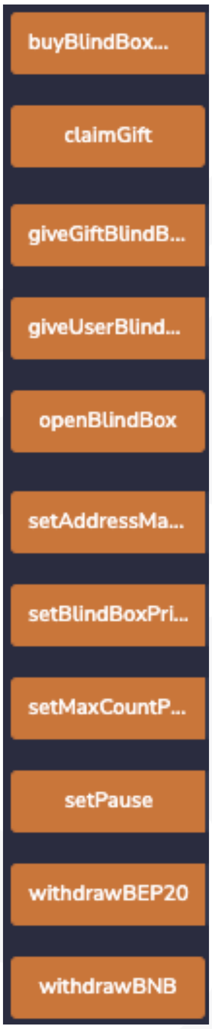
Write functions of contract

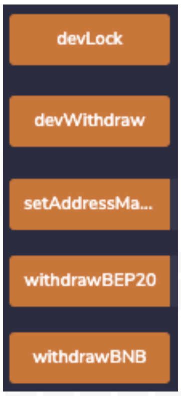
CryptoFprSpeed

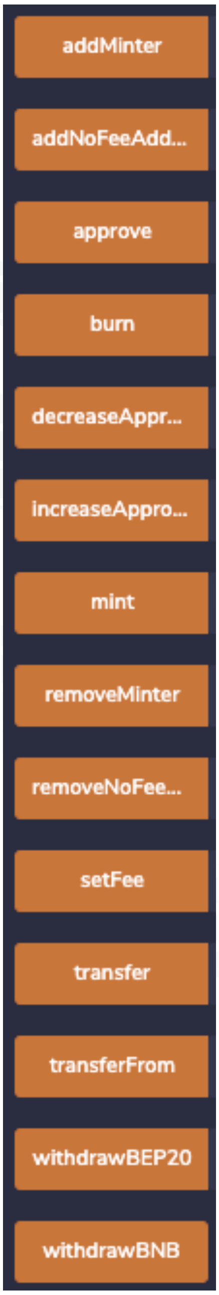
-NFT: -TakeLP: -BlindBox -DevLock -Token



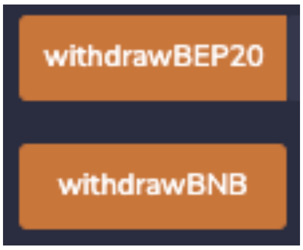








Z_CryptoForSpeedGlobalImpl:



Deployer cannot mint any new tokens

File	Name	Exist	Tested	Verified
A_CRYPTOFORSPEEDSTAKELP	Deployer cannot mint	–	–	–
A_CRYPTOFORSPEEDNFT	Deployer cannot mint	✓	✓	✗
A_CRYPTOFORSPEEDBLINDBOX	Deployer cannot mint	–	–	–
A_CRYPTOFORSPEEDDEVLOCK	Deployer cannot mint	–	–	–
A_CRYPTOFORSPEEDTOKEN	Deployer cannot mint	✓	✓	✗
CRYPTOFORSPEEDGLOBALIMPL	Deployer cannot mint	–	–	–

Max / Total Supply: 100.000.000

Deployer cannot burn or lock user funds

File	Name	Exist	Teste d	Verified
A_CRYPTOFOR SPEEDSTAKEL P	Deployer cannot lock	✓	✓	✓
A_CRYPTOFOR SPEEDSTAKEL P	Deployer cannot burn	–	–	–
A_CRYPTOFOR SPEEDNFT	Deployer cannot lock	✓	✓	✓
A_CRYPTOFOR SPEEDNFT	Deployer cannot burn	✓	✓	✗
A_CRYPTOFOR SPEEDBLINDB OX	Deployer cannot lock	✓	✓	✓
A_CRYPTOFOR SPEEDBLINDB OX	Deployer cannot burn	✓	✓	✗
A_CRYPTOFOR SPEEDDEVLO CK	Deployer cannot lock	–	–	–
A_CRYPTOFOR SPEEDDEVLO CK	Deployer cannot burn	–	–	–
A_CRYPTOFOR SPEEDTOKEN	Deployer cannot lock	✓	✓	✓
A_CRYPTOFOR SPEEDTOKEN	Deployer cannot burn	✓	✓	✗
CRYPTOFORS PEEDGLOBALI MPL	Deployer cannot lock	–	–	–
CRYPTOFORS PEEDGLOBALI MPL	Deployer cannot burn	–	–	–

