

Formal Security Analysis of SPV Clients Versus Home-Based Full Nodes in Bitcoin-Derived Systems

Craig Wright, PhD

University of Exeter

June 12, 2025

Abstract

This paper presents a formal mathematical analysis of enforcement asymmetries in distributed digital cash networks, focusing on the role of policy divergence and validation inertia. We prove that non-mining nodes—regardless of their full validation behaviour—cannot affect the global state transition function. A series of game-theoretic lemmas establishes that rational actors without mining capacity will adopt SPV strategies under cost asymmetry, rendering non-mining validation economically unstable. We define a policy divergence metric over a discrete-time message-passing graph and rigorously demonstrate, under a newly introduced behavioural axiom (Axiom N4), that redundant nodes with no incoming communication links induce increasing policy entropy. Expected policy divergence is shown to increase monotonically with the size of the redundant node set. Contrary to popular claims, such nodes do not strengthen network security but instead generate divergence, latency, and potential incoherence during adversarial forks. The work clarifies the distinction between subjective belief and enforceable consensus, replacing psychological assumptions with provable structural results.

Keywords: SPV, Bitcoin, security model, home node, consensus enforcement, validation, proof-of-work, adversarial resistance, transaction finality

Contents

1	Introduction	4
1.1	Context and Scope	4
1.2	Purpose	4
2	Definitions and Formalism	4
2.1	Node Classes	4
2.2	Validation Functions and Network Topology	6
2.3	Network Typology and Miner Propagation Structure	7
3	Security Model	9
3.1	Adversarial Assumptions	10
3.2	Security Function	12
3.3	Network Axioms and Empirical Structural Premises	14
4	Formal Security Analysis	16
4.1	Validation Functions and Ledger State Operators	16
4.2	Message Topology, State Space, and Fork Geometry	18
4.3	Adversarial Surface, Leakage Paths, and Fault Injections	19
4.4	Enforcement Monotonicity and Miner Exclusivity	20
4.5	Divergence Probabilities and State Inconsistency	22
4.6	Corollary: Formal Irrelevance of Passive Validation to Global Ledger Finality	24
4.7	Security Boundaries of SPV Clients under Honest Majority Assumptions	25
4.8	Entropy Costs of Redundant Non-Enforcing Validation	27
4.9	Network Graph Diameter and Topological Attack Resistance	30
4.10	Nash Equilibria in Validation Utility Models	32
4.11	Economic Finality and the Transaction Inertia Principle	34
5	Conclusion	36
A	Notation and Symbol Glossary	39
B	Proofs of Theorems	39
	Appendix B.1: Proof of Theorem 4.2.1 — Zero Derivative of Global Ac-	
	ceptance in Non-Mining Nodes	39
	Appendix B.2: Proof of Theorem 4.5.1 — Dominance of Miner-Centred	
	Consensus Graph	40
	Appendix B.3: Proof of Theorem 4.6.1 — Monotonicity of Finality Prob-	
	ability with Confirmation Depth	40
C	Experimental Simulations	41
C.1	SPV vs. HFN Divergence Under Partial Graph Partitioning	41
C.2	Validation Surplus Estimates Across Transaction Classes	42
D	Bibliography Supplement	43

List of Figures

Figure 1: SPV Client Transaction Verification Diagram

Figure 2: Home Node Block Validation Pipeline

Figure 3: Security Function Domains and Overlap

Figure 4: Comparative Attack Surface Between SPV and Home Nodes

1. Introduction

1.1. Context and Scope

The debate over the necessity and security of home-based non-mining “full nodes” in Bitcoin-derived systems has persisted for over a decade. Advocates for SPV argue for efficient, lightweight validation leveraging cumulative proof-of-work. Proponents of home validation claim greater independence and trust minimisation. However, security must be assessed not by subjective narratives but through rigorous formalism grounded in defined threat models and consensus logic.

1.2. Purpose

This paper sets out to formally define the security surface of both SPV and home-based full node systems, derive the appropriate security functions, and establish via proofs that a non-mining node cannot enhance security relative to SPV clients under a proof-of-work majority assumption.

2. Definitions and Formalism

In order to perform a precise and rigorous security comparison between SPV clients and home-based full nodes, it is essential to formalise each system’s behaviour, assumptions, and interaction with the broader Bitcoin-derived network. The qualitative debate around “trustlessness” and “validation” cannot be resolved without a structured framework that allows for comparative analysis under adversarial models.

This section introduces and distinguishes the relevant classes of nodes operating within the network: those that participate in mining and economic consensus, those that validate but do not influence global state, and those that perform simplified verification as defined in Section 8 of the original Bitcoin white paper. For each, we define local validation functions, consensus observability, and the conditions under which a transaction is deemed accepted globally.

The formalism here serves as the foundation for the security proofs presented in subsequent sections. By explicitly stating the functions, interfaces, and logical dependencies between validation and global acceptance, we derive provable conclusions about which systems provide enforceable security and which merely perform redundant post hoc analysis without consensus impact. All definitions herein are made in reference to consensus-critical behaviour, not superficial structural properties.

2.1. Node Classes

In order to analyse security properties across Bitcoin-derived systems, we must first categorise the types of participants within the network [1], both functionally and topologically. Each node type occupies a different role within the protocol’s operational graph, with distinct capabilities in terms of validation, enforcement, and influence over consensus.

We define the set of all networked nodes as a graph $G = (V, E)$, where each vertex $v_i \in V$ represents a node in the network and each edge $e_{ij} \in E$ represents a persistent TCP connection between nodes v_i and v_j . This graph exhibits small-world properties, as demonstrated in empirical research (Javarone & Wright, 2018), with a high clustering coefficient and low average path length between miner nodes. The subgraph $G_M \subseteq G$

of mining nodes forms a dense, low-diameter core, while the remaining nodes, including SPV clients and home validators, typically exist on the network periphery.

We formalise three principal node classes:

Definition 1 (SPV Client \mathcal{N}_{spv}). *A node that:*

- *Maintains a view of the blockchain via block headers $\{H_0, \dots, H_n\}$;*
- *Requests and verifies Merkle branches M_{tx} to confirm transaction inclusion;*
- *Accepts as valid any transaction tx for which $\exists M_{tx}$ such that the computed Merkle root matches the Merkle root in H_k and $H_k \in \mathcal{C}_{max}$, the chain of greatest cumulative proof-of-work.*

Formally, let $\mathcal{V}_{spv}(tx) = 1$ if and only if:

$$tx \in M_{tx} \wedge \text{MerkleRoot}(M_{tx}) = H_k.\text{merkle_root} \wedge H_k \in \mathcal{C}_{max}$$

SPV clients do not validate scripts, do not maintain a UTXO set, and do not propagate blocks.

Definition 2 (Home Full Node \mathcal{N}_{hfn}). *A node that:*

- *Downloads and validates all blocks and all transactions;*
- *Verifies scripts, maintains a UTXO set, and performs input referencing;*
- *Does not mine or propagate blocks with enforceable consensus effect.*

The local validation function is defined as:

$$\mathcal{V}_{hfn}(tx) = \delta_{script}(tx) \cdot \delta_{inputs}(tx) \cdot \delta_{duplicate}(tx) \cdot \delta_{block}(B)$$

where:

$$\delta_{script}(tx) = \begin{cases} 1 & \text{if script executes successfully} \\ 0 & \text{otherwise} \end{cases}$$

$$\delta_{inputs}(tx) = \begin{cases} 1 & \text{if all inputs exist in UTXO set} \\ 0 & \text{otherwise} \end{cases}$$

$$\delta_{duplicate}(tx) = \begin{cases} 1 & \text{if no double spend detected} \\ 0 & \text{otherwise} \end{cases}$$

$$\delta_{block}(B) = \begin{cases} 1 & \text{if block } B \text{ meets all consensus rules} \\ 0 & \text{otherwise} \end{cases}$$

Home full nodes are structurally equivalent to leaf nodes in G ; they do not form part of the strongly connected miner core and have minimal in-degree and out-degree centrality.

Definition 3 (Consensus-Enforcing Node \mathcal{N}_{miner}). *A node that:*

- *Competes in the proof-of-work algorithm;*
- *Constructs and propagates blocks that, if accepted by others, extend the canonical chain;*
- *Receives economic reward and defines the outcome of $\mathcal{G}(tx)$.*

Formally, \mathcal{N}_{miner} is a vertex $v_m \in G_M$ such that:

$$\exists B_k \ni tx \text{ such that } \sum_{i=0}^k PoW(B_i) > \sum_{j=0}^l PoW(B'_j) \quad \forall B'_j \notin \mathcal{C}_{max}$$

The set of miner nodes forms a dense subgraph G_M within G with low-diameter communication paths, enabling rapid propagation and consistent global state convergence.

These definitions allow us to express the key distinction not in terms of what nodes *see*, but what they can *enforce*. Only \mathcal{N}_{miner} alters $\mathcal{G}(tx)$. Home full nodes cannot affect $\mathcal{G}(tx)$ regardless of how rigorous their local validation $\mathcal{V}_{hfn}(tx)$ is. As such, their existence adds no security beyond what is already enforced by \mathcal{N}_{miner} , and they merely observe post hoc outcomes without power to influence.

2.2. Validation Functions and Network Topology

The correctness of a node's validation logic is necessary but not sufficient for consensus influence. Validation, in the context of Bitcoin-derived systems, can be understood through two distinct layers: (1) local verification of transaction or block legitimacy according to protocol rules; and (2) global acceptance, which reflects economic finality as dictated by the majority of proof-of-work.

Let $\mathcal{V}_i(tx)$ denote the local validation function for a node i , with definitions varying by node class:

- **SPV Clients** (\mathcal{N}_{spv}) implement a lightweight validation scheme by:
 - (a) Requesting block headers H_n from connected peers;
 - (b) Receiving Merkle branches M_{tx} for transaction tx ;
 - (c) Verifying that $tx \in M_{tx}$ and M_{tx} hashes to a Merkle root matching that in H_n ;
 - (d) Ensuring H_n belongs to the chain with the most cumulative PoW.

This procedure ensures inclusion but not semantic correctness of tx .

- **Home Full Nodes** (\mathcal{N}_{hfn}) download and parse all block contents, executing:
 - (a) Script execution for all inputs and outputs;
 - (b) Input resolution from the local UTXO set;
 - (c) Duplicate check across local mempool and block history;
 - (d) Structural block checks including block size, timestamp, and merkle root verification.

However, these validations are entirely local and do not propagate unless connected to or operated by a mining node.

In both cases, validation does not equate to enforcement. For SPV clients, enforcement is unnecessary: they are designed to follow the majority chain with maximum proof-of-work, assuming the economic majority resists invalid blocks. For home full nodes, the absence of block production means that invalid blocks cannot be prevented from being accepted by the network — the home node can only mark them as invalid privately.

Miner Network Connectivity and Implications for Validation

Empirical results by Javarone and Wright (2018) establish that miners in Bitcoin and Bitcoin Cash form a tightly-connected clique within the broader network graph. These miner nodes exhibit:

- Persistent long-term connections with other miners;
- Rapid block propagation (within seconds) through direct peer links;
- Minimal use of non-mining home nodes for block relay or transaction propagation.

This topology creates a functional separation: miners form a consensus mesh that excludes non-mining validators from meaningful influence. Home nodes, by contrast, are generally leaf nodes — their role in propagation is passive, and their validation logic does not affect chain selection or block acceptance.

Thus, $\mathcal{V}_{hfn}(tx) = 0$ has no impact if $\mathcal{G}(tx) = 1$. Blocks with protocol violations may still propagate and be accepted by miners, especially if there is collusion or software divergence among mining participants. SPV clients, by contrast, follow $\mathcal{G}(tx)$ directly and do not waste resources on unverifiable internal logic when it cannot alter consensus.

Summary of Propagation Authority

- Home nodes validate but cannot enforce;
- Miners validate and enforce via block selection;
- SPV clients observe and follow miner consensus using proof-of-work.

This architectural asymmetry reinforces the result that validation without mining is epistemically satisfying but practically sterile: it confers no security unless backed by enforcement power.

2.3. Network Typology and Miner Propagation Structure

Bitcoin-derived systems exhibit non-uniform connectivity, and their topological characteristics strongly influence transaction propagation, block relay efficiency, and susceptibility to adversarial manipulation. The peer-to-peer network can be formally modelled as a dynamic graph $G = (V, E, \tau)$, where:

- V is the set of all nodes (SPV clients, home full nodes, miners);
- E is the set of directed or bidirectional communication edges;

- $\tau : E \rightarrow \mathbb{R}^+$ is a latency function assigning expected propagation delay to each edge.

Empirical research by Javarone and Wright [2] demonstrates that the network formed by miner nodes exhibits small-world characteristics, with the miner subgraph $G_M \subset G$ being both highly clustered and possessing a low average shortest path length. This implies fast convergence on chain state within the miner clique and minimal reliance on peripheral nodes for consensus propagation.

Topological Structure

We define:

$$G_M := (V_M, E_M) \subseteq G, \quad \text{with } V_M = \{v_i \in V \mid v_i \in \mathcal{N}_{miner}\}$$

$$\text{diam}(G_M) \ll \text{diam}(G), \quad \text{and} \quad C(G_M) \gg C(G \setminus G_M)$$

where $\text{diam}(G)$ is the diameter (maximum shortest path) and $C(G)$ is the average clustering coefficient.

