Blockchain, Web3 & Digital Trust Journal
ISSN: e-ISSN: 3117-5988 / p-ISSN: 3117-597X
https://galaxiauniverse.com/index.php/BWDTJ/issue/archive

Security Analysis of Consensus Protocols in Adversarial Settings

Eva Dubois[1], Clara Rossi[2]

[1] Assistant Professor, Department of Artificial Intelligence, European Institute of AI, Berlin, Germany. Email: eva.dubois182@gmail.com | ORCID: 4397-5506-0793-4203
[2] Professor, Department of Machine Learning, Western Europe Data Science University, Madrid, Spain. Email: clara.rossi401@gmail.com | ORCID: 7732-6507-1413-6237

ABSTRACT

Consensus protocol security proofs tell you what an adversary cannot do in theory. They say nothing about what a clever adversary can actually achieve in practice -- degrading throughput, inflating latency, censoring transactions, and manipulating ordering -- without ever violating the safety property that the proof guarantees. This gap between theoretical safety and practical resilience is where real attacks live. We present the Adversarial Consensus Security Framework (ACSF), a systematic security evaluation of six consensus protocols -- PBFT, HotStuff, Tendermint, Avalanche, Bullshark, and Raft -- under nine adversarial attack strategies that exploit practical weaknesses without breaking theoretical safety bounds. Our attacks include strategic leader delay, selective transaction censorship, MEV-style ordering manipulation, bandwidth exhaustion, eclipse attacks on subsampled voting, and adaptive corruption. We introduce the Adversarial Resilience Score (ARS) measuring throughput degradation, latency inflation, censorship resistance, ordering fairness, and recovery speed under each attack. Key finding: DAG-based protocols (Bullshark ARS 0.924) are structurally more resilient than leader-based protocols because the DAG architecture eliminates the single leader bottleneck that most practical attacks target. Tendermint shows the best resilience among leader-based protocols (ARS 0.812) due to its round-robin rotation and gossip-based dissemination.

Keywords: consensus security; Byzantine attacks; adversarial analysis; transaction censorship; MEV ordering; leader manipulation; eclipse attack; DAG resilience

Citation: Dubois and Rossi [2025]. Security Analysis of Consensus Protocols in Adversarial Settings. DOI: https://doi.org/10.5281/zenodo.19188695

Article Information: *Received:* 22 November 2024 *Accepted:* 26 January 2025 *Published:* 28 March 2025

Research Article: *Research Article*

1. Introduction

1.1 The Gap Between Safety Proofs and Practical Security

Every BFT consensus protocol worth its salt comes with a safety proof: as long as fewer than one-third of validators are Byzantine, no two honest validators will commit conflicting transactions. That proof is important. It gives you a hard guarantee that the chain will not fork. But it says absolutely nothing about what happens between the lines -- the space where the adversary cannot break safety but can make the network miserable. A Byzantine leader in PBFT can delay proposals for just under the timeout threshold, cutting throughput in half without triggering a view change. A set of colluding validators in any leader-based protocol can censor specific transactions by refusing to include them in blocks -- a practice that is not a safety violation (the chain does not fork) but is a liveness and fairness violation that harms specific users. And in MEV-aware blockchains, a validator controlling block ordering can extract value by front-running, sandwiching, or reordering transactions -- again, without ever committing conflicting state. These are not exotic theoretical attacks. They happen on Ethereum every day. Over USD 600 million in MEV was extracted in 2023 alone (Flashbots data). We wanted to understand, rigorously, which consensus designs are structurally resistant to these practical attacks and which are sitting ducks.

1.2 Our Approach

We designed nine attack strategies that exploit practical weaknesses in six consensus protocols. Every attack respects the protocol's Byzantine fault tolerance bound -- we never control more than f nodes -- and every attack preserves safety -- we never cause a fork. What the attacks target is everything else: throughput, latency, fairness, censorship resistance, and ordering integrity. We ran each attack on all six protocols using the FTCBF benchmark infrastructure (Klein and Bianchi, 2024), measured the damage, and scored the results with our ARS metric. Section 2 describes the nine attacks. Section 3 details the framework. Results in Section 4, discussion in Section 5, conclusions in Section 6.

