AB Core

Security Assessment

CertiK Assessed on Mar 25th, 2025

CertiK Assessed on Mar 25th, 2025

AB Core

The security assessment was prepared by CertiK, the leader in Web3.0 security.

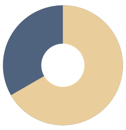
Executive Summary

TYPES	ECOSYSTEM	METHODS
Layer 1, Platform	Ethereum (ETH)	Manual Review, Static Analysis

LANGUAGE	TIMELINE	KEY COMPONENTS
Golang	Delivered on 03/25/2025	N/A

CODEBASE	COMMITS
tag: v1.13.15-ab-1.0	2c08edb4b37daf1e33649299e003d2f67b1dce4d
View All in Codebase Page	View All in Codebase Page

Vulnerability Summary



3	1	0	2	0
Total Findings	Resolved	Partially Resolved	Acknowledged	Declined

	Count	Severity	Status	Description
■	0	Centralization		Centralization findings highlight privileged roles & functions and their capabilities, or instances where the project takes custody of users' assets.
■	0	Critical		Critical risks are those that impact the safe functioning of a platform and must be addressed before launch. Users should not invest in any project with outstanding critical risks.
■	0	Major		Major risks may include logical errors that, under specific circumstances, could result in fund losses or loss of project control.
■	0	Medium		Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.
■	2	Minor	1 Resolved, 1 Acknowledged	Minor risks can be any of the above, but on a smaller scale. They generally do not compromise the overall integrity of the project, but they may be less efficient than other solutions.
■	1	Informational	1 Acknowledged	Informational errors are often recommendations to improve the style of the code or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

TABLE OF CONTENTS | AB CORE

CODEBASE | AB CORE

Repository

tag: v1.13.15-ab-1.0

Commit

2c08edb4b37daf1e33649299e003d2f67b1dce4d

AUDIT SCOPE | AB CORE

3 files audited ● 3 files without findings

ID	Repo	File	SHA256 Checksum
BFG	ABFoundationGlobal/abcore	consensus/clique/api.go	e4de34b46ea458886c1f2ffb5237b1309f9677fdb4ed6d974d6a6ab8a7d417ac
CLI	ABFoundationGlobal/abcore	consensus/clique/clique.go	1299c63e515cacd3ffe3dfca48211fc10a25e743f6816fd8219015a81f529080
SNA	ABFoundationGlobal/abcore	consensus/clique/snapshot.go	fc38d902a4c1cb361ffa2d10ee5b524c3aeb3d03714682e498cffa3d0ae1028a

APPROACH & METHODS | AB CORE

This report has been prepared for AB Chain to discover issues and vulnerabilities in the source code of the AB Core project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Manual Review and Static Analysis techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Testing the smart contracts against both common and uncommon attack vectors;
- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

REVIEW NOTES | AB CORE

The AB Core is a fork of go-ethereum `v1.13.15`, incorporating modifications to implement a Proof of Authority (PoA) consensus engine. Below is the scope of the auditing engagement:

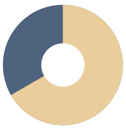
Clique Consensus Module:

The audit will evaluate AB Core's implementation of a PoA (Proof of Authority) consensus model, which is configured with a 1-second block interval.

Upstream Known Vulnerabilities:

Known vulnerabilities from the most recent upstream go-ethereum release, as of the time of writing, particularly within the Execution Layer (EL), have been identified and addressed.

FINDINGS | AB CORE



3	0	0	0	0	2	1
Total Findings	Critical	Centralization	Major	Medium	Minor	Informational

This report has been prepared to discover issues and vulnerabilities for AB Core. Through this audit, we have uncovered 3 issues ranging from different severity levels. Utilizing the techniques of Manual Review & Static Analysis to complement rigorous manual code reviews, we discovered the following findings:

ID	Title	Category	Severity	Status
2C0-01	Potential Issues From Upstream `go-ethereum`	Logical Issue	Minor	Resolved
REA-01	Risks Of Implementing 1-Second Block Time In A PoA-Based Ethereum Chain	Design Issue	Minor	Acknowledged
ABG-01	Potential Limitations Of Ethereum's Clique POA Consensus Algorithm	Design Issue	Informational	Acknowledged