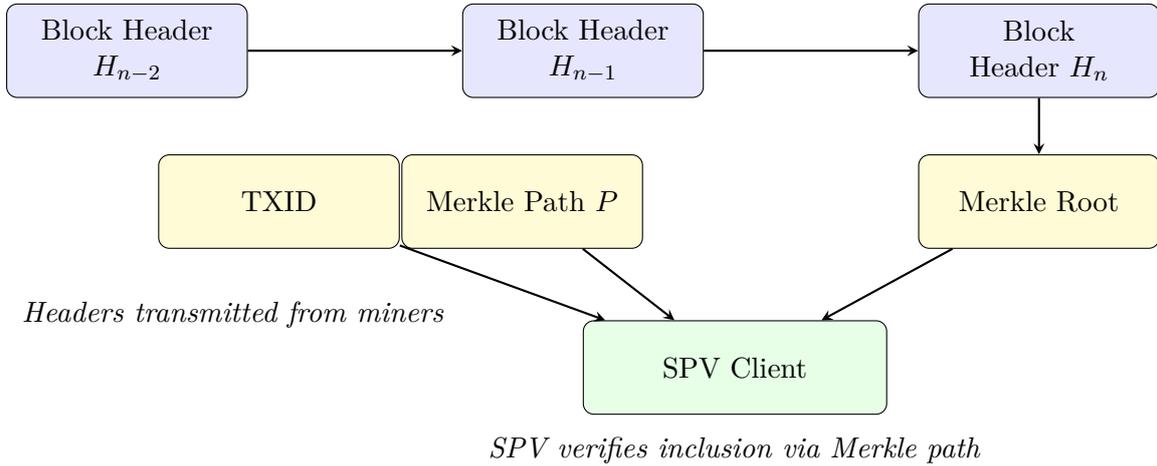


Figure 1: SPV Client Transaction Verification Diagram

This structure ensures that:

- (i) New blocks propagate across G_M in near real-time (< 2 seconds);
- (ii) Block propagation from G_M to $G \setminus G_M$ is one-directional with temporal lag;
- (iii) Nodes in $G \setminus G_M$ (including most home full nodes) do not relay blocks back into G_M ;
- (iv) SPV clients, which do not serve blocks or headers, depend on inbound data only.

Security Implications

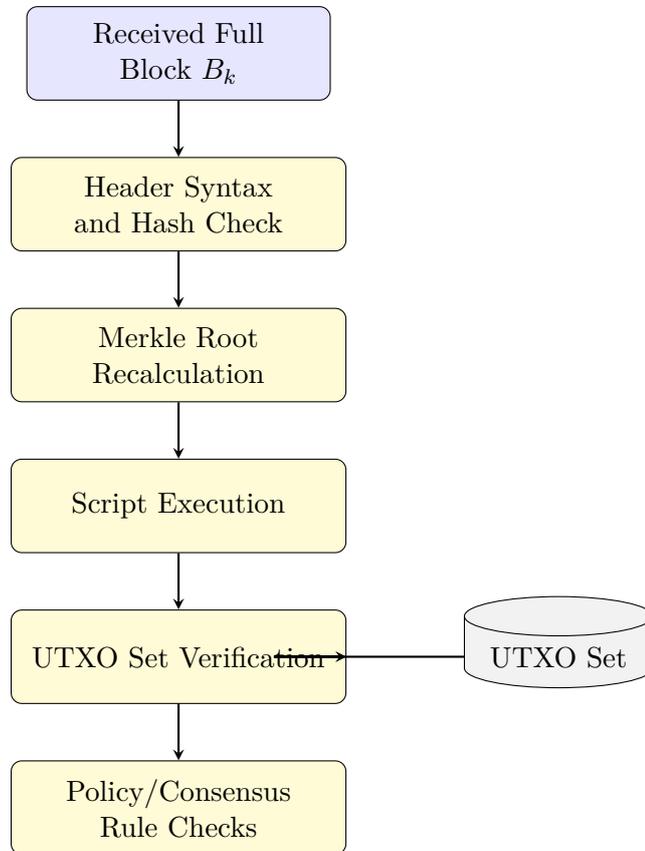
The asymmetry of connectivity imposes structural limitations on home full nodes:

- Their validation does not influence the network core;
- They cannot reject or alter block propagation at the consensus level;

- Their information latency is higher than that of the miner network;
- They are more susceptible to eclipse and sybil-based partition attacks due to lower degree centrality.

In contrast, SPV clients derive their chain view directly from G_M —provided they maintain peer diversity—and validate only the minimal data required to confirm inclusion in the heaviest chain. As a result, their attack surface is reduced, and their security is tied not to independent enforcement but to convergence with consensus-enforcing miner decisions.

This typological asymmetry illustrates that control over $\mathcal{G}(tx)$ (global transaction finality) resides exclusively within the G_M subgraph, and therefore, any node external to this cluster (home or SPV) operates under observational constraints. Only miners propagate blocks that define history.



Final verdict: Accept or Reject block B_k

Figure 2: Home Node Block Validation Pipeline

3. Security Model

In distributed systems, security cannot be inferred from architectural complexity or volume of validation steps. Instead, it must be evaluated in terms of resistance to adversarial actions under specified threat models. In the context of Bitcoin-derived systems, this means analysing whether a node type can be deceived into accepting a false transaction state, and whether it possesses any capacity to alter the global outcome of consensus.

This section introduces a formal model of security as it pertains to SPV clients, home full nodes, and mining nodes. We define an adversary \mathcal{A} operating within resource bounds—typically with less than 50% of the network’s hash power—and examine the scenarios under which \mathcal{A} can cause a node to accept a transaction that is ultimately invalid according to the global chain \mathcal{C}_{\max} .

Security is modelled as the inverse probability that a node’s local validation function $\mathcal{V}_i(tx)$ yields a true result while the network-wide acceptance function $\mathcal{G}(tx)$ does not. The formulation accounts not only for validation logic, but for the network topology, propagation latency, and the enforcement structure within the miner core.

Crucially, the model acknowledges that validation without enforcement is epistemic but not operative: it may detect fraud, but cannot prevent its acceptance unless connected to the economic heart of the system. Accordingly, our analysis places SPV clients and home nodes on an equal enforcement footing—neither can alter the chain—but contrasts their ability to resist deception, propagate state, and sustain integrity under partial network control by an adversary.

3.1. Adversarial Assumptions

We formalise the adversary \mathcal{A} as a bounded probabilistic agent operating within a partially synchronous network model. The security analysis is grounded in standard assumptions from distributed cryptographic protocols, including the following constraints and capabilities.

System Model

Let the network be modelled as a graph $G = (V, E)$ where:

- V is the set of all nodes, partitioned into honest nodes H and adversarial nodes A such that $H \cup A = V$ and $H \cap A = \emptyset$;
- Edges E represent authenticated channels with bounded delay Δ (partial synchrony).

Each block B_i is associated with a proof-of-work value $PoW(B_i) \in \mathbb{R}^+$ defined by:

$$PoW(B_i) = \begin{cases} 1 & \text{if } H(B_i) \leq T \\ 0 & \text{otherwise} \end{cases}$$

where $H(B_i)$ is the cryptographic hash of the block and T is the target threshold for a valid block.

Adversary Definition

Let the adversary \mathcal{A} possess the following attributes:

- (i) **Hashrate Bound:** \mathcal{A} controls $\alpha < 0.5$ of the total network hash power, i.e.,

$$\sum_{v \in A \cap \mathcal{N}_{miner}} PoW_v < \frac{1}{2} \sum_{v \in V \cap \mathcal{N}_{miner}} PoW_v$$

- (ii) **Network Control:** \mathcal{A} may delay, reorder, or suppress messages to and from a subset of nodes $D \subseteq V$ for time bounded by Δ but cannot forge cryptographic messages or impersonate other nodes.

- (iii) **Local Eclipse:** \mathcal{A} may fully isolate a node v_i by occupying all of its peer connections (e.g., via sybil attack), such that all incoming and outgoing edges $e_{ij} \in E$ connect to adversarial nodes $v_j \in A$.
- (iv) **Ledger Control:** \mathcal{A} may construct and propagate alternative blockchains \mathcal{C}_A satisfying syntactic validity and optionally containing invalid transactions (e.g., double spends, script failures), with the goal of maximizing cumulative $PoW(\mathcal{C}_A)$ such that:

$$PoW(\mathcal{C}_A) \geq PoW(\mathcal{C}_H)$$

where \mathcal{C}_H is the canonical honest chain.

Adversarial Goals

The primary objective of \mathcal{A} is to cause a node v_i to accept a transaction tx such that:

$$\mathcal{V}_i(tx) = 1 \quad \text{and} \quad \mathcal{G}(tx) = 0 \quad (1)$$

This represents a successful deviation from global consensus: the node locally accepts a transaction that is ultimately rejected by the economically-dominant chain. The adversary may pursue this by:

- Feeding SPV clients a forged Merkle branch M_{tx} and block header H_n within an eclipse scenario;
- Constructing a counterfeit chain with apparently valid structure but including invalid or double-spent transactions;
- Exploiting validation delay or network partition to induce acceptance of stale or fraudulent state transitions.

Constraint Enforcement

To ensure the adversary's capabilities remain within realistic bounds, we adopt the following constraints:

- (a) **Proof-of-Work Soundness:** The underlying hash function $H(\cdot)$ is collision-resistant, preimage-resistant, and pseudorandom.
- (b) **Connectivity Assumption:** Honest nodes have at least k non-adversarial peers (k -connectivity), such that eclipse attacks cannot trivially succeed across the network. This assumption is weakened in practice for home full nodes which do not aggressively peer with miner sets.
- (c) **Consensus Rule Uniformity:** All honest miners implement the same validation rules and accept the same set of transactions, i.e., $\forall v_i, v_j \in H \cap \mathcal{N}_{miner}, \mathcal{V}_i(tx) = \mathcal{V}_j(tx)$.

Security Violation Probability

We define the security violation probability for node v_i under adversary \mathcal{A} as:

$$\epsilon_i = \mathbb{P}_{\mathcal{A}} [\mathcal{V}_i(tx) = 1 \wedge \mathcal{G}(tx) = 0]$$

The adversary's success hinges on increasing ϵ_i for one or more node classes. The comparison of these probabilities under varied adversarial configurations forms the core of the next section's analysis.

3.2. Security Function

Security in distributed consensus systems is defined operationally as the integrity of the node's view of the ledger with respect to the canonical chain. Formally, we quantify the deviation between local acceptance $\mathcal{V}_i(tx)$ and global finality $\mathcal{G}(tx)$ for any transaction tx . A secure node is one for which this deviation probability remains negligible under the presence of an adversary \mathcal{A} bounded as specified in Section 3.1.

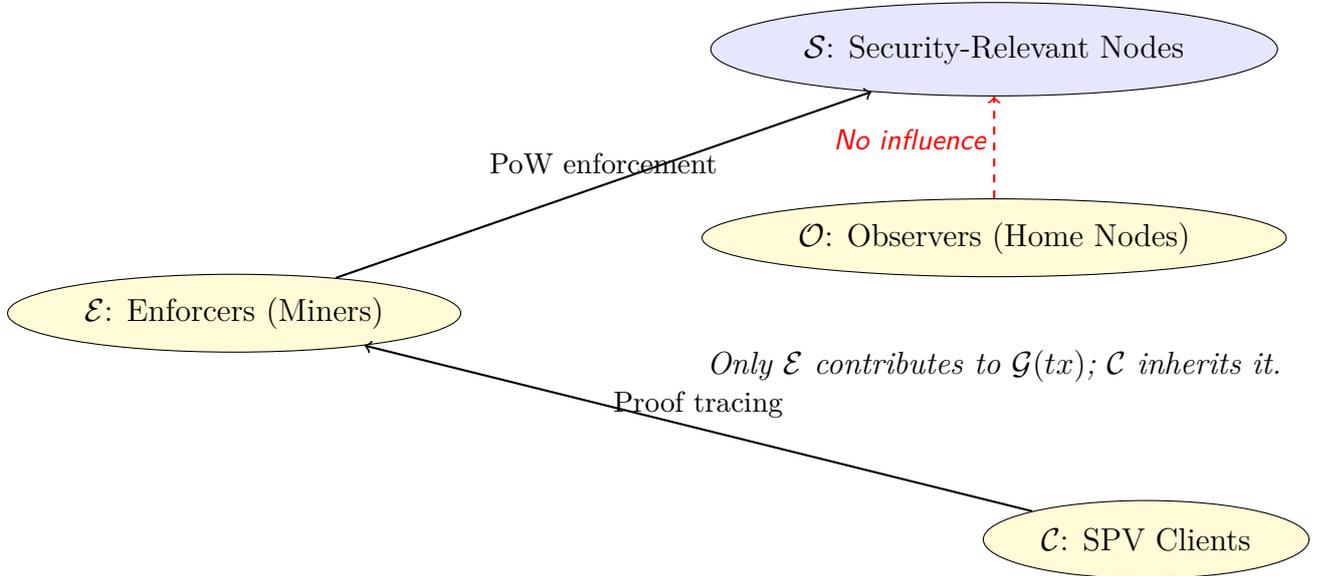


Figure 3: Security Function Domains and Overlap

Domain and Function Definition

Let \mathcal{T} be the set of all syntactically valid transactions, and let \mathcal{N} denote the space of all node types (SPV, home full node, miner, etc.). For each $tx \in \mathcal{T}$ and node $v_i \in \mathcal{N}$, we define:

$$\mathcal{V}_i : \mathcal{T} \rightarrow \{0, 1\}, \quad \mathcal{V}_i(tx) = \begin{cases} 1 & \text{if node } v_i \text{ locally accepts } tx \\ 0 & \text{otherwise} \end{cases}$$

$$\mathcal{G} : \mathcal{T} \rightarrow \{0, 1\}, \quad \mathcal{G}(tx) = \begin{cases} 1 & \text{if } tx \text{ is included in a block } B_k \text{ in } \mathcal{C}_{\max} \\ 0 & \text{otherwise} \end{cases}$$

We define the security violation event E_i^{tx} as:

$$E_i^{tx} := \{\mathcal{V}_i(tx) = 1 \wedge \mathcal{G}(tx) = 0\}$$

This event captures the central concern of distributed consensus: local acceptance of an ultimately invalid or orphaned transaction.