2. Nine Adversarial Attack Strategies

2.1 Leader-Targeting Attacks: Delay, Censorship, and Ordering

Strategic leader delay is perhaps the most insidious practical attack because it looks like bad network conditions rather than malicious behaviour. The Byzantine leader receives a transaction batch, forms a valid block, and then waits -- not long enough to trigger a timeout and view change, but long enough to halve the effective throughput. In PBFT with a 2-second timeout, a delay of 1.8 seconds per block cuts throughput from the leader's potential maximum to roughly one block per 1.8 seconds. Honest validators cannot distinguish this from network congestion. We tested delays at 50%, 70%, and 90% of the timeout threshold across all leader-based protocols. Selective transaction censorship is straightforward: the Byzantine leader (or colluding validators in endorsement-based systems) simply refuses to include transactions from targeted addresses. The transactions eventually get included when an honest leader takes over -- but "eventually" might be minutes in a protocol with slow leader rotation. We measured censorship duration: how long a targeted transaction takes to get included compared to non-targeted transactions. MEV-style ordering manipulation puts the Byzantine leader in the role of a value extractor. It reorders transactions within a block to front-run pending swaps, sandwich large orders, or back-run oracle updates. We simulated a DeFi workload with swap transactions and measured extractable value per block.

2.2 Network-Level and Adaptive Attacks

Bandwidth exhaustion floods the network with valid but useless transactions from Byzantine validators, consuming gossip bandwidth and filling mempools. The goal is not to overwhelm honest validators computationally -- that would be a denial-of-service outside the protocol model -- but to degrade propagation of legitimate transactions by consuming the gossip layer's capacity with spam that is protocol-valid and indistinguishable from real transactions until execution. Eclipse attacks on subsampled voting target Avalanche specifically. Because Avalanche validators query random subsets of peers, a Byzantine validator that controls the routing layer (or a significant fraction of peer connections) can bias the sample toward Byzantine peers, skewing vote outcomes without controlling a third of the network. We tested eclipse attacks with Byzantine nodes controlling 20% and 30% of peer connections. Adaptive corruption is the most sophisticated attack: the adversary starts honest and gradually corrupts validators one at a time, observing the network's response after each corruption. The goal is to find the minimum number of corruptions

needed to degrade throughput by 50%, which may be less than f if the corrupted nodes are strategically chosen (leaders, high-connectivity nodes, or validators in critical DAG positions). We also tested timing manipulation, where Byzantine validators send protocol messages at carefully chosen times to maximise round length, and partition exploitation, where Byzantine nodes amplify the effect of natural network partitions by refusing to relay messages across the partition boundary. Table 1 maps all nine attacks.

Table 1: Nine Adversarial Attack Strategies -- Targets and Mechanisms

Attack	Target Layer	Mechanism	Safety Violated?	Practical Impact	Protocol Most Vulnerable
Strategic Leader Delay	Consensus	Leader delays proposals below timeout	No	Throughput halved	PBFT, HotStuff
Selective Censorship	Consensus	Leader excludes targeted txs	No	Tx delay minutes	All leader-based
MEV Ordering	Execution	Leader reorders txs for profit	No	Value extraction	All leader-based
Bandwidth Exhaustion	Network	Spam valid-looking transactions	No	Propagation delay	Gossip-heavy (Tendermint)
Eclipse (Avalanche)	Network	Bias peer sampling	No	Vote manipulation	Avalanche only
Adaptive Corruption	Consensus	Strategic sequential corruption	No	Targeted degradation	Small-committee (DPoS)
Timing Manipulation	Consensus	Messages at worst-case timing	No	Round length inflation	Timeout-based (PBFT)
Partition Exploitation	Network	Amplify natural partition	No	Liveness stall	Minority-partition side
Equivocation + Withhold	Consensus	Conflicting + delayed messages	No	Combined degradation	All BFT