Deployer cannot pause the contract

File	Name	Exist	Tested	Verified
A_CRYPTOFORSPEEDSTAKELP	Deployer cannot pause	–	–	–
A_CRYPTOFORSPEEDNFT	Deployer cannot pause	–	–	–
A_CRYPTOFORSPEEDBLINDBOX	Deployer cannot pause	✓	✓	✗
A_CRYPTOFORSPEEDDEVLOCK	Deployer cannot pause	–	–	–
A_CRYPTOFORSPEEDTOKEN	Deployer cannot pause	–	–	–
CRYPTOFORSPEEDGLOBALIMPL	Deployer cannot pause	–	–	–

Overall checkup (Smart Contract Security)

Tested	Verified
✓	✓

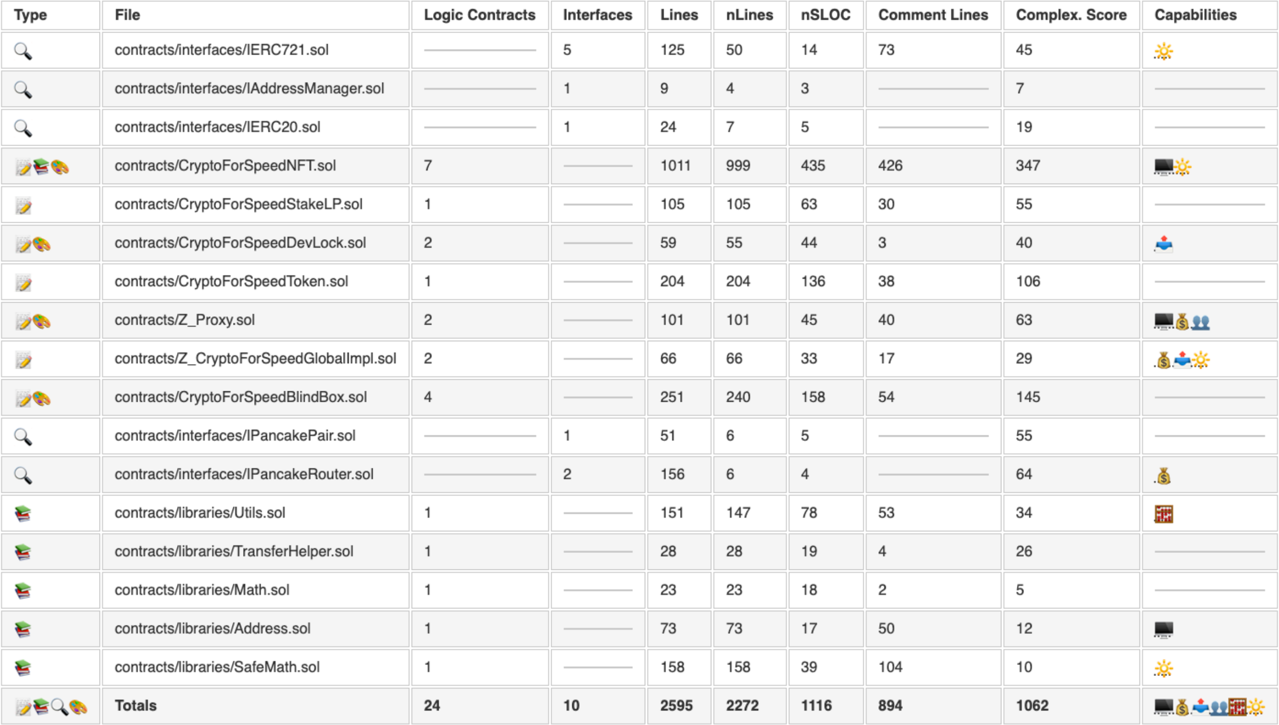
Legend

Attribute	Symbol
Verfified / Checked	✓
Partly Verified	⚑
Unverified / Not checked	✗
Not available	–