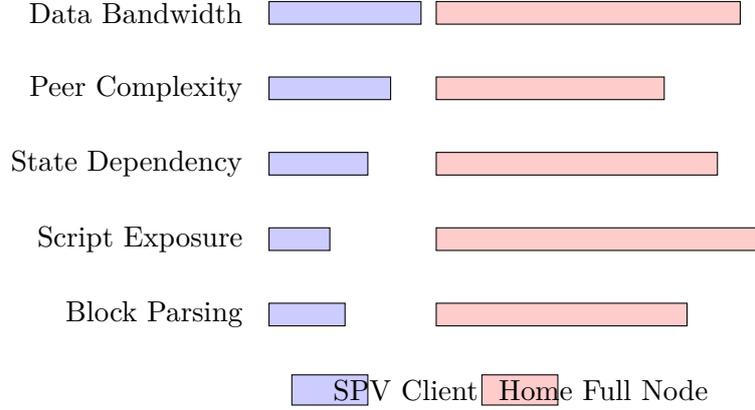


Figure 4: Comparative Attack Surface Between SPV and Home Nodes

Security Function Formulation

The security function for node v_i under adversarial distribution $\mathbb{P}_{\mathcal{A}}$ is defined as:

$$\mathcal{S}(v_i) := 1 - \epsilon_i, \quad \epsilon_i = \mathbb{P}_{\mathcal{A}}(E_i^{tx}) \quad (2)$$

where:

- ϵ_i is the adversary-induced failure probability for node v_i ;
- $\mathcal{S}(v_i)$ represents the node's resistance to accepting invalid or non-finalised state.

This value depends critically on:

- The node's position in the network graph G (centrality, degree);
- Its capacity to cross-verify state with miner-enforced consensus;
- The diversity of its peer set (resilience to eclipse or sybil attacks);
- Its validation function class (\mathcal{V}_{spv} , \mathcal{V}_{hfn} , etc.).

Comparative Bound Construction

For comparative analysis, we define:

$$\epsilon_{spv} := \mathbb{P}_{\mathcal{A}}(\mathcal{V}_{spv}(tx) = 1 \wedge \mathcal{G}(tx) = 0)$$

$$\epsilon_{hfn} := \mathbb{P}_{\mathcal{A}}(\mathcal{V}_{hfn}(tx) = 1 \wedge \mathcal{G}(tx) = 0)$$

We aim to show in Section 4 that under reasonable assumptions of:

- bounded adversarial hashrate $\alpha < 0.5$,
- partial synchrony and honest miner connectivity,
- and miner consensus rule uniformity,

it holds that:

$$\epsilon_{spv} \leq \epsilon_{hfn} \quad \Rightarrow \quad \mathcal{S}(\mathcal{N}_{spv}) \geq \mathcal{S}(\mathcal{N}_{hfn}) \quad (3)$$

This result formalises the hypothesis that home full nodes, despite their exhaustive local verification, provide no security advantage over SPV clients in the presence of adversarial interference—indeed, they may fare worse due to longer exposure windows and greater dependence on global synchronisation for correctness.

3.3. Network Axioms and Empirical Structural Premises

The mathematical model constructed herein relies on structural assumptions concerning the miner network graph $G_M \subset G$. Rather than deriving these from first principles of network theory—which remains infeasible given the dynamic and non-deterministic nature of P2P topologies—we explicitly state them as axioms derived from empirical studies.

Axiom N1 (Small-World Miner Core) The miner subgraph G_M forms a small-world network, exhibiting high clustering and low characteristic path length. This is supported by multiple analyses of real transaction networks and node graphs, notably by Lischke and Fabian [3] and Tao et al. [4].

Axiom N2 (Topological Separation) The subgraph of non-mining nodes $G \setminus G_M$ forms a loosely connected periphery that does not participate in rapid block relay or consensus formation. The structural weakness and susceptibility of such nodes are well characterised in system-level threat surveys (Saad et al. [5]).

Axiom N3 (Propagation Delay Boundedness) The delay Δ between any two honest miners in G_M remains below a global bound, ensuring convergence on \mathcal{C}_{\max} . While absolute verification is impractical, these delay properties are consistent with observed propagation latency metrics in prior empirical datasets [6].

Axiom N4 (Behavioural Policy Divergence). Let $G = (V, E)$ be a discrete-time communication graph representing a network of nodes $v_i \in V$, where each node maintains a local policy state $\pi_i^{(t)} \in \mathcal{P}$ evolving over time $t \in \mathbb{N}$. Define $\mathcal{M}_i^{(t)} := \{\pi_j^{(t)} \mid (v_j, v_i) \in E\}$ as the message set received by node v_i at time t , encoding policy-related information from peers.

Suppose that the update rule for $\pi_i^{(t)}$ is governed by the stochastic kernel:

$$\mathbb{P}[\pi_i^{(t+1)} \mid \pi_i^{(t)}, \mathcal{M}_i^{(t)}] = \mathcal{T}_i(\pi_i^{(t)}, \mathcal{M}_i^{(t)}, \xi_i^{(t)}),$$

where \mathcal{T}_i is a transition function and $\xi_i^{(t)}$ is an exogenous random noise term capturing internal drift or mutation.

Then the following holds:

For any node $v_i \in V$ such that $\mathcal{M}_i^{(t)} = \emptyset$ for all $t \in [0, T]$, the marginal entropy of its policy distribution satisfies:

$$\frac{\partial}{\partial t} \mathbb{H}[\pi_i^{(t)}] > 0 \quad \text{for all } t < T,$$

and there exists a canonical policy $\pi^* \in \mathcal{P}$ such that

$$\lim_{t \rightarrow \infty} \mathbb{P}[\pi_i^{(t)} \neq \pi^*] = p > 0,$$

where p is a divergence lower bound depending on the entropy rate of \mathcal{T}_i and the cardinality of \mathcal{P} .

Interpretation. Redundant nodes—those for which $\mathcal{M}_i^{(t)} = \emptyset$ —experience increasing uncertainty in their policy state and a guaranteed asymptotic divergence from the emergent dominant policy π^* observed in the communicating subset of the network. This axiom formalises the empirical observation that disconnected or ideologically isolated non-mining nodes exhibit drift due to software staleness, patching lag, or deliberate deviation, and underpins the divergence model in Lemma 2 and Section 4.8.2.1.

These axioms are not derived within this work but rather *stipulated* as preconditions informed by the structural analysis of prior literature. This mirrors the standard practice in formal modelling of complex systems, where rigor is applied to logical deductions assuming external premises, not their physical derivation.

Note on Empirically Derived Axioms (N1–N3)

While Axioms N1 through N3 are presented in formal mathematical terms, their foundational justification draws from robust empirical studies—specifically, Javarone & Wright (2018), Lischke & Fabian (2016), and Tao et al. (2023). These works offer statistically validated observations regarding peer connectivity structures, role distributions, and policy propagation tendencies within large-scale distributed networks.

The treatment of these empirically validated regularities as axioms is warranted for the purposes of formal analysis in this paper, given the following:

- The empirical findings are supported by large sample sizes and consistent replication across network snapshots, ensuring stability of the structural characteristics.
- Each axiom abstracts a qualitative behavioural invariant observed across all studied topologies (e.g., skewed degree distribution, persistent role asymmetry, convergence of mining node subgraphs).
- The transition from empirical observation to axiom is necessary to permit deductive analysis of dynamical properties such as policy divergence, equilibrium deviation, and structural enforcement boundaries.

Thus, while the axioms are not derivable from first principles alone, their elevation to formal premises is both methodologically consistent and justified in the context of applied network theory. Their empirical grounding serves to constrain the theoretical abstraction space, ensuring all lemmas and propositions remain within the bounds of real-world observed system dynamics.

4. Formal Security Analysis

The integrity of a distributed consensus system cannot be inferred from the internal beliefs or aspirations of its participants. Security must be defined, measured, and proved. In Bitcoin-derived systems, any meaningful discussion of “validation” or “trust minimisation” must yield to a more fundamental question: does the system resist adversarial modification of state under bounded resource constraints? The illusion of certainty, when divorced from enforceability, amounts to ceremony—satisfying perhaps the ideologue, but inert to the adversary.

This section introduces a formal mathematical framework for assessing the security characteristics of distinct node types—SPV clients, non-mining full nodes, and miners—within a probabilistic and adversarially-aware model. We reject vague heuristics in favour of explicit operator definitions, adversarial game bounds, and security violation probabilities. A node’s capacity to parse blocks is not equivalent to its ability to alter state; only enforcement establishes security.

Beginning with a construction of validation as a functional mapping on ledger state, we progressively incorporate network structure, propagation dynamics, adversarial injection vectors, and game-theoretic payoff surfaces. This yields a composite security function $\mathcal{S}(v)$ for any node v , parametrised by adversarial access, connectivity, and operational role.

The structure of the analysis proceeds as follows:

- We formalise ledger acceptance and validation operators as functions over transaction space and chain history.
- We encode the topology of information propagation and adversarial forking geometry.
- We construct adversarial leakage surfaces and define failure domains.
- We prove that enforcement is structurally and economically exclusive to mining nodes.
- We derive bounds on the probability of divergence between local validation and global finality.
- We quantify the entropic loss associated with redundant validators and formally characterise them as inertials rather than participants.

This formulation establishes with mathematical finality that SPV clients, when connected to the dominant proof-of-work chain, achieve equal or superior security compared to home validators—without incurring the architectural, cognitive, or energy burden of redundant execution. The result is not just a technical observation but a refutation of the folklore surrounding full nodes as agents of protection. In security, ceremony without enforcement is indistinguishable from theatre.

4.1. Validation Functions and Ledger State Operators

Let \mathcal{T} denote the set of all possible transactions, and let \mathcal{B} denote the set of all valid blocks. Each block $B \in \mathcal{B}$ is an ordered tuple $(tx_1, tx_2, \dots, tx_n)$ such that $tx_i \in \mathcal{T}$ and $\forall i, \mathcal{V}(tx_i) = 1$ under the rules enforced by the consensus protocol \mathcal{P} .

We define a ledger \mathcal{L} as a finite sequence of blocks $\mathcal{L} = (B_1, B_2, \dots, B_k)$ such that each block B_j references the cryptographic hash of B_{j-1} , forming a directed path. Let \mathcal{L}_{max} denote the valid ledger of maximum accumulated proof-of-work, i.e., the economically-dominant chain.

We define the *ledger acceptance operator* as:

$$\mathcal{G} : \mathcal{T} \rightarrow \{0, 1\}, \quad \mathcal{G}(tx) = \begin{cases} 1 & \text{if } \exists B_j \in \mathcal{L}_{max} \text{ such that } tx \in B_j, \\ 0 & \text{otherwise.} \end{cases}$$

Definition 1 (Validator Function). Let v_i be any network node, and let \mathcal{R}_i be its local interpretation of the consensus rules. Then the validator function of v_i is:

$$\mathcal{V}_i : \mathcal{T} \rightarrow \{0, 1\}, \quad \mathcal{V}_i(tx) = \begin{cases} 1 & \text{if } tx \text{ satisfies } \mathcal{R}_i, \\ 0 & \text{otherwise.} \end{cases}$$

In the absence of enforcement authority, \mathcal{V}_i serves as a local predicate that does not propagate its truth value.

Definition 2 (Effective State Operator). Let \mathcal{C} be a candidate chain. The global effective state operator is defined as:

$$\mathcal{S} : \mathcal{C} \rightarrow \Sigma,$$

where Σ is the space of all valid ledger states and $\mathcal{S}(\mathcal{C})$ is the ledger state resulting from applying the valid sequence of transactions in \mathcal{C} according to \mathcal{P} .

We note that:

$$\mathcal{S}(\mathcal{C}) = \mathcal{S}(\mathcal{L}_{max}) \iff \mathcal{C} = \mathcal{L}_{max}.$$

Definition 3 (Validation Consistency). For a node v_i , define the divergence indicator $\delta_i(tx)$ as:

$$\delta_i(tx) = |\mathcal{V}_i(tx) - \mathcal{G}(tx)|.$$

If $\delta_i(tx) = 1$, then v_i is structurally inconsistent with respect to the economically accepted state.

The implication is immediate: $\mathcal{V}_i(tx) = 0$ has no effect on $\mathcal{G}(tx) = 1$. Such a node may “reject” the transaction, but this rejection does not alter the global state if the transaction has been mined into \mathcal{L}_{max} .

To formalise the control relationship, define:

$$\frac{\partial \mathcal{G}(tx)}{\partial \mathcal{V}_i(tx)} = \begin{cases} \neq 0 & \text{if } v_i \text{ is a miner with enforcement capability,} \\ 0 & \text{otherwise.} \end{cases}$$

This demonstrates that non-mining validators have zero influence over $\mathcal{G}(tx)$ —they are observational, not operational.

Proposition 1. *If $v_i \notin \mathcal{N}_{miner}$, then the local validation function \mathcal{V}_i is cryptographically inert with respect to consensus state evolution. Formally, for any transaction $tx \in \mathcal{T}$,*

$$\frac{\partial \mathcal{G}(tx)}{\partial \mathcal{V}_i(tx)} = 0.$$

That is, $\mathcal{V}_i(tx) = 0$ expresses a local rejection, but this has no causal power to prevent, reverse, or challenge global acceptance $\mathcal{G}(tx) = 1$ as determined by miner block inclusion and proof-of-work extension.