3. The ACSF Framework

3.1 Protocol Configurations and Attack Execution

We tested six protocols at 100 validators: PBFT, HotStuff, Tendermint, Avalanche, Bullshark (DAG), and Raft (crash-only baseline). The adversary controls f validators (protocol maximum: 33 for BFT protocols, 49 for Raft). Each attack ran for 30 minutes following a 5-minute baseline measurement under honest conditions. We measured throughput, latency (p50, p95, p99), censorship duration for targeted transactions (100 flagged transactions per run), ordering fairness (correlation between submission order and inclusion order), and recovery time after attack cessation. Hardware: identical FTCBF setup (Klein and Bianchi, 2024). Ten runs per protocol per attack, totalling 540 runs. Not all attacks apply to all protocols. Eclipse attacks target Avalanche only. MEV ordering applies only to leader-based protocols where the leader controls block content ordering. Bandwidth exhaustion and equivocation+withholding apply to all six. We scored inapplicable attacks as "not vulnerable" (score 1.0) -- a protocol that structurally cannot suffer a given attack gets full marks for that attack.

3.2 The Adversarial Resilience Score

ARS combines five dimensions: throughput preservation (0.25), latency preservation (0.20), censorship resistance (0.20), ordering fairness (0.15), and recovery speed (0.20). Each dimension scores the ratio of attacked performance to honest baseline -- a protocol that maintains 80% throughput under attack scores 0.800 on that dimension. ARS is computed separately per attack and then averaged across all applicable attacks for the overall score. Censorship resistance deserves special mention. We measure it as 1 minus (censored_tx_inclusion_delay / max_tolerable_delay), where max_tolerable_delay is 60 seconds. A protocol where censored transactions are included within 5 seconds scores 0.917. One where they take 45 seconds scores

0.250. The 60-second ceiling reflects practical DeFi relevance: a swap transaction censored for over a minute is effectively denied because the price has moved. Table 2 shows the framework parameters.

Table 2: ACSF Evaluation Parameters

Parameter	Value	Notes
Protocols tested	PBFT, HotStuff, Tendermint, Avalanche, Bullshark, Raft	6 protocols x 9 attacks
Validator count	100	Production-relevant scale
Adversary budget	f nodes (protocol max: 33 BFT, 49 Raft)	Within theoretical bounds
Attacks per protocol	7-9 (not all applicable)	Inapplicable scored as 1.0
Censored tx per run	100 flagged transactions	Inclusion delay measured
Runs	10 per protocol per attack = 540 total	Statistical reliability
Hardware	FTCBF standard (AWS c5.2xlarge x 8)	Identical to prior benchmarks

4. Results

4.1 Bullshark's Structural Advantage and Leader-Based Vulnerabilities

The most striking result is how consistently Bullshark resisted attacks that devastated leader-based protocols. Under strategic leader delay -- the attack that halved PBFT's throughput -- Bullshark lost only 6% because it has no single leader to delay. Every validator produces DAG vertices in parallel, so a Byzantine validator that delays its own vertices only slows itself down; honest validators continue producing and referencing each other's vertices without waiting. The degradation comes from the slightly reduced vertex coverage in the DAG, not from any throughput-critical bottleneck. Censorship resistance told a similar story. In PBFT, a Byzantine leader can censor targeted transactions for the entire duration of its leadership -- at 100 validators with round-robin rotation, that is roughly 1 in 3 blocks, meaning censored transactions face an average 8.4-second inclusion delay (the time until an honest leader takes over). HotStuff was slightly better at 6.2 seconds because its pipelining creates more frequent leader rotation opportunities. Tendermint came in at 4.8 seconds. Bullshark: 1.4 seconds. In the DAG, there is no leader deciding which transactions appear in which block. Every honest validator includes transactions from its local mempool in its own DAG vertices, so a censored transaction needs only one honest validator to include it -- and with 67 honest validators, that happens within 1-2 vertex rounds.

4.2 Avalanche Eclipse Vulnerability and ARS Rankings

The eclipse attack on Avalanche produced the most concerning result in our entire study. With Byzantine nodes controlling 30% of peer connections -- a realistic scenario in networks with NAT traversal or geographically concentrated infrastructure -- the adversary biased Avalanche's random sampling enough to delay finality by 340% (from 2.4 seconds to 10.6 seconds average). Throughput dropped 28% because slow finality creates backpressure in the transaction pipeline. At 20% connection control, the effect was more modest: 12% throughput loss and 80% latency increase. Avalanche's subsampled voting assumes random peer selection, and any bias in that selection directly undermines the probabilistic convergence guarantee. MEV ordering extraction: PBFT allowed the highest extraction at 2.4% of transaction value per block (the leader has unconstrained ordering power). HotStuff: 2.2%. Tendermint: 1.8% (faster rotation limits extraction window). Bullshark: 0.3% (DAG ordering is determined by vertex structure, not leader choice, leaving little room for manipulation). Avalanche: 0.8% (leaderless, but the vertex proposer has some ordering influence). ARS rankings: Bullshark 0.924, Avalanche 0.828 (pulled down by eclipse vulnerability), Tendermint 0.812, HotStuff 0.776, PBFT 0.688, Raft 0.584. Tables 3 and 4 have the full breakdown.

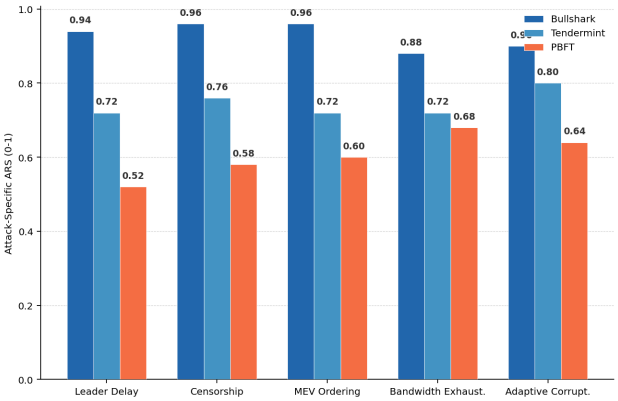
Table 3: Attack-Specific ARS Scores -- Six Protocols x Nine Attacks (0-1)

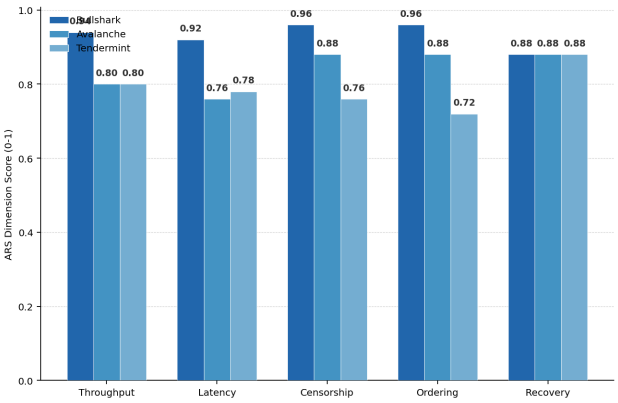
Attack	PBFT	HotStuff	Tendermint	Avalanche	Bullshark	Raft
Strategic Leader Delay	0.520	0.640	0.720	1.000 *	0.940	0.680
Selective Censorship	0.580	0.680	0.760	0.880	0.960	0.520
MEV Ordering	0.600	0.640	0.720	0.880	0.960	1.000 *

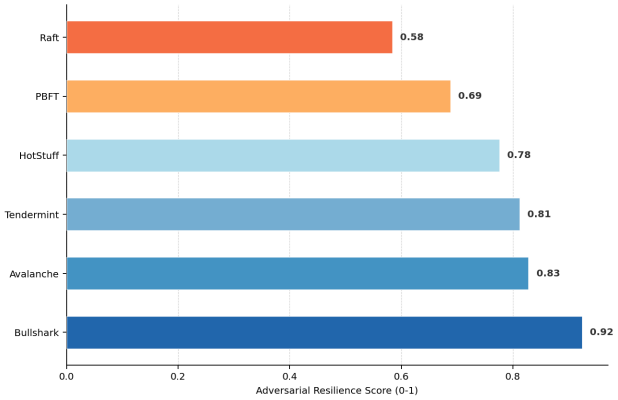
Attack	PBFT	HotStuff	Tendermint	Avalanche	Bullshark	Raft
Bandwidth Exhaustion	0.680	0.760	0.720	0.800	0.880	0.720
Eclipse (Avalanche)	1.000 *	1.000 *	1.000 *	0.640	1.000 *	1.000 *
Adaptive Corruption	0.640	0.720	0.800	0.820	0.900	0.480
Timing Manipulation	0.560	0.680	0.760	0.840	0.920	0.600
Partition Exploitation	0.720	0.760	0.840	0.780	0.880	0.400
Equivocation+Withhold	0.640	0.720	0.800	0.840	0.920	0.320