CallGraph

Source Units in Scope

v1.0



Type	File	Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
🔍	contracts/interfaces/IERC721.sol	—	5	125	50	14	73	45	🔆
🔍	contracts/interfaces/IAddressManager.sol	—	1	9	4	3	—	7	—
🔍	contracts/interfaces/IERC20.sol	—	1	24	7	5	—	19	—
📝📚🎨	contracts/CryptoForSpeedNFT.sol	7	—	1011	999	435	426	347	🖥🔆
📝	contracts/CryptoForSpeedStakeLP.sol	1	—	105	105	63	30	55	—
📝🎨	contracts/CryptoForSpeedDevLock.sol	2	—	59	55	44	3	40	📥
📝	contracts/CryptoForSpeedToken.sol	1	—	204	204	136	38	106	—
📝🎨	contracts/Z_Proxy.sol	2	—	101	101	45	40	63	🖥💰👥
📝	contracts/Z_CryptoForSpeedGlobalImpl.sol	2	—	66	66	33	17	29	💰📥🔆
📝🎨	contracts/CryptoForSpeedBlindBox.sol	4	—	251	240	158	54	145	—
🔍	contracts/interfaces/IPancakePair.sol	—	1	51	6	5	—	55	—
🔍	contracts/interfaces/IPancakeRouter.sol	—	2	156	6	4	—	64	💰
📚	contracts/libraries/Utils.sol	1	—	151	147	78	53	34	🧮
📚	contracts/libraries/TransferHelper.sol	1	—	28	28	19	4	26	—
📚	contracts/libraries/Math.sol	1	—	23	23	18	2	5	—
📚	contracts/libraries/Address.sol	1	—	73	73	17	50	12	🖥
📚	contracts/libraries/SafeMath.sol	1	—	158	158	39	104	10	🔆
📝📚🔍🎨	**Totals**	**24**	**10**	**2595**	**2272**	**1116**	**894**	**1062**	🖥💰📥👥🧮🔆

Legend

Attribute	Description
Lines	total lines of the source unit
nLines	normalized lines of the source unit (e.g. normalizes functions spanning multiple lines)
nSLOC	normalized source lines of code (only source-code lines; no comments, no blank lines)
Comment Lines	lines containing single or block comments
Complexity Score	a custom complexity score derived from code statements that are known to introduce code complexity (branches, loops, calls, external interfaces, ...)

Audit Results

AUDIT PASSED

Critical issues

- no critical issues found -

High issues

- no high issues found -

Medium issues

- no medium issues found -

Low issues

Issue	File	Type	Line	Description
#1	Main	Contract doesn't import npm packages from source (like OpenZeppelin etc.)	-	We recommend to import all packages from npm directly without flatten the contract. Functions could be modified or can be susceptible to vulnerabilities
#2	CryptoForSpeedNFT	Missing Zero Address Validation (missing-zero-check)	816	Check that the address is not zero
#3	CryptoForSpeedToken	Missing Zero Address Validation (missing-zero-check)	31	Check that the address is not zero
#4	CryptoForSpeedToken	State variable visibility is not set	10, 11, 12, 14, 20	It is best practice to set the visibility of state variables explicitly
#5	CryptoForSpeedBlindBox	Missing Events Arithmetic	93	Emit an event for critical parameter changes
#6	CryptoForSpeedStakeLP	Missing Events Arithmetic	40, 44	Emit an event for critical parameter changes

#7	CryptoForSpeed Token	Missing Events Arithmetic	180	Emit an event for critical parameter changes
#8	CryptoForSpeed NFT	Missing Events Arithmetic	440	Emit an event for critical parameter changes
#9	CryptoForSpeed BlindBox	Multiple calls in a loop	163	A_CryptoForSpeedBlindBox.openBlindBox() (CryptoForSpeedBlindBox.sol:163-190) has external calls inside a loop: (succ1,id1) = nft.mint(msg.sender,userInfo[msg.sender]._type,dna) (CryptoForSpeedBlindBox.sol#178) Favor [pull over push](https://github.com/ethereum/wiki/wiki/Safety#favor-pull-over-push-for-external-calls) strategy for external calls

Informational issues

Issue	File	Type	Line	Description
#1	CryptoForSpeed NFT	Functions that are not used	455-457	Remove unused functions
#2	CryptoForSpeed NFT	Missing inheritance	794-1012	A_CryptoForSpeedNFT (CryptoForSpeedNFT.sol:794-1012) should inherit from CryptoForSpeedNFT (CryptoForSpeedBlindBox.sol#13-15)
#3	CryptoForSpeed StakeLP	Missing inheritance	10-106	A_CryptoForSpeedStakeLP (CryptoForSpeedStakeLP.sol:10-106) should inherit from CryptoForSpeedStakeLP (CryptoForSpeedBlindBox.sol#17-19)

#4	CryptoForSpeedToken	Missing inheritance	8-205	A_CryptoForSpeedToken (CryptoForSpeedToken.sol:8-205) should inherit from CryptoForSpeedToken (CryptoForSpeedDevLock.sol #8-10)

Audit Comments

29. October 2021:

- All contracts which inherited from Proxy contract can lock ether
    - Remove the payable attribute or add a withdraw function

SWC Attacks

ID	Title	Relationships	Status
SWC-136	Unencrypted Private Data On-Chain	CWE-767: Access to Critical Private Variable via Public Method	PASSED
SWC-135	Code With No Effects	CWE-1164: Irrelevant Code	PASSED
SWC-134	Message call with hardcoded gas amount	CWE-655: Improper Initialization	PASSED
SWC-133	Hash Collisions With Multiple Variable Length Arguments	CWE-294: Authentication Bypass by Capture-replay	PASSED
SWC-132	Unexpected Ether balance	CWE-667: Improper Locking	PASSED
SWC-131	Presence of unused variables	CWE-1164: Irrelevant Code	PASSED
SWC-130	Right-To-Left-Override control character (U+202E)	CWE-451: User Interface (UI) Misrepresentation of Critical Information	PASSED
SWC-129	Typographical Error	CWE-480: Use of Incorrect Operator	PASSED
SWC-128	DoS With Block Gas Limit	CWE-400: Uncontrolled Resource Consumption	PASSED

SWC-127	Arbitrary Jump with Function Type Variable	CWE-695: Use of Low-Level Functionality	PASSED
SWC-125	Incorrect Inheritance Order	CWE-696: Incorrect Behavior Order	PASSED
SWC-124	Write to Arbitrary Storage Location	CWE-123: Write-what-where Condition	PASSED
SWC-123	Requirement Violation	CWE-573: Improper Following of Specification by Caller	PASSED
SWC-122	Lack of Proper Signature Verification	CWE-345: Insufficient Verification of Data Authenticity	PASSED
SWC-121	Missing Protection against Signature Replay Attacks	CWE-347: Improper Verification of Cryptographic Signature	PASSED
SWC-120	Weak Sources of Randomness from Chain Attributes	CWE-330: Use of Insufficiently Random Values	PASSED
SWC-119	Shadowing State Variables	CWE-710: Improper Adherence to Coding Standards	PASSED
SWC-118	Incorrect Constructor Name	CWE-665: Improper Initialization	PASSED
SWC-117	Signature Malleability	CWE-347: Improper Verification of Cryptographic Signature	PASSED

SWC-116	Timestamp Dependence	CWE-829: Inclusion of Functionality from Untrusted Control Sphere	PASSED
SWC-115	Authorization through tx.origin	CWE-477: Use of Obsolete Function	PASSED
SWC-114	Transaction Order Dependence	CWE-362: Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')	PASSED
SWC-113	DoS with Failed Call	CWE-703: Improper Check or Handling of Exceptional Conditions	PASSED
SWC-112	Delegatecall to Untrusted Callee	CWE-829: Inclusion of Functionality from Untrusted Control Sphere	PASSED
SWC-111	Use of Deprecated Solidity Functions	CWE-477: Use of Obsolete Function	PASSED
SWC-110	Assert Violation	CWE-670: Always-Incorrect Control Flow Implementation	PASSED
SWC-109	Uninitialized Storage Pointer	CWE-824: Access of Uninitialized Pointer	PASSED
SWC-108	State Variable Default Visibility	CWE-710: Improper Adherence to Coding Standards	NOT PASSED
SWC-107	Reentrancy	CWE-841: Improper Enforcement of Behavioral Workflow	PASSED
SWC-106	Unprotected SELFDESTRUCT Instruction	CWE-284: Improper Access Control	PASSED

SWC-105	Unprotected Ether Withdrawal	CWE-284: Improper Access Control	**PASSED**
SWC-104	Unchecked Call Return Value	CWE-252: Unchecked Return Value	**PASSED**
SWC-103	Floating Pragma	CWE-664: Improper Control of a Resource Through its Lifetime	**PASSED**
SWC-102	Outdated Compiler Version	CWE-937: Using Components with Known Vulnerabilities	**PASSED**
SWC-101	Integer Overflow and Underflow	CWE-682: Incorrect Calculation	**PASSED**
SWC-100	Function Default Visibility	CWE-710: Improper Adherence to Coding Standards	**PASSED**

Solid Proofed

Blockchain Security | Smart Contract Audits | KYC

MADE IN GERMANY