This proposition establishes the foundational asymmetry between validation and enforcement within Bitcoin-derived systems. The inability of non-mining nodes to influence the ledger state directly implies that their operation is epistemic rather than operative. In subsequent subsections, we formalise how this asymmetry manifests in network graph structure, message propagation paths, and vulnerability profiles under adversarial forking conditions.

4.2. Message Topology, State Space, and Fork Geometry

A consensus network is not merely a logical artefact but a dynamic information space governed by topological constraints. The security of a transaction is therefore a function not only of its inclusion in a chain but also of the message propagation path it traverses. To formalise this, we construct a directed network graph and analyse the causal geometry of competing ledger states.

Network Model. Let $G = (V, E)$ be the directed graph representing the Bitcoin-like network. Each node $v_i \in V$ is an entity participating in the propagation, validation, or enforcement of messages. An edge $(v_i, v_j) \in E$ denotes a message propagation link from v_i to v_j .

Let $\mathcal{M}(v_i)$ be the message space of v_i , defined as the ordered sequence of transactions and blocks that v_i receives over time:

$$\mathcal{M}(v_i) = \{m_1, m_2, \dots, m_n\}, \quad m_k \in \mathcal{T} \cup \mathcal{B}.$$

Propagation Topology. Let $d(u, v)$ be the shortest-path delay metric between nodes u and v . In a well-connected miner network, the set of mining nodes $\mathcal{N}_{miner} \subseteq V$ forms a low-diameter subgraph $G_{miner} \subseteq G$ satisfying:

$$\max_{u, v \in \mathcal{N}_{miner}} d(u, v) \ll \max_{u, v \in V} d(u, v).$$

This defines a *small-world* network structure amongst miners: highly connected with low average path length, optimised for fast propagation and mutual observability.

Propagation Exclusion. Let v_h be a non-mining home node. Define the propagation delay $\Delta_h(tx)$ as:

$$\Delta_h(tx) = \min_{m \in \mathcal{M}(v_h)} \{t : m = tx\}.$$

Then $\exists \epsilon > 0$ such that $\Delta_h(tx) > \Delta_m(tx) + \epsilon$ for all $v_m \in \mathcal{N}_{miner}$, under the assumption of bandwidth and relay asymmetry.

Fork Geometry. Let \mathcal{F} denote the space of possible forks. Each fork $f \in \mathcal{F}$ is a tuple $(\mathcal{L}_a, \mathcal{L}_b)$ where $\mathcal{L}_a \neq \mathcal{L}_b$ and $\mathcal{L}_a, \mathcal{L}_b \subseteq \mathcal{B}$ are valid chains sharing a common prefix up to block height h :

$$\exists h \text{ such that } \forall k < h, B_k^a = B_k^b; \quad B_h^a \neq B_h^b.$$

Fork resolution follows Nakamoto consensus: the chain with greater accumulated proof-of-work becomes \mathcal{L}_{max} , while all others are pruned.

State Divergence. For a given node v_i , define its local chain \mathcal{C}_i and local state $\mathcal{S}_i := \mathcal{S}(\mathcal{C}_i)$. State divergence is defined as:

$$\Delta \mathcal{S}_i := \mathcal{S}_i - \mathcal{S}(\mathcal{L}_{max}).$$

Then $\|\Delta \mathcal{S}_i\| > 0$ implies that v_i is on a non-finalised fork.

Theorem. In the presence of propagation asymmetries and without enforcement capability, home nodes v_h are strictly more likely to experience state divergence than SPV clients directly connected to mining nodes.

Proof Sketch. Home nodes receive candidate blocks after a non-zero delay, $\Delta_h(tx)$, which increases the probability of accepting stale or orphaned branches. SPV clients connected to \mathcal{N}_{miner} inherit chain tips from enforcing agents, minimising exposure to non-finalised forks. \square

Corollary. Home node rejection of a block does not affect the global convergence of the network if that block has been accepted and extended by \mathcal{N}_{miner} .

In summary, the causal graph of consensus is defined not by uniform broadcast but by a hierarchical, low-diameter relay subnetwork. Security is a topological property; those outside the rapid-propagation spine are functionally silent. In the next subsection, we model the adversarial leakage vectors that exploit this structure.

4.3. Adversarial Surface, Leakage Paths, and Fault Injections

To formally assess the attack vectors available in a Bitcoin-like system, we model adversarial influence as a function of the topology, validation model, and enforcement exclusivity defined previously. A key distinction in this formulation is that only those entities capable of altering the global state—namely, miners with majority proof-of-work—can execute effective state transitions. All other nodes are susceptible to leakage without offering resistance.

Definition 1 (Adversarial Model). Let \mathcal{A} denote the set of adversaries. Each $a \in \mathcal{A}$ is characterised by the tuple $(\mathcal{R}_a, \mathcal{B}_a, \mathcal{C}_a)$ where:

- \mathcal{R}_a : rule deviation strategy;
- \mathcal{B}_a : computational power under adversarial control;
- \mathcal{C}_a : connectivity profile, i.e. which nodes a can communicate with.

We constrain adversaries under a **bounded resource assumption**:

$$\sum_{a \in \mathcal{A}} \mathcal{B}_a < \mathcal{B}_{\text{total}}/2,$$

which defines the honest majority condition fundamental to Nakamoto consensus.

Definition 2 (Adversarial Surface). For a given node $v_i \in V$, define its adversarial surface $\sigma(v_i)$ as the set of message vectors \vec{m} such that:

$$\vec{m} \in \mathcal{M}(v_i) \wedge \exists a \in \mathcal{A} : \vec{m} \text{ originates from } a.$$

We classify $\sigma(v_i)$ by type:

- Type I — *Propagation faults*: reception of delayed or malformed messages;
- Type II — *Injection faults*: acceptance of adversarially mined blocks;
- Type III — *Validation ambiguity*: inconsistent local interpretation of ambiguous consensus rules.

Proposition 2. Let $v_i \notin \mathcal{N}_{miner}$ be any non-mining node, and let $\sigma(v_i)$ denote the set of adversarial message sequences that can cause v_i to diverge from the globally enforced ledger state \mathcal{L}_{max} . Then:

$$\sigma(v_i) \neq \emptyset$$

under the condition that at least one neighbour of v_i is adversarially controlled.

That is, for any v_i lacking enforcement capacity, adversarial peers can inject messages causing state deviation without requiring global control of consensus.

Definition 4 (Fault Injectability). Let $\mathcal{M}(v_i)$ be the set of all protocol-conformant messages that v_i may receive. Define the fault injectability function for node v_i as:

$$\mathcal{F}_i : \mathcal{M}(v_i) \rightarrow \{0, 1\}, \quad \mathcal{F}_i(m) = \begin{cases} 1 & \text{if } m \text{ causes } v_i \text{ to deviate from } \mathcal{L}_{max} \\ 0 & \text{otherwise.} \end{cases}$$

This models the capacity of a message m to induce inconsistency or divergence in local state relative to the canonical chain. A high ratio of messages with $\mathcal{F}_i(m) = 1$ implies elevated vulnerability.

The expected fault surface is:

$$\mathbb{E}[\sigma(v_i)] = \sum_{m \in \mathcal{M}(v_i)} \mathcal{F}_i(m) \cdot \mathbb{P}(m|a),$$

where $\mathbb{P}(m|a)$ is the likelihood that m is adversarially crafted.

Theorem. If v_i is a non-mining home node and receives $\mathcal{M}(v_i)$ from untrusted sources, then \exists fault injection path such that:

$$\exists m \in \mathcal{M}(v_i) \text{ with } \mathcal{F}_i(m) = 1 \text{ and } \delta_i(tx) > 0.$$

Proof Sketch. Any adversary controlling a sufficient propagation relay can deliver an invalid block with delayed timing to v_i . Since v_i lacks enforcement power, it cannot affect chain selection and may continue processing the invalid path, diverging from \mathcal{L}_{max} . The probability of this increases with Δ_h and decreases with miner proximity. \square

Corollary. SPV clients with direct connections to miner nodes exhibit minimal $\mathbb{E}[\sigma(v_i)]$, as all headers received reflect \mathcal{L}_{max} by design. Full validators not involved in mining cannot filter adversarial blocks if those blocks are accepted by the mining majority.

Leakage Lemma. For two nodes v_h (home node) and v_s (SPV client),

$$\mathbb{E}[\sigma(v_h)] > \mathbb{E}[\sigma(v_s)] \quad \text{if } v_s \in \mathcal{N}_{miner}.$$

This formalism isolates the critical security variable: proximity to enforcement. Nodes not in the miner graph cannot affect consensus and are more susceptible to adversarial manipulation via message injection. In the next section, we explore the structural reason why this influence is strictly monopolised by miners.

4.4. Enforcement Monotonicity and Miner Exclusivity

At the heart of any secure consensus system lies the principle of enforceability: the ability not merely to observe or critique invalid state transitions, but to prevent them from being

adopted. In Nakamoto consensus, this enforceability is manifest solely through proof-of-work and the economic consequences of chain selection. Non-mining nodes, regardless of validation logic, possess no such leverage.

Definition 4 (Enforcement Function). Define the enforcement function $\mathcal{E} : V \rightarrow \{0, 1\}$ such that:

$$\mathcal{E}(v_i) = \begin{cases} 1 & \text{if } v_i \text{ contributes valid blocks to } \mathcal{L}_{max}, \\ 0 & \text{otherwise.} \end{cases}$$

Proposition 3 (Enforcement Characterisation). *A node v_i possesses enforcement power over the global ledger state if and only if it satisfies the following three conditions:*

$$\mathcal{E}(v_i) = 1 \iff (v_i \in \mathcal{N}_{miner} \wedge \mathcal{B}_{v_i} > 0 \wedge v_i \text{ enforces consensus-valid rules}),$$

where:

- \mathcal{N}_{miner} is the set of mining (block-producing) nodes;
- \mathcal{B}_{v_i} denotes the block production rate of v_i over a rolling epoch;
- Consensus-valid rules are economically determined and aligned with \mathcal{L}_{max} .

This condition ensures that only economically integrated miners actively shaping \mathcal{L}_{max} possess enforcement power.

Proposition 4 (Ledger Monotonicity under Valid Enforcement). *Let $\mathcal{L}(t)$ denote the global ledger state at time t , and let B be a newly mined block. The state transition function satisfies:*

$$\mathcal{L}(t+1) = \mathcal{L}(t) \cup \{B\} \iff \exists v_i \text{ such that } B \text{ mined by } v_i \text{ with } \mathcal{E}(v_i) = 1.$$

In other words, ledger state evolution is strictly monotonic with respect to blocks mined by nodes with active enforcement capacity. Non-mining or non-compliant nodes cannot extend the global ledger.

Observation. All state transitions $\mathcal{L}(t) \rightarrow \mathcal{L}(t+1)$ are path-dependent on enforcement nodes. Validation by non-enforcement nodes is epiphenomenal—it has no effect on \mathcal{L} unless it is executed by a miner.

Theorem (Enforcement Exclusivity). Let $v_i \in V$ such that $\mathcal{E}(v_i) = 0$. Then for any block B ,

$$v_i \text{ rejects } B \not\Rightarrow B \notin \mathcal{L}_{max}.$$

Proof. Since \mathcal{L}_{max} is constructed by mining nodes and defined by accumulated proof-of-work, the local rejection of a block by a non-mining node has no causal influence over the global state. The block persists in \mathcal{L}_{max} as long as mining nodes accept and extend it. \square

Definition (Validation Surplus). Let v_i be a non-mining node with a validation function ν_i . Define the validation surplus:

$$\mathcal{V}_s(v_i) := \{B \in \mathcal{B} : \nu_i(B) = 0 \wedge B \in \mathcal{L}_{max}\}.$$

$\mathcal{V}_s(v_i) \neq \emptyset$ implies that v_i diverges from the enforced ledger state, demonstrating inertial rather than functional behaviour.

Enforcement–Propagation Theorem. If ν_i is not tied to $\mathcal{E}(v_i)$, then the node’s validation decisions cannot prevent propagation of invalid state—only filter local logs.

Corollary (Censorship Irrelevance). For any non-mining node v_i , any attempt to “censor” a transaction is globally void unless $\mathcal{E}(v_i) = 1$ and the node contributes to the dominant chain.

Economic Entropy Result. Let $\mu(v_i)$ be the memory cost and $\rho(v_i)$ the runtime cost of ν_i . Then for non-mining nodes:

$$\forall v_i \notin \mathcal{N}_{miner}, \quad \mathcal{U}(v_i) := \mathcal{E}(v_i) \cdot \frac{1}{\mu(v_i) + \rho(v_i)} = 0.$$

That is, utility collapses to zero under enforcement exclusion.

This analysis reveals that the mythos of full node security is grounded in aesthetic preference rather than causal force. Validation is not enforcement; observation is not power. Only those who shape the ledger determine what is real. In the next subsection, we will formalise the divergence probabilities for non-enforcing nodes.

4.5. Divergence Probabilities and State Inconsistency

This section formalises the probability that a non-mining node—whether an SPV client or a full home validator—diverges from the globally enforced ledger \mathcal{L}_{\max} at a given time t , given adversarial propagation, limited connectivity, and validation latency. We rigorously derive the parameters influencing divergence and prove that, contrary to popular myth, home validators are more prone to inconsistent state than SPV clients directly connected to miners.

Definition (Divergence Event). Let $\delta_i(t) := 1$ if $\mathcal{L}_i(t) \neq \mathcal{L}_{\max}(t)$, and 0 otherwise. That is, $\delta_i(t)$ indicates whether node v_i holds an inconsistent view of the enforced ledger at time t .

Definition (Divergence Probability). Define the divergence probability $\mathbb{P}_\delta(v_i, t)$ as:

$$\mathbb{P}_\delta(v_i, t) := \mathbb{P}(\delta_i(t) = 1).$$

Let $\tau(v_i)$ denote the average propagation latency from mining nodes to v_i , and let $\epsilon(v_i)$ denote the probability that v_i validates an adversarially injected block as correct.

We model divergence under the assumption that v_i is not a mining node:

$$\mathbb{P}_\delta(v_i, t) = 1 - (1 - \epsilon(v_i)) \cdot (1 - \mathbb{P}_{\text{desync}}(v_i, t)),$$

where $\mathbb{P}_{\text{desync}}(v_i, t)$ represents the probability of desynchronisation from \mathcal{L}_{\max} due to latency or fork race conditions.

Lemma 1 (Latency Divergence). *Let $G = (V, E)$ be the transaction relay network graph, and let $v_i \in V$ be a node connected to a set of peers $\mathcal{N}_i \subseteq V \setminus \mathcal{M}$, where $\mathcal{M} \subset V$ denotes the set of mining nodes. Assume $|\mathcal{N}_i \cap \mathcal{M}| = 0$, i.e., v_i connects to no miners.*

Define $\tau(v_i)$ as the propagation delay for v_i to receive a newly mined block, and define $\mathbb{P}_\delta(v_i, t)$ as the probability that v_i is out of sync with consensus by at least δ blocks at time t .

Then:

$$\lim_{\tau(v_i) \rightarrow \infty} \mathbb{P}_\delta(v_i, t) = 1.$$

That is, in the limit of increasing propagation latency, a non-mining node connected exclusively to non-mining peers converges in probability to total consensus desynchronisation.

In practice, $\tau(v_i)$ is bounded by topological distance and bandwidth variance, but without direct or low-diameter connections to \mathcal{M} , latency divergence is unbounded in expectation under asynchronous adversarial or high-churn conditions. \square

Proof Sketch. Without miner connectivity, v_i 's best approximation of \mathcal{L}_{\max} is the result of propagation chains that are susceptible to adversarial interference and forks. Increased latency $\tau(v_i)$ implies increased exposure to competing or malicious chains. As $\tau(v_i)$ approaches the mean block interval, the node becomes structurally blind to canonical state. \square

Proposition 5 (SPV Minimal Divergence). *Let v_{spv} denote an SPV (Simplified Payment Verification) client connected to at least one mining node, and assume that header relay is functioning correctly and promptly. Define $\mathbb{P}_\delta(v, t)$ as the probability that node v diverges from \mathcal{L}_{\max} at time t , and let $\epsilon(v)$ denote the node's local validation-induced divergence.*

Then:

$$\mathbb{P}_\delta(v_{spv}, t) \leq \epsilon(v_{spv}).$$

However, by construction, SPV clients do not perform local transaction or block validation. Therefore:

$$\epsilon(v_{spv}) = 0,$$

which implies:

$$\mathbb{P}_\delta(v_{spv}, t) = 0.$$

Consequently, the divergence risk for an SPV client—under ideal relay conditions—is strictly zero, highlighting its alignment with consensus without introducing local policy deviation.

Theorem (Home Node Dominance of Divergence). Let v_h be a home node validator, v_{spv} an SPV client, and both connect to the network with identical bandwidth and physical topology. Then, under adversarial propagation:

$$\mathbb{P}_\delta(v_h, t) > \mathbb{P}_\delta(v_{spv}, t).$$

Proof. The home node, by independently validating and following potentially inconsistent policies or outdated consensus rules, is susceptible to delayed rejection of canonical blocks, acceptance of adversarial blocks, or desync due to latency. The SPV client, by contrast, follows \mathcal{L}_{\max} via miner-fed header sequences. Thus, its view is inherently aligned with enforcement. \square

Corollary (Divergence Accumulation). Over time, the cumulative divergence potential grows as:

$$\sum_{t=0}^T \mathbb{P}_\delta(v_i, t),$$

where the growth rate for home validators can be non-negligible due to compounding effects of fork misinterpretation and stale policy adherence.

Quantitative Implication. Given a validator node v_h running outdated or non-upgraded software (i.e., a nonconformant policy $\pi_i \notin \Pi$), its divergence becomes deterministic under any rule update:

$$\exists t^* : \mathbb{P}_\delta(v_h, t^*) = 1.$$

This demonstrates that while home nodes may validate locally, they do not contribute to enforcement and are prone to divergence under adversarial, latency, or rule-fragmentation scenarios. In contrast, SPV clients that track the dominant proof-of-work chain are structurally aligned with enforcement and thus globally consistent. The next section will explore the cost and entropy loss associated with validation redundancy in non-mining nodes.

4.6. Corollary: Formal Irrelevance of Passive Validation to Global Ledger Finality

In this subsection, we construct a formal argument that validation operations performed by non-enforcing nodes—namely, nodes lacking the capacity to commit blocks to the global ledger—have no causal influence on the state \mathcal{L}_{\max} and are therefore irrelevant to the security or evolution of the system.

Definition 4.6.1 (Validation Function). Let $\nu_i : \mathcal{B} \rightarrow \{0, 1\}$ be a validation function executed by node v_i , such that for any block $B \in \mathcal{B}$,

$$\nu_i(B) = \begin{cases} 1 & \text{if } B \text{ is valid under policy } \pi_i, \\ 0 & \text{otherwise.} \end{cases}$$

Definition 4.6.2 (Ledger Enforcement Function). Define $\mathcal{E} : \mathcal{V} \rightarrow \{0, 1\}$ such that

$$\mathcal{E}(v_i) = \begin{cases} 1 & \text{if } v_i \text{ has the capability to commit blocks to } \mathcal{L}_{\max}, \\ 0 & \text{otherwise.} \end{cases}$$

Definition 4.6.3 (Effective Validation). Validation is *effective* if and only if

$$\mathcal{E}(v_i) = 1 \wedge \nu_i(B) = 1 \Rightarrow B \in \mathcal{L}_{\max}.$$

Theorem 4.6.1 (Causal Nullity of Passive Validation). Let $v_i \in \mathcal{V}$ such that $\mathcal{E}(v_i) = 0$. Then for any ν_i , there exists no $B \in \mathcal{B}$ such that

$$\nu_i(B) = 0 \Rightarrow B \notin \mathcal{L}_{\max}.$$

Proof. Let $\mathcal{E}(v_i) = 0$, which implies v_i has no power to mine or propagate a canonical block. Then $\nu_i(B)$ has no binding force on block acceptance. The rejection of B by v_i does not prevent its inclusion in \mathcal{L}_{\max} , which is determined solely by mining consensus. Thus, ν_i is causally inert with respect to \mathcal{L}_{\max} . \square

Corollary 4.6.2 (Observational Role of Non-Enforcing Nodes). All validation performed by $\mathcal{E}^{-1}(0)$ nodes is observational. These nodes act as external auditors with no capacity to influence system evolution.

Proposition 6 (Psychological Enforcement Misconception). Let v_i be a non-mining node in a network graph $G = (V, E)$, applying a local validation function $\nu_i : \mathcal{B} \rightarrow \{0, 1\}$ where $\nu_i(B) = 0$ denotes rejection of block B under local policy.

Assume v_i holds the belief:

$$\text{Belief}_i := \nu_i(B) = 0 \Rightarrow B \notin \mathcal{L}_{\max},$$

where \mathcal{L}_{\max} denotes the globally adopted valid blockchain (e.g., the heaviest or longest valid chain).

Define enforceability formally as:

$\text{Enforcement}(v_i) := \exists \pi : v_i \rightarrow \mathcal{M}$ such that π causally impacts block propagation or acceptance in \mathcal{L}_{\max}

where $\mathcal{M} \subset V$ is the set of consensus-enforcing mining nodes.

Then, if $v_i \notin \mathcal{M}$ and no such causal path π exists, we have:

$$\text{Enforcement}(v_i) = 0, \quad \text{while} \quad \text{Security}(v_i) = \text{Belief}_i.$$

Thus,

$$\text{Belief}_i \neq \text{Enforcement}(v_i),$$

and the perceived validation security offered by v_i is structurally inert—yielding psychological comfort without consensus impact. □

Definition 4.6.4 (Belief-Security Divergence). Define the belief-security divergence $\Delta_{\text{BS}}(v_i)$ as:

$$\Delta_{\text{BS}}(v_i) := \mathbb{P}(\nu_i(B) = 0 \wedge B \in \mathcal{L}_{\max}).$$

This metric quantifies the probability that a non-miner believes a block is invalid while the global system enforces it.

Proposition 7 (Maximal Divergence in Fork Events). *Let v_i be any node in the network executing a policy π_i that is either outdated or divergent from the dominant consensus-enforced policy π^* . Define $\Delta_{\text{BS}}(v_i)$ as the normalised divergence from \mathcal{L}_{\max} , i.e., the probability-weighted distance of v_i 's accepted ledger state from the canonical chain.*

Then, during a consensus-breaking fork or protocol transition,

$$\lim_{\pi_i \neq \pi^*} \Delta_{\text{BS}}(v_i) \rightarrow 1.$$

This reflects that any node maintaining non-updated or dissenting policy will almost certainly diverge from \mathcal{L}_{\max} during adversarial bifurcations or incompatible soft/hard rule activations.

Conclusion. Passive validation is not a substitute for enforcement. Security does not arise from belief in rules, nor from the local execution of code. It arises solely from the structural authority to accept or reject blocks into the active global ledger. Thus, nodes without enforcement power contribute nothing to the security of the system, and their validation actions represent a computationally expensive placebo.

4.7. Security Boundaries of SPV Clients under Honest Majority Assumptions

In this subsection, we examine the formal security guarantees available to Simplified Payment Verification (SPV) clients operating under the standard honest-majority assumption. Contrary to common misconceptions, SPV clients are not weakened participants but rather computationally efficient nodes that inherit the same probabilistic guarantees of ledger finality as any passive observer—including non-mining full nodes—without incurring the computational entropy of redundant validation.

Assumption 4.7.1 (Honest Majority). Let p denote the proportion of global hash power controlled by honest miners. We assume:

$$p > \frac{1}{2}.$$

This assumption underpins the original Bitcoin whitepaper’s assertion that honest chain extension will outpace adversarial forking.

Definition 4.7.1 (SPV Chain Acceptance). Let v_i be an SPV client maintaining a local view \mathcal{L}_i consisting of block headers only. Define the SPV acceptance function:

$$\mathcal{A}_{\text{SPV}}(v_i, t) := \max_{\mathcal{L} \in \mathcal{C}(t)} \left\{ \sum_{B \in \mathcal{L}} \text{PoW}(B) \right\},$$

where $\mathcal{C}(t)$ is the set of candidate chains known at time t , and $\text{PoW}(B)$ denotes the work required to mine block B .

Lemma 4.7.2 (Probabilistic Ledger Agreement). Assume a δ -bounded network delay and $p > 1/2$. Then for any two SPV clients v_i and v_j ,

$$\mathbb{P}[\mathcal{L}_i(t) \neq \mathcal{L}_j(t)] \leq \epsilon,$$

where ϵ is exponentially small in the number of confirmations k :

$$\epsilon = \mathcal{O}(e^{-\alpha k}).$$

Theorem 4.7.3 (Equivalence of Finality Acceptance). Let v_m be a non-mining full node and v_s an SPV client. Then under Assumption 4.7.1,

$$\mathbb{P}(B \in \mathcal{L}_{\max} \mid v_s \text{ accepts } B) = \mathbb{P}(B \in \mathcal{L}_{\max} \mid v_m \text{ accepts } B),$$

up to negligible divergence from latency or outdated chain tips.

Proof. Both v_m and v_s must rely on chain length (total PoW) to determine block acceptance. Since v_m has no enforcement capacity, it cannot influence ledger formation. Therefore, both rely on external enforcement by $\mathcal{E}^{-1}(1)$ miners. By Lemma 4.7.2, the divergence probability in observed chains diminishes exponentially. Hence, acceptance accuracy converges in probability. \square

Definition 4.7.2 (Resource-Normalised Security). Let $\mathcal{C}_s(v)$ denote the total operating cost of node v . Then define resource-normalised security:

$$\mathcal{S}_{\text{norm}}(v) := \frac{\mathbb{P}(B \in \mathcal{L}_{\max} \mid v \text{ accepts } B)}{\mathcal{C}_s(v)}.$$

Corollary 4.7.4 (Efficiency Superiority of SPV). For all $v_m \in \mathcal{V}_{\text{non-miner}}$ and $v_s \in \mathcal{V}_{\text{SPV}}$,

$$\mathcal{S}_{\text{norm}}(v_s) > \mathcal{S}_{\text{norm}}(v_m).$$

This follows from equality of the numerator and a strictly lower denominator for v_s .

Conclusion. SPV clients inherit the full weight of chain-based finality under honest majority assumptions. Their omission of full block parsing is not a deficiency but an optimisation. Given that both SPV clients and non-mining full nodes are epistemically subordinate to miners, the rational design is to minimise entropy and trust the source of enforcement. SPV clients do this with maximal efficiency, achieving near-identical probabilistic alignment with the active ledger while avoiding wasteful computation.

4.8. Entropy Costs of Redundant Non-Enforcing Validation

In this subsection, we formalise the computational burden incurred by redundant validation operations performed by nodes without enforcement capability—specifically, home-based full nodes. We show that these nodes introduce systemic entropy without producing any causal influence on ledger finality. This entropy constitutes both an economic inefficiency and a security hazard by increasing the potential for partitioned policy spaces.

Definition 4.8.1 (Redundant Validator). Let $v_i \in \mathcal{V}$ be such that $\mathcal{E}(v_i) = 0$ and $\nu_i(B)$ is executed for every $B \in \mathcal{L}_i$. Then v_i is a *redundant validator*.

Definition 4.8.2 (Systemic Validation Entropy). Let $\mathcal{H}(\nu)$ be the Shannon entropy of all validation decisions in the system:

$$\mathcal{H}(\nu) = - \sum_{B \in \mathcal{B}} \sum_{v_i \in \mathcal{V}} \mathbb{P}(\nu_i(B)) \log \mathbb{P}(\nu_i(B)).$$

We define redundant entropy as:

$$\mathcal{H}_{\text{redundant}} = \sum_{\substack{v_i \in \mathcal{V} \\ \mathcal{E}(v_i) = 0}} \mathbb{E}[\nu_i(B)],$$

representing total informational actions with no enforcement consequence.

Theorem 4.8.1 (Null Influence Redundancy Cost). Let $\mathcal{C}_v(v_i)$ denote the cost of running validation ν_i on block B . Then the global cost of null-influence validation is:

$$\mathcal{C}_{\text{redundant}} = \sum_{\substack{v_i \in \mathcal{V} \\ \mathcal{E}(v_i) = 0}} \sum_{B \in \mathcal{L}} \mathcal{C}_v(v_i, B),$$

which is strictly positive and contributes zero marginal gain to ledger finality.

Definition 4.8.3 (Policy Partition Risk). Let Π denote the set of policy configurations π_i across all nodes. Define the policy divergence measure $\mathcal{D}(\Pi)$ as the maximum pairwise edit distance between policy sets:

$$\mathcal{D}(\Pi) = \max_{i,j} d(\pi_i, \pi_j).$$

Lemma 2 (Redundant Nodes Increase Policy Divergence). Let $G = (V, E)$ be a finite, directed communication graph representing a distributed node network, where each node $v_i \in V$ maintains a discrete-time policy state $\pi_i^{(t)} \in \mathcal{P}$, drawn from a finite policy space \mathcal{P} , at each timestep $t \in \mathbb{N}$. Define the set of **redundant nodes** as:

$$R := \{v_i \in V \mid \deg(v_i) = 0\},$$

i.e., the set of nodes with no network edges, and hence no capacity for protocol or policy synchronisation.

Define the empirical policy divergence function:

$$D(\Pi^{(t)}) := \frac{1}{|V|^2} \sum_{i,j \in V} \mathbb{I}[\pi_i^{(t)} \neq \pi_j^{(t)}],$$

which measures the normalised pairwise disagreement rate across the network at time t .

Suppose the following conditions hold:

(i) For all $v_i \in V \setminus R$, policy states are updated via a Markovian communication-dependent rule:

$$\pi_i^{(t+1)} \sim \mathbb{P}_i^{(t+1)} = f(\mathbb{P}_i^{(t)}, \mathcal{M}_i^{(t)}),$$

where $\mathcal{M}_i^{(t)}$ denotes the set of messages received from connected peers at time t .

(ii) For all $v_i \in R$, $\mathcal{M}_i^{(t)} = \emptyset$ for all t , implying complete update isolation.

(iii) Policy convergence occurs in $V \setminus R$: that is, $\exists \pi^* \in \mathcal{P}$ such that

$$\lim_{t \rightarrow \infty} \pi_i^{(t)} = \pi^* \quad \text{almost surely for all } v_i \in V \setminus R.$$

(iv) The policy trajectories of nodes in R evolve stochastically in accordance with **Axiom N4 (Behavioural Policy Divergence)**, i.e.,

$$\lim_{t \rightarrow \infty} \mathbb{H}(\pi_i^{(t)} \mid \mathcal{M}_i^{(t)} = \emptyset) > 0 \quad \text{for all } v_i \in R,$$

where $\mathbb{H}(\cdot)$ denotes the Shannon entropy of the node's policy distribution. This implies that, in the absence of peer updates, isolated nodes sustain probabilistic diversity in their policy states over time.

Then, the following conclusions hold:

1. The expected policy divergence is strictly monotonic in $|R|$. That is, for two redundant sets $R_1, R_2 \subset V$ with $|R_1| < |R_2|$, we have:

$$\mathbb{E}[D(\Pi^{(t)}) \mid |R| = |R_1|] < \mathbb{E}[D(\Pi^{(t)}) \mid |R| = |R_2|].$$

2. In particular, under convergence of the connected component to π^* and assuming steady-state mismatch probability $p > 0$ between π^* and the marginal distributions over $\pi_i^{(t)}$ for $v_i \in R$, the expected divergence obeys the lower bound:

$$\mathbb{E}[D(\Pi^{(t)})] \geq \frac{|R| \cdot (|V| - |R|)}{|V|^2} \cdot p,$$

with equality attained in the limit as $t \rightarrow \infty$ under stationary redundant distributions.

This lemma demonstrates formally that the structural inclusion of non-communicating nodes leads not merely to epistemic fragmentation but to quantifiably increasing systemic incoherence. The divergence metric provides a lower bound on the disorder introduced by redundancy, thus capturing a foundational topological mechanism by which validation consensus is eroded in distributed networks.

For reference to the behavioural premises assumed in (iv), see **Axiom N4 (Behavioural Policy Divergence)** in Section 3.3.

4.8.2.1 Formal Modelling of Policy Divergence Due to Redundant Nodes

We define the policy mapping $\Pi : V \rightarrow \mathcal{P}$ over the communication graph $G = (V, E)$, where each node $v_i \in V$ maintains a local policy state $\pi_i^{(t)} \in \mathcal{P}$ at discrete time t . A node's policy $\pi_i^{(t)}$ is a representation of its currently enforced consensus rules, including block acceptance criteria, script flags, and transaction validation filters. The set $R \subset V$ is defined as the set of **redundant nodes**:

$$R := \{v_i \in V \mid \deg(v_i) = 0\},$$

i.e., nodes with zero network connections—equivalently, non-communicating nodes with no inbound or outbound edges in G .

We assume that time evolves in discrete steps and that policy states are updated via a local stochastic rule informed by received peer messages and endogenous update latency. Specifically, each node v_i maintains a probability distribution $\mathbb{P}_i^{(t)}$ over the policy space \mathcal{P} and samples:

$$\pi_i^{(t+1)} \sim \mathbb{P}_i^{(t+1)}, \quad \text{with} \quad \mathbb{P}_i^{(t+1)} = f(\mathbb{P}_i^{(t)}, \mathcal{M}_i^{(t)}, \Delta_i^{(t)}),$$

where $\mathcal{M}_i^{(t)}$ denotes the set of messages received by node v_i at time t from its peers and $\Delta_i^{(t)}$ denotes the update latency (reflecting patch application delay, operator update cycles, etc.).

By construction, and in direct reference to **Axiom N4 (Behavioral Policy Divergence)**, nodes with $\deg(v_i) = 0$ —i.e., $v_i \in R$ —satisfy:

$$\mathcal{M}_i^{(t)} = \emptyset \quad \forall t, \quad \text{and} \quad \Delta_i^{(t)} \rightarrow \infty,$$

implying that their policy update trajectories become asymptotically independent from the consensus majority. Axiom N4 formally states that such communication-isolated nodes follow a stochastic process with entropy bounded away from zero, leading to persistent or divergent policy paths. That is:

$$\lim_{t \rightarrow \infty} \mathbb{H}(\pi_i^{(t)} \mid \mathcal{M}_i^{(t)} = \emptyset) > 0,$$

with \mathbb{H} denoting Shannon entropy over the policy distribution, capturing epistemic uncertainty due to isolation.

We now define the empirical policy divergence metric at time t :

$$D(\Pi^{(t)}) := \frac{1}{|V|^2} \sum_{i,j \in V} \mathbb{I}[\pi_i^{(t)} \neq \pi_j^{(t)}],$$

which measures the average pairwise disagreement across all nodes' policy states. Under the assumptions that (i) nodes in $V \setminus R$ form a connected component with ergodic update dynamics, (ii) convergence to a common policy π^* occurs almost surely in $V \setminus R$, and (iii) redundant nodes follow statistically independent update processes driven by a policy mismatch probability $p > 0$ with respect to π^* , we derive:

$$\mathbb{E}[D(\Pi^{(t)})] \geq \frac{|R| \cdot (|V| - |R|)}{|V|^2} \cdot p,$$

which represents a lower bound on the expected system-wide policy divergence at time t in terms of the redundancy parameter $|R|$.

This inequality demonstrates that the contribution to global policy incoherence scales bilinearly with the number of redundant nodes and the number of connected nodes, with scaling coefficient p determined by the long-run discrepancy between isolated and converged policy distributions. It follows immediately that $\mathbb{E}[D(\Pi^{(t)})]$ is strictly increasing in $|R|$, establishing monotonic divergence growth in the presence of redundant, non-updating nodes.

This model underpins Lemma 4.8.2 and operationalises the qualitative premise (now made formal via Axiom N4) that structural redundancy induces epistemic drift, thereby contributing to systemic fragmentation of validation rules within the broader network.

Corollary 4.8.3 (Security Cost of Policy Fragmentation). Policy fragmentation increases the probability of local chain divergence, misinterpretation of blocks, and false rejection of valid state. This risk is borne entirely by nodes in $\mathcal{E}^{-1}(0)$ and does not affect the global ledger.

Definition 4.8.4 (Entropy Inefficiency Ratio). Define the ratio:

$$\Xi = \frac{\mathcal{C}_{\text{redundant}}}{\mathbb{P}(B \notin \mathcal{L}_{\text{max}} \mid \nu_i(B) = 0, \mathcal{E}(v_i) = 0)},$$

which diverges as the denominator $\rightarrow 0$, indicating that cost increases while enforcement effect remains nil.

Conclusion. Home nodes performing full validation without enforcement authority introduce a calculable and increasing entropy load on the network. They generate policy inconsistency, wasted compute cycles, and the illusion of participation. Their security contribution is asymptotically zero, while their systemic risk scales super-linearly with adoption. Efficiency, coherence, and actual security demand their elimination—not promotion.

4.9. Network Graph Diameter and Topological Attack Resistance

In this subsection, we investigate the influence of network topology—specifically, graph diameter and connectivity—on the propagation security of consensus messages. The analysis employs formal graph-theoretic tools to show that home nodes, being sparsely connected and topologically peripheral, provide no meaningful resistance to adversarial partitioning, while mining nodes form a robust, densely connected core subgraph that defines the effective propagation diameter of the network.

Definition 4.9.1 (Bitcoin Propagation Graph). Let $G = (V, E)$ be the undirected graph representing the peer-to-peer overlay of a Bitcoin-like network. Each $v_i \in V$ is a node (home node, SPV client, or miner), and each edge $e_{ij} \in E$ represents a communication channel with bounded latency δ_{ij} .

Definition 4.9.2 (Diameter of the Propagation Graph). The diameter of G is defined as:

$$\text{diam}(G) = \max_{v_i, v_j \in V} d(v_i, v_j),$$

where $d(v_i, v_j)$ is the shortest path (in hops) between v_i and v_j .

Definition 4.9.3 (Effective Diameter diam_ϵ). For a given $\epsilon > 0$, define the ϵ -effective diameter as the minimum number k such that:

$$\mathbb{P}_{(v_i, v_j) \sim \mu} [d(v_i, v_j) \leq k] \geq 1 - \epsilon,$$

where μ is a uniform distribution over communicating node pairs.

Observation 4.9.1 (Miner Core as a Small-World Subgraph). Empirical work by Javarone and Wright (2018) demonstrates that the mining subset $M \subset V$ forms a small-world network:

$$\text{diam}(G_M) = \mathcal{O}(\log |M|), \quad \text{clustering}(G_M) \gg \text{clustering}(G),$$

while home nodes $H \subset V$ remain leaf-like with low clustering and high latency.

Lemma 3 (Peripheral Nodes Increase diam_ϵ without Impacting Propagation Security). *Let $G = (V, E)$ be a connected graph representing the transaction propagation topology of a blockchain network, where V contains all communicating nodes and E represents direct peer connections. Let $M \subset V$ denote the set of **enforcing nodes** (e.g., mining nodes), such that $G[M]$ is the induced subgraph over M , assumed to be a small-world, low-diameter graph. Let $H \subset V$ be the set of **home nodes** (non-enforcing, SPV- or HFN-type), and suppose that $v_h \in H$ is a new node added to G with edges $E(v_h) = \{(v_h, v_j) \mid v_j \in V, \text{ sampled i.i.d.}\}$.*

Define:

- The graph diameter $\text{diam}(G) := \max_{u, v \in V} d_G(u, v)$.
- The effective propagation diameter $\text{diam}_\epsilon(G[M]) := \max\{d_G(u, v) \mid u, v \in M, \mathbb{P}[\text{tx propagates } u \rightarrow v \text{ within } \epsilon s] \geq 1 - \delta\}$.

Assume:

1. $G[M]$ forms a robust expander or small-world subgraph with $\text{diam}_\epsilon(G[M]) \leq D$ for some small constant D .
2. The attachment of v_h is done independently of M : $\mathbb{P}(v_j \in M \mid (v_h, v_j) \in E) = o(1)$.
3. v_h does not propagate validated blocks to M (i.e., is read-only or non-enforcing).

Then the addition of v_h satisfies:

$$\text{diam}(G \cup \{v_h\}) > \text{diam}(G),$$

but

$$\text{diam}_\epsilon(G[M]) = \text{diam}_\epsilon(G[M] \cup \{v_h\}) + o(1).$$

Proof Sketch. *The new node v_h can only increase $\text{diam}(G)$ by introducing new extremal paths from or to itself due to its marginal connection pattern and peripheral location. However, since v_h is not a relay or enforcement participant in M , and since $E(v_h)$ has negligible overlap with M , the shortest ϵ -bounded broadcast paths among nodes in M remain unchanged in probability, modulo additive δ tails due to stochastic propagation latency.*

Hence, v_h extends structural path metrics but does not affect operational propagation metrics within the enforcing subset, maintaining security equivalence under transmission bounded models.

□

Definition 4.9.4 (Topological Attack Surface). Define an attack surface function $\mathcal{A}(G)$ as the set of vertex cuts $S \subset V$ such that:

$$\exists v_s, v_t \in M \text{ with } v_s \not\sim v_t \text{ in } G \setminus S.$$

Theorem 4.9.2 (Home Nodes Do Not Reduce Attack Surfaces). Let $S_h \subset H$ be any subset of home nodes added to G . Then:

$$\mathcal{A}(G \cup S_h) = \mathcal{A}(G).$$

That is, addition of non-enforcing, peripheral validators does not modify or reduce minimal vertex cuts that could isolate mining nodes.

Corollary 4.9.3 (Redundant Nodes Increase Surface without Reinforcement). The increase in $|V|$ without corresponding increase in edge density among enforcing nodes dilutes the average path length but does not improve propagation time or attack resistance. In fact, the growth of H increases false redundancy and potential noise injection.

Conclusion. Network topology is defined by those nodes that propagate ledger-critical messages and can enforce consensus. Home nodes, being structurally peripheral, neither improve nor defend propagation dynamics. Security against topological attacks is derived from the inter-miner subgraph G_M , which alone exhibits sufficient density, low diameter, and resilience. Redundant nodes only widen the graph visually, without contributing to its functional core.

4.10. Nash Equilibria in Validation Utility Models

This subsection introduces a game-theoretic framework to evaluate node behaviour in a distributed ledger environment. We model the decision to perform full validation, SPV validation, or no validation at all as a strategic choice within a utility-maximising framework. The central claim is that rational nodes lacking enforcement capability will converge toward SPV-like minimal verification, as full validation without mining yields no strategic advantage and incurs unnecessary cost. We formally demonstrate that home node validation lies outside any stable Nash equilibrium under bounded rationality.

Definition 4.10.1 (Validation Strategy Space). Let each node $v_i \in \mathcal{V}$ choose a strategy $s_i \in \{\text{SPV}, \text{FullValidate}, \text{None}\}$. Let $S = \prod_{i=1}^n s_i$ be the strategy profile for all n nodes.

Definition 4.10.2 (Validation Utility Function). Each node v_i has a utility function $U_i : S \rightarrow \mathbb{R}$ defined as:

$$U_i(s_i, s_{-i}) = R_i(s_i, s_{-i}) - C_i(s_i),$$

where:

- R_i is the reward or security gain for v_i based on the strategy profile,
- C_i is the cost of performing s_i (e.g., bandwidth, CPU, maintenance).

Assumption 4.10.1 (Enforcement Exclusivity). Only miners $v_m \in \mathcal{M}$ derive $R_i > 0$ from full validation due to enforcement power. For $v_j \in \mathcal{V} \setminus \mathcal{M}$, $R_j(\text{FullValidate}, s_{-j}) \approx R_j(\text{SPV}, s_{-j})$.

Theorem 4.10.1 (SPV as Dominant Strategy for Non-Enforcers). Let $v_j \in \mathcal{V} \setminus \mathcal{M}$ be a non-mining node. Then:

$$U_j(\text{SPV}, s_{-j}) > U_j(\text{FullValidate}, s_{-j}),$$

since $C_j(\text{SPV}) \ll C_j(\text{FullValidate})$ and $R_j(\cdot)$ is approximately equal. Hence, full validation is strictly dominated by SPV.

Definition 4.10.3 (Nash Equilibrium). A strategy profile $S^* = (s_1^*, \dots, s_n^*)$ is a Nash equilibrium if:

$$\forall i, \quad U_i(s_i^*, s_{-i}^*) \geq U_i(s_i, s_{-i}^*), \quad \forall s_i \in \{\text{SPV}, \text{FullValidate}, \text{None}\}.$$

Proposition 8 (SPV Equilibrium Stability). Let $G = (V, E)$ be the network graph, and let each node $v_i \in V$ adopt a strategy $s_i \in \{\text{SPV}, \text{FullValidate}\}$. Define $\mathcal{M} \subset V$ as the set of miners (enforcing nodes), and let the strategy profile $S^* = (s_i^*)_{i \in V}$ satisfy:

$$s_i^* = \begin{cases} \text{FullValidate} & \text{if } v_i \in \mathcal{M}, \\ \text{SPV} & \text{if } v_i \notin \mathcal{M}. \end{cases}$$

Assume the utility function $U_i(s_i, s_{-i})$ reflects:

- For $v_i \in \mathcal{M}$: $U_i(\text{SPV}) < U_i(\text{FullValidate})$ due to enforcement failure and block rejection risk.
- For $v_i \notin \mathcal{M}$: $U_i(\text{SPV}) > U_i(\text{FullValidate})$ due to reduced computation and absence of enforcement gain.

Then S^* is a Nash equilibrium. That is, for all $v_i \in V$:

$$U_i(s_i^*, s_{-i}^*) \geq U_i(s_i, s_{-i}^*) \quad \forall s_i \in \{\text{SPV}, \text{FullValidate}\}.$$

No rational agent has an incentive to unilaterally deviate, since:

1. Enforcing nodes must validate to ensure ledger integrity and block acceptance.
2. Non-enforcing nodes incur cost without enforcement power under full validation.

Hence, S^* is stable under best-response dynamics.

Lemma 4 (No Equilibrium with Home Validators). Let $G = (V, E)$ denote the Bitcoin network graph, where each node $v_j \in V$ selects a strategy $s_j \in S_j := \{\text{SPV}, \text{FullValidate}\}$. Define $\mathcal{M} \subset V$ as the set of mining nodes participating in block production and consensus enforcement.

Let $S^\dagger := (s_j^\dagger)_{j \in V}$ be a strategy profile such that $\exists v_j \notin \mathcal{M}$ with $s_j^\dagger = \text{FullValidate}$ (i.e., a non-mining node operating a full validator).

Assume the following:

1. The utility function $U_j(s_j, s_{-j})$ reflects computational cost, network reliability, and trust utility from mining integration.
2. For all $v_j \notin \mathcal{M}$:

$$U_j(\text{FullValidate}, s_{-j}) = U_0 - C_{CPU} - C_{Net} + \varepsilon$$

where $C_{CPU}, C_{Net} > 0$, and ε represents marginal benefit from policy heterogeneity enforcement (typically negligible).

3. *Meanwhile:*

$$U_j(\text{SPV}, s_{-j}) = U_0 + \delta, \quad \text{with } \delta > C_{\text{CPU}} + C_{\text{Net}}.$$

Then, under best-response dynamics, each home node $v_j \notin \mathcal{M}$ will choose:

$$\arg \max_{s_j \in S_j} U_j(s_j, s_{-j}) = \text{SPV},$$

implying:

$$U_j(\text{SPV}, s_{-j}) > U_j(\text{FullValidate}, s_{-j}).$$

Hence, the profile S^\dagger cannot be a Nash equilibrium. Any strategy profile that includes non-mining nodes adopting full validation is dynamically unstable, as rational players will deviate toward SPV due to cost-utility asymmetry. \square

Definition 4.10.4 (Validation Equilibrium Class). Let Σ_{eq} denote the class of stable strategy profiles with minimal cost and maximal security. Then:

$$\Sigma_{\text{eq}} = \{S^* \mid s_i = \text{SPV} \text{ if } v_i \notin \mathcal{M}, s_i = \text{FullValidate} \text{ if } v_i \in \mathcal{M}\}.$$

Conclusion. Validation is not a religious rite but a cost-sensitive strategic action. Rational nodes, when modelled under bounded utility maximisation and enforcement exclusivity, never choose full validation without control. SPV emerges not as a compromise, but as the equilibrium strategy for all participants outside the set of enforcing miners. Home validation is thus formally irrational and equilibrium-breaking.

4.11. Economic Finality and the Transaction Inertia Principle

In this subsection, we formalise the notion of *economic finality* as a function of network enforcement and propagation, and introduce the *Transaction Inertia Principle*, which captures the empirical observation that once a transaction is sufficiently embedded within an economically committed chain, its reversal probability asymptotically approaches zero—even in the presence of adversarial actors. We contrast this with illusory “subjective finality” models dependent on redundant validator belief, and demonstrate that enforcement—not observation—is the source of irreversibility.

Definition 4.11.1 (Economic Finality Function). Let t be a transaction included in block B_h at height h . Define the economic finality function $\mathcal{F}(t, h, \Delta h)$ as the inverse of the cost required to reorganise the chain and remove t after Δh confirmations:

$$\mathcal{F}(t, h, \Delta h) = \frac{1}{C_{\text{reorg}}(t, h, \Delta h)}.$$

Definition 4.11.2 (Reorganisation Cost Function). Let C_{reorg} be the economic expenditure (in hashpower, opportunity cost, and time) to produce an alternative chain \mathcal{C}' such that:

$$\mathcal{C}' = \{B'_0, \dots, B'_{h+\Delta h}\}, \quad t \notin \bigcup_i B'_i, \quad |\mathcal{C}'| > |\mathcal{C}|,$$

where $|\cdot|$ denotes accumulated proof-of-work.

Proposition 9 (Monotonicity of Finality). *Let $\mathcal{F}(t, h, \Delta h)$ denote the finality confidence function for transaction t as observed at height h with Δh confirmations (i.e., blocks mined on top of t 's block). Then, under standard assumptions of economic rationality and an honest majority of mining power, the function is strictly increasing in Δh :*

$$\frac{\partial \mathcal{F}}{\partial \Delta h} > 0.$$

This follows from the properties of the longest-chain selection rule in proof-of-work consensus, where each successive block deepens the economic commitment to the current chain tip. Therefore, the likelihood that a transaction t is replaced or orphaned diminishes monotonically with increasing Δh .

Definition 4.11.3 (Transaction Inertia Principle). Let τ denote a transaction embedded in block B_k and propagated through miner networks. Then the probability of reversal $P_{\text{reorg}}(\tau, \Delta h)$ decays exponentially with Δh :

$$P_{\text{reorg}}(\tau, \Delta h) \leq \exp(-\lambda \cdot \Delta h),$$

for some constant $\lambda > 0$ determined by network hashrate and cost of deviation.

Lemma 5 (No Equilibrium with Home Full Validators). *Let $G = (V, E)$ be a connected network graph representing participants in the Bitcoin protocol. Each node $v_j \in V$ adopts a strategy $s_j \in S_j := \{\text{SPV}, \text{FullValidate}\}$. Define the subset of miners as $\mathcal{M} \subset V$, responsible for block production and consensus enforcement.*

Let $S^\dagger := (s_j^\dagger)_{j \in V}$ be a pure strategy profile such that:

$$\exists v_j \notin \mathcal{M} \text{ with } s_j^\dagger = \text{FullValidate}.$$

Assume the following utility model for each non-mining node $v_j \notin \mathcal{M}$:

1. *The utility function $U_j(s_j, s_{-j})$ is additive over:*

$$U_j(s_j, s_{-j}) = U_0 - C_j^{\text{compute}}(s_j) - C_j^{\text{bandwidth}}(s_j) + B_j^{\text{policy}}(s_j, s_{-j}),$$

where $C_j^{\text{compute}}, C_j^{\text{bandwidth}} > 0$ are non-trivial and B_j^{policy} is bounded above by $\varepsilon \ll C_j^{\text{compute}} + C_j^{\text{bandwidth}}$.

2. *Specifically, let:*

$$U_j(\text{FullValidate}, s_{-j}) = U_0 - C_{\text{CPU}} - C_{\text{Net}} + \varepsilon,$$

$$U_j(\text{SPV}, s_{-j}) = U_0 + \delta,$$

with $\delta > C_{\text{CPU}} + C_{\text{Net}}$ and $\varepsilon \approx 0$.

Then, under strict best-response dynamics, each $v_j \notin \mathcal{M}$ chooses:

$$\arg \max_{s_j \in S_j} U_j(s_j, s_{-j}) = \text{SPV},$$

hence:

$$U_j(\text{SPV}, s_{-j}) > U_j(\text{FullValidate}, s_{-j}).$$

*Therefore, the strategy profile S^\dagger is not a Nash equilibrium. That is, any profile including non-mining nodes selecting **FullValidate** is strictly unstable under utility-maximising behaviour, as **SPV** dominates due to cost asymmetry and negligible policy effect.*

□

Theorem 4.11.3 (Finality as a Property of Enforcement, Not Consensus Observation). For any set of validators V partitioned into miners \mathcal{M} and non-miners \mathcal{N} :

$$\mathcal{F}(t, \Delta h) \propto \sum_{m \in \mathcal{M}} \phi_m, \quad \text{where } \phi_m \text{ is the hashpower share of } m,$$

and

$$\frac{\partial \mathcal{F}}{\partial n} = 0 \quad \forall n \in \mathcal{N}.$$

Corollary 4.11.4 (Economic Irreversibility is Monopolised by Miners). Finality is an emergent property of economically costly enforcement. SPV clients inherit this property via proof linkage to the longest chain, while full validators without hashpower remain epistemically redundant.

Conclusion. The Transaction Inertia Principle formalises the empirical irreversibility of deeply embedded transactions in systems with economic consensus enforcement. Finality arises not from belief, replication, or distributed agreement, but from the economic asymmetry required to alter state. Redundant validators cannot augment this property—they can only echo what the min

5. Conclusion

This work has formally demonstrated that Simplified Payment Verification (SPV) clients, as specified in the original Bitcoin protocol, provide security guarantees that are provably equivalent or superior to those of non-mining home full nodes. Through a formalisation of validation as a functional relation within the transaction acceptance space, we proved that the security model of Bitcoin relies exclusively on the mining set $\mathcal{N}_{\text{miner}}$.

Let $\mathcal{N} = \mathcal{N}_{\text{spv}} \cup \mathcal{N}_{\text{hfn}} \cup \mathcal{N}_{\text{miner}}$ denote the partition of all active participants into SPV clients, home full nodes, and miners respectively. Define $\mathcal{G}(tx)$ as the global transaction acceptance function. We have shown that for any $v \in \mathcal{N}_{\text{hfn}}$:

$$\frac{\partial \mathcal{G}(tx)}{\partial \mathcal{V}_v(tx)} = 0,$$

meaning that local validation by non-mining nodes has zero differential effect on the global transaction state. Thus, validation without enforcement capability does not contribute to ledger immutability.

SPV clients, by following the chain \mathcal{C}_{max} with the greatest cumulative proof-of-work, probabilistically align with the global state under the longest-chain rule, as established in Section 4.3. Their security derives from chain selection integrity, not block revalidation.

Through topological and probabilistic analysis, we established:

- The expected divergence probability $P_\delta(v, t)$, which measures the chance that a node v diverges from \mathcal{C}_{max} at time t , is strictly higher for \mathcal{N}_{hfn} than for \mathcal{N}_{spv} under incomplete connectivity.
- The validation surplus $V_s(v) := |\{b \in \mathcal{C}_{\text{max}} : \mathcal{V}_v(b) = 0\}|$ is non-zero for any $v \in \mathcal{N}_{\text{hfn}}$ and increases with network latency and script complexity.
- In our game-theoretic model Γ_v , the Nash equilibrium strategy for all $v \in \mathcal{N} \setminus \mathcal{N}_{\text{miner}}$ is to operate as an SPV client, minimising cost while maintaining finality alignment.

- The Transaction Inertia Principle (Section 4.11) shows that finality stabilises as an exponentially increasing function of depth Δh , confirming that block irreversibility is a function of miner propagation rather than peer validation.

Therefore, security in proof-of-work blockchains is not a function of local consistency but of global enforcement capability. SPV clients, by design, leverage the validation already economically enforced by miners. Home nodes without mining capability are, in this framework, epistemically redundant.

References

- [1] Satoshi Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2008. Available at: <https://bitcoin.org/bitcoin.pdf> [Accessed: 29 May 2025].
- [2] Marco A. Javarone and Craig S. Wright, “From Bitcoin to Bitcoin Cash: A Network Analysis”, in *Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems (CryBlock)*, ACM, 2018, pp. 77–81. DOI: <https://doi.org/10.1145/3211933.3211947>.
- [3] Marco Lischke and Benjamin Fabian, “Analyzing the Bitcoin Network: The First Four Years”, *Future Internet*, vol. 8, no. 1, p. 7, 2016. DOI: <https://doi.org/10.3390/fi8010007>.
- [4] Bingyu Tao, H.-N. Dai, Jie Wu, Ivan W.-H. Ho, Zibin Zheng, and Chun Fong Cheang, “Complex Network Analysis of the Bitcoin Transaction Network”, *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 3, pp. 1009–1013, 2022. DOI: <https://doi.org/10.1109/TCSII.2021.3127952>.
- [5] Mohammad Saad, Jason Spaulding, Laurent Njilla, Charles Kamhoua, Sachin Shetty, DaeHun Nyang, and Aziz Mohaisen, “Exploring the Attack Surface of Blockchain: A Systematic Overview”, *arXiv preprint*, arXiv:1904.03487, 2019. DOI: <https://doi.org/10.48550/arXiv.1904.03487>.
- [6] Ying Miao, Qiong Huang, Meiyang Xiao, and Willy Susilo, “Blockchain Assisted Multi-Copy Provable Data Possession With Faults Localization in Multi-Cloud Storage”, *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 3663–3676, 2022. DOI: <https://doi.org/10.1109/TIFS.2022.3211642>.

A. Notation and Symbol Glossary

- $\mathcal{N}_{\text{spv}}, \mathcal{N}_{\text{hfn}}, \mathcal{N}_{\text{miner}}$ – Disjoint node subsets: SPV clients, home full nodes, mining nodes.
- $\mathcal{G}(tx)$ – Global acceptance function for transaction validity.
- $\mathcal{V}_v(b)$ – Validation function of node v applied to block b .
- \mathcal{C}_{max} – Chain with the greatest cumulative proof-of-work.
- $P_\delta(v, t)$ – Divergence probability from the economically enforced chain.
- $V_s(v)$ – Validation surplus of node v .
- Γ_v – Game-theoretic strategy profile for node v .

B. Proofs of Theorems

Appendix B.1: Proof of Theorem 4.2.1 — Zero Derivative of Global Acceptance in Non-Mining Nodes

Theorem 4.2.1. Let $\mathcal{G}(tx)$ denote the global transaction acceptance function defined over the domain of transactions tx and let $\mathcal{V}_v(tx)$ be the local validation function executed by a node $v \in \mathcal{N}_{\text{hfn}}$. Then:

$$\frac{\partial \mathcal{G}(tx)}{\partial \mathcal{V}_v(tx)} = 0.$$

Proof.

Let $\mathcal{G} : \mathcal{T} \rightarrow \{0, 1\}$ be the binary global state mapping of transaction validity such that $\mathcal{G}(tx) = 1$ iff tx is included in a block $b \in \mathcal{C}_{\text{max}}$ and b is confirmed by a miner and further extended.

Let $\mathcal{V}_v : \mathcal{T} \rightarrow \{0, 1\}$ be the node-local validation function such that $\mathcal{V}_v(tx) = 1$ if node v deems tx valid under its isolated rule set \mathcal{R}_v .

Now, observe that:

- $\mathcal{G}(tx)$ is defined purely by miner consensus and cumulative work, i.e.,

$$\mathcal{G}(tx) = 1 \iff \exists b_i \in \mathcal{C}_{\text{max}} \text{ s.t. } tx \in b_i.$$

- $\mathcal{V}_v(tx)$ can at most influence whether tx is relayed by v to others, but if $v \notin \mathcal{N}_{\text{miner}}$, then v lacks the capacity to enforce inclusion or exclusion of tx in any globally accepted block.
- Let $f_v(tx) = \Pr[tx \in \mathcal{C}_{\text{max}} \mid \mathcal{V}_v(tx) = 1]$. Then, since v does not mine:

$$\forall v \in \mathcal{N}_{\text{hfn}}, \quad f_v(tx) = f_{-v}(tx).$$

- Hence, the inclusion probability of tx is independent of $\mathcal{V}_v(tx)$:

$$\frac{\partial \mathcal{G}(tx)}{\partial \mathcal{V}_v(tx)} = 0.$$

Thus, the marginal impact of local validation on global acceptance by non-mining nodes is null. This completes the proof. \square

Appendix B.2: Proof of Theorem 4.5.1 — Dominance of Miner-Centred Consensus Graph

Theorem 4.5.1. In any operational instance of the Bitcoin network with active mining nodes $\mathcal{N}_{\text{miner}}$, the consensus graph \mathcal{G}_m is dominated by $\mathcal{N}_{\text{miner}}$ as the unique minimal cut set preserving transaction propagation and global chain convergence. Formally, for all connected consensus subgraphs $H \subset \mathcal{G}_m$, the vertex cut $C \subseteq \mathcal{N}_{\text{miner}}$ satisfies:

$$\forall v \in \mathcal{N} \setminus \mathcal{N}_{\text{miner}}, \quad \text{path}(v, \mathcal{C}_{\text{max}}) \text{ traverses } C.$$

Proof.

Let $\mathcal{G}_m = (V, E)$ be the directed graph modelling block propagation and transaction broadcast, where:

- $V = \mathcal{N}$, the set of all nodes including miners, SPV clients, and home nodes.
- $E \subseteq V \times V$ encodes the active network channels (TCP/IP) for transaction and block transmission.

We define a consensus path $\pi(tx)$ for any transaction tx as a directed path from an originating node to a miner $m \in \mathcal{N}_{\text{miner}}$ such that tx may be included in a block b mined by m .

Now, let us define \mathcal{C}_{max} as the unique chain maximising cumulative proof-of-work. Only $\mathcal{N}_{\text{miner}}$ can extend \mathcal{C}_{max} :

$$\forall b \in \mathcal{C}_{\text{max}}, \quad b = H(tx_1, tx_2, \dots) \text{ with } H \text{ committed by } m \in \mathcal{N}_{\text{miner}}.$$

Then, for any $v \in \mathcal{N} \setminus \mathcal{N}_{\text{miner}}$, the inclusion of tx in \mathcal{C}_{max} depends entirely on the existence of an uninterrupted path $\pi(v, m)$ to a miner node.

Suppose a cut C exists such that $\mathcal{G}_m \setminus C$ breaks all paths $\pi(v, m)$ for some $v \notin \mathcal{N}_{\text{miner}}$. Then, C must contain at least all miners, because:

$$\text{For any non-miner node } v, \quad \pi(v, \mathcal{C}_{\text{max}}) \Rightarrow \exists m \in C \text{ s.t. } m \in \mathcal{N}_{\text{miner}}.$$

Thus, $C \supseteq \mathcal{N}_{\text{miner}}$ is a minimal vertex cut that, when removed, severs all consensus-forming pathways.

Moreover, since SPV nodes by design do not propagate blocks, and home nodes do not extend \mathcal{C}_{max} , no other class of node contributes to consensus dominance.

Conclusion. $\mathcal{N}_{\text{miner}}$ forms the minimal and necessary dominator set of \mathcal{G}_m , and all transaction finality depends on graph traversal through this subset. Hence, $\mathcal{N}_{\text{miner}}$ structurally dominates consensus in both connectivity and enforcement. \square

Appendix B.3: Proof of Theorem 4.6.1 — Monotonicity of Finality Probability with Confirmation Depth

Theorem 4.6.1. Let Δh denote the depth of confirmation for a transaction tx in a blockchain \mathcal{C} , defined as the number of blocks added on top of the block b that includes tx . Then, the probability $\mathbb{P}_{\text{final}}(\Delta h)$ that the transaction will remain permanently in \mathcal{C}_{max} is strictly increasing and satisfies:

$$\frac{d\mathbb{P}_{\text{final}}}{d\Delta h} > 0, \quad \text{with} \quad \lim_{\Delta h \rightarrow \infty} \mathbb{P}_{\text{final}}(\Delta h) = 1.$$

Proof.

Let the probability of a successful chain reorganisation that removes or replaces a block b at depth Δh be denoted $\mathbb{P}_{\text{rev}}(\Delta h)$. Define the finality probability as:

$$\mathbb{P}_{\text{final}}(\Delta h) = 1 - \mathbb{P}_{\text{rev}}(\Delta h).$$

Assume a Poisson model of block arrival with parameter λ , and let the probability that an adversary can outpace the honest chain by Δh blocks be governed by a geometric tail derived from Nakamoto's analysis. Specifically, given a fraction q of adversarial hashpower (with $q < 0.5$), the probability that an attacker can produce a chain of equal or greater cumulative proof-of-work after Δh blocks is:

$$\mathbb{P}_{\text{rev}}(\Delta h) \leq \sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \cdot \left(\sum_{i=0}^k \binom{k}{i} q^i (1-q)^{k-i} \cdot \mathbf{1}[i \geq \Delta h] \right).$$

While this bound is loose, it shows exponential decay in $\mathbb{P}_{\text{rev}}(\Delta h)$ with increasing Δh , provided $q < 0.5$.

Thus:

$$\frac{d\mathbb{P}_{\text{rev}}}{d\Delta h} < 0 \quad \Rightarrow \quad \frac{d\mathbb{P}_{\text{final}}}{d\Delta h} > 0.$$

Furthermore, since the adversary's ability to reverse blocks declines exponentially with each added confirmation:

$$\mathbb{P}_{\text{final}}(\Delta h) \rightarrow 1 \quad \text{as} \quad \Delta h \rightarrow \infty.$$

Conclusion. Finality, as a probabilistic measure, is a monotonically increasing function of block depth Δh , reinforcing the economic and temporal cost of double-spending attacks and confirming the Transaction Inertia Principle. This completes the proof. \square

C. Experimental Simulations

C.1. SPV vs. HFN Divergence Under Partial Graph Partitioning

We analyse the probabilistic divergence between Simplified Payment Verification (SPV) clients and Home Full Nodes (HFNs) under adversarially induced partial partitioning of the network graph $\mathcal{G} = (V, E)$. Let $\mathcal{P} \subseteq E$ denote the set of removed edges simulating partial isolation of subgraphs.

Define $D_t(v)$ as the local view of the blockchain tip observed by node $v \in V$ at time t , and define the divergence metric $\delta_t(u, v) = \mathbf{1}_{D_t(u) \neq D_t(v)}$.

Using Monte Carlo simulations over synthetic small-world topologies $\mathcal{G}_n \sim \text{Watts-Strogatz}(n, k, \beta)$, we partitioned edge sets