2C0-01 | POTENTIAL ISSUES FROM UPSTREAM `go-ethereum`

Category	Severity	Location	Status
Logical Issue	Minor	framework.go (2c08edb): 65~66; big.go (2c08edb): 57~59; integer.go (2c08edb): 49~50; generate.go (2c08edb): 639~659; snapshot.go (2c08edb): 831~832; validation.go (2c08edb): 203~204; simulated_beacon.go (2c08edb): 266~268; api.go (2c08edb): 1167~1185, 1490~1492; metrics.go (2c08edb): 69~73; table.go (2c08edb): 49~50; nat.go (2c08edb): 141~142; iterator.go (2c08edb): 314~315	Resolved

Description

Repository

- AB Core

Commit hash:

- `2c08edb4b37daf1e33649299e003d2f67b1dce4d`

Files:

- `trie/iterator.go`
- `cmd/devp2p/internal/v4test/framework.go`
- `core/state/snapshot/snapshot.go`
- `core/state/snapshot/generate.go`
- `common/math/big.go`
- `common/math/integer.go`

When the trie contains a value node whose key is a prefix of the passed start path, this value node's key (with terminator) compares >= to the seeked path, so seek stops at it, although the actual path is lexicographically less than the start path.

Recommend to fix the issue according to the `PR-27838`

A typo in code node.TCP() => node.UDP()

Recommend to fix the issue according to the `PR-29879`

---

enode.Node has separate accessor functions for getting the IP, UDP port and TCP port. These methods performed separate checks for attributes set in the ENR. The accessor methods will now return cached information, and the endpoint is determined when the node is created. The logic to determine the preferred endpoint is now more correct, and considers how 'global' each address is when both IPv4 and IPv6 addresses are present in the ENR.

Recommend to fix the issue according to the `PR-29801` and `PR-29827`

Data races happen in snapshot access.

Recommend to fix the issue according to the `PR-30001` and `PR-30011`

Out of bounds access in json unmarshalling in math.

Recommend to fix the issue according to the `PR-30014`

AddMapping always returns 0 which may lead to some misconfiguration per the connection specifications with implementation for --nat extip:. Returning 0 causes various trouble, as we now treat this as our externally reachable port. As node.UDPEndpoint returns ok == false when our port is 0, this causes us to generate an invalid ping packet.

```
func (ExtIP) AddMapping(string, int, int, string, time.Duration) (uint16, error
) { return 0, nil }
```

Recommend to fix the issue according to the `PR-30234`

The address recover is executed and cached in ValidateTransaction already. It's expected that the cached one is returned in ValidateTransactionWithState. However, currently, we use the wrong function signer.Sender instead of types.Sender which will do all the address recover again.

Recommend to fix the issue according to the `PR-30208`

There is a flaw in the snapshot sync that it attempts to stop the state snapshot generation, which could potentially cause the system to halt if the generation is not currently running.

Recommend to fix the issue according to the `PR-30040`

Recommendation

Recommend to fix the issues according to the aforementioned PRs.

Alleviation

[AB Core Team - 03/20/2025] :

The team acknowledged the issues and fix them according to upstream PRs, the change is reflected in the main branch with commit hash: `c91b0164d4e6e91b8d7cb205792a6aef61010478` .

REA-01 | RISKS OF IMPLEMENTING 1-SECOND BLOCK TIME IN A POA-BASED ETHEREUM CHAIN

Category	Severity	Location	Status
Design Issue	Minor	README.md (2c08edb): 7~8	Acknowledged

Description

Repository

- AB Core

Commit hash:

- `2c08edb4b37daf1e33649299e003d2f67b1dce4d`

Setting a 1-second block time in a Proof of Authority (PoA) Ethereum chain significantly increases transaction throughput but introduces below potential risks:

1. Network Latency and Forking Risks

Block Propagation Delays:

In PoA, validators sequentially produce blocks. With a 1-second block time, even minor network delays (e.g., 500ms) can prevent blocks from reaching all nodes before the next validator begins mining.

Consequence:

Temporary forks (orphaned blocks) occur, requiring nodes to resolve conflicts by reorganizing the chain. Frequent reorgs reduce transaction finality, enabling double-spend attacks if malicious actors exploit the ambiguity in block confirmations.

2. Validator Synchronization Challenges

Strict Timing Requirements:

Validators must process transactions, sign blocks, and broadcast them within 1 second. Hardware bottlenecks (e.g., slow disk I/O or CPU) can disrupt this cycle.

Consequence:

A single slow validator may stall the chain or force other nodes to skip its turn, breaking the rotation schedule.

3. Smart Contract Execution Risks

Gas Limits and Execution Time:

Complex smart contracts (e.g., DeFi protocols) may exceed gas limits or require execution times longer than 1 second.