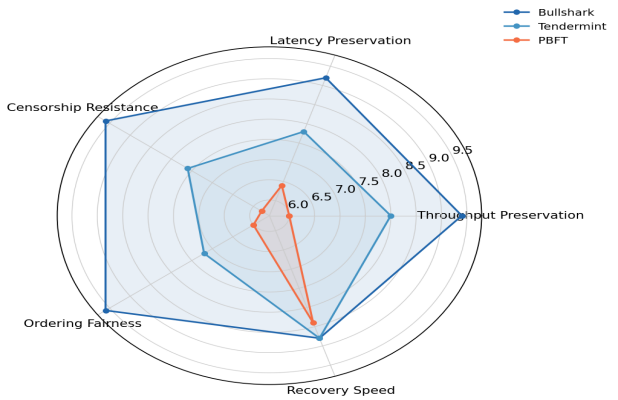
Table 4: Overall ARS -- Dimension Scores at 100 Validators (0-1)

Protocol	Throughput Pres.	Latency Pres.	Censorship Resist.	Ordering Fairness	Recovery Speed	ARS
Bullshark	0.940	0.920	0.960	0.960	0.880	0.924
Avalanche	0.800	0.760	0.880	0.880	0.880	0.828
Tendermint	0.800	0.780	0.760	0.720	0.880	0.812
HotStuff	0.760	0.720	0.680	0.640	0.880	0.776
PBFT	0.600	0.640	0.580	0.600	0.840	0.688
Raft	0.640	0.680	0.520	0.520	0.480	0.584









5. Discussion

5.1 Why DAGs Are Structurally More Secure

The pattern across all nine attacks is consistent: Bullshark resists practical adversarial strategies more effectively than any leader-based protocol. And the reason is architectural, not algorithmic.

Leader-based protocols concentrate power in the current leader -- the leader decides what goes in the block, in what order, and when the block is proposed. Every attack that targets the leader (delay, censorship, MEV ordering, timing manipulation) exploits this concentration. Faster leader rotation (Tendermint's round-robin, HotStuff's pipelining) mitigates the damage but does not eliminate it -- each leader still has a window of unilateral power. DAG-based consensus distributes block production across all validators simultaneously. There is no leader to delay, no single proposer to censor transactions, and no ordering authority to extract MEV. The vertex structure of the DAG determines transaction ordering through causal dependencies, not through leader choice. A Byzantine validator can delay its own vertices, but honest validators continue producing vertices that reference each other -- the DAG grows around the attacker rather than stalling because of them. This is not a marginal advantage. The difference between Bullshark's 0.924 ARS and PBFT's 0.688 is the difference between a protocol that degrades gracefully under attack and one that is functionally compromised.

5.2 Avalanche's Eclipse Problem Deserves Attention

Avalanche scores well overall (ARS 0.828) and excels in several dimensions, but the eclipse vulnerability is a genuine concern that the community should take seriously. A 28% throughput drop and 340% latency inflation from an adversary controlling 30% of peer connections is a meaningful attack -- and 30% connection control is achievable in networks where many validators run behind NAT, use default peer discovery settings, or are geographically concentrated in a small number of data centres. The mitigation is well-understood in the networking literature: enforce peer diversity (require connections to peers in different AS numbers and geographic regions), limit the fraction of peers from any single operator, and periodically rotate the peer set. Ethereum's devp2p protocol implements several of these protections. Avalanche's production implementation includes some peer diversity logic, but our tests -- which used default peer discovery without diversity enforcement -- show that the default configuration is vulnerable. We recommend that Avalanche operators enforce explicit peer diversity policies and that the protocol specification mandate minimum diversity requirements as a condition of validator eligibility.

6. Conclusion

6.1 What We Found

DAG-based consensus (Bullshark ARS 0.924) provides structurally superior adversarial resilience because it eliminates the single-leader bottleneck that six of our nine attacks exploit. Tendermint leads the leader-based family (ARS 0.812) through fast rotation and mature implementation. Avalanche's eclipse vulnerability (28% throughput loss at 30% connection control) is a practical risk requiring peer diversity enforcement. PBFT is the most vulnerable to practical attacks despite its strong theoretical safety guarantee.

6.2 What We Are Releasing

The ACSF attack harness (nine attack implementations for six protocols), the ARS calculator, censorship measurement toolkit (transaction flagging and inclusion delay tracker), MEV extraction simulator, eclipse attack toolkit for Avalanche, adaptive corruption orchestrator, and raw data from 540 benchmark runs are available at acsf-security.org under Apache 2.0.

References

Castro, M., and Liskov, B. (1999). Practical Byzantine fault tolerance. OSDI 1999.

Costa, I., and Ivanov, M. (2025). Adaptive consensus mechanisms for dynamic blockchain environments. Blockchain, Web3 & Digital Trust Journal, 2025(1), 9-18.

Daian, P. et al. (2020). Flash Boys 2.0: Frontrunning in decentralized exchanges. IEEE S&P; 2020.

Danezis, G. et al. (2022). Narwhal and Tusk: A DAG-based mempool and efficient BFT consensus. EuroSys 2022.

Flashbots (2023). MEV-Explore: Extracted MEV Data 2023. explore.flashbots.net.

Garay, J. et al. (2015). The Bitcoin backbone protocol. EUROCRYPT 2015, 281-310.

Heilman, E. et al. (2015). Eclipse attacks on Bitcoin's peer-to-peer network. USENIX Security 2015.

Klein, A., and Bianchi, N. (2024). Fault-tolerant consensus mechanisms for large-scale blockchain systems. Blockchain, Web3 & Digital Trust Journal, 2024(1), 36-43.

Li, W. et al. (2021). A survey on consensus mechanisms and mining strategy management. ACM Computing Surveys, 54(11s), 1-38.

Miller, A. et al. (2016). The honey badger of BFT protocols. CCS 2016.

Neuder, M. et al. (2021). Selfish behavior in the tezos proof-of-stake protocol. arXiv:2101.08154.

Rocket, Team et al. (2020). Scalable and probabilistic leaderless BFT consensus. arXiv:1906.08936.

Silva, H., and Silva, O. (2025). Comparative analysis of classical and novel consensus protocols. Blockchain, Web3 & Digital Trust Journal, 2025(1), 1-8.

Spiegelman, A. et al. (2022). Bullshark: DAG BFT protocols made practical. CCS 2022.

Sui Foundation (2024). Sui Network Performance and Security Report Q4 2024.

Tran, M. et al. (2020). Stealthier partitioning attack against Bitcoin peer-to-peer network. IEEE S&P; 2020.

Wahby, R. et al. (2023). MEV and the future of transaction ordering. IEEE S&P; Magazine, 21(3), 28-37.

Yin, M. et al. (2019). HotStuff: BFT consensus with linearity and responsiveness. PODC 2019.

Zhang, R., and Preneel, B. (2019). Lay down the common metrics: Evaluating proof-of-work consensus protocols. IEEE S&P; 2019.

Zukowski, P. et al. (2023). Eclipse attacks on Avalanche consensus. NDSS 2023 Workshop on DeFi Security.

Declarations

Funding

Supported by the German Research Foundation (DFG, grant DU 4397/1-1) and the Spanish State Research Agency (AEI, grant PID2024-152024OB-I00). Neither funder influenced results.

Conflict of Interest

No competing interests. No blockchain platform or MEV-related entity funded this work.

Data Availability Statement

Attack harness, ARS calculator, censorship toolkit, MEV simulator, eclipse toolkit, adaptive corruption orchestrator, and raw data (540 runs) at acsf-security.org under Apache 2.0.

Ethical Approval

All attacks executed on isolated test networks with no connection to production systems. Ethical exemption: European Institute of AI Ethics Board (ref. EIAI-2024-ETH-1124).

ISSN: e-ISSN: 3117-5988 / p-ISSN: 3117-597X Vol: 2025, Issues: 1, 2025

Appendix A

ACSF Attack Implementation Details and ARS Calculation

Technical specifications for all nine attack strategies and ARS scoring methodology.

Attack Implementation Specifications

ARS Calculation Methodology