Consequence:

Transactions fail or revert, increasing chain congestion and reducing reliability.

Recommendation

If ultra-high performance is not a priority, it is recommended to opt for slightly longer block times (e.g., 3–5 seconds) based on specific design requirements. This approach strikes a safer balance between performance and reliability.

Reference: `what is the safest minimum block time to use without having any problem on proof`

Alleviation

[AB Core Team - 03/20/2025] :
The team acknowledged this issue:

> After theoretical analysis and actual testing on the test network, we confirmed that 1-second block generation is feasible.

ABG-01 | POTENTIAL LIMITATIONS OF ETHEREUM'S CLIQUE POA CONSENSUS ALGORITHM

Category	Severity	Location	Status
Design Issue	● Informational	clique.go (2c08edb): 17~18	● Acknowledged

Description

Repository

- AB Core

Commit hash:

- `2c08edb4b37daf1e33649299e003d2f67b1dce4d`

Files:

- `consensus/clique/clique.go`

The Clique Proof-of-Authority (PoA) consensus algorithm, while lightweight and easy to deploy, has some limitations that can compromise network stability and security:

Forking Risks:

In Clique, blocks created by `in-turn` validators are published immediately. `Out-of-turn` validators create blocks that are published after a short delay. `In-turn` blocks have a higher difficulty than `out-of-turn` blocks, which allows small forks to resolve to the chain with more in-turn blocks. However, when the `out-of-turn` delay is shorter than the block propagation delay, `out-of-turn` blocks may be published before `in-turn` blocks. This may cause large, irresolvable forks in a network.

Lack of Finality:

Clique is a probabilistic consensus mechanism, requiring multiple confirmations to ensure transaction irreversibility. This leaves room for short-term reorgs (chain reorganizations).

For networks requiring deterministic finality and Byzantine fault tolerance, QBFT (or other BFT-based PoA algorithms) is a superior alternative. Clique remains suitable only for low-stakes testnets or closed environments with tightly controlled validators and ultra-low-latency networks.

Reference:

- `Consensys quorum`

Recommendation

Recommend if possible, consider using enterprise-grade consensus protocol like QBFT instead of Clique.

Alleviation

[AB Core Team - 03/20/2025] :

The team acknowledged this issue:

> We use AB Consensus

OPTIMIZATIONS | AB CORE

ID	Title	Category	Severity	Status
ABF-01	Fix Consensus Config String Representations	Coding Issue	Optimization	Acknowledged

ABF-01 | FIX CONSENSUS CONFIG STRING REPRESENTATIONS

Category	Severity	Location	Status
Coding Issue	Optimization	config.go (2c08edb): 385~387	Acknowledged

Description

Repository

- AB Core

Commit hash:

- `2c08edb4b37daf1e33649299e003d2f67b1dce4d`

Files:

- `params/config.go`

A more suitable representation for the consensus configuration `String` exists, beyond the current implementation outlined below:

`params/config.go`

```
func (c *CliqueConfig) String() string {
    return "clique"
}
```

It is recommended to refer to the PR `params: fix consensus config string representations #29635` to address the clique related part.

Reference:

- `params: fix consensus config string representations #29635`

Recommendation

Recommend to refer to the PR `params: fix consensus config string representations #29635` to address the clique related part.

Alleviation

[AB Core Team - 03/20/2025] :

The team acknowledged this issue:

> No need to merge now.

APPENDIX | AB CORE

Finding Categories

Categories	Description
Coding Issue	Coding Issue findings are about general code quality including, but not limited to, coding mistakes, compile errors, and performance issues.
Logical Issue	Logical Issue findings indicate general implementation issues related to the program logic.
Design Issue	Design Issue findings indicate general issues at the design level beyond program logic that are not covered by other finding categories.

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux "sha256sum" command against the target file.

DISCLAIMER | CERTIK

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you ("Customer" or the "Company") in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes, nor may copies be delivered to any other person other than the Company, without CertiK's prior written consent in each instance.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts CertiK to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing 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. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

The assessment services provided by CertiK is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, CERTIK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, CERTIK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, CERTIK MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET CUSTOMER'S OR ANY OTHER PERSON'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE. WITHOUT LIMITATION TO THE FOREGOING, CERTIK PROVIDES NO WARRANTY OR

UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER CERTIK NOR ANY OF CERTIK'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. CERTIK WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED "AS IS" AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT CERTIK'S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF CERTIK CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.

FOR AVOIDANCE OF DOUBT, THE SERVICES, INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

Elevating Your Entire Web3 Journey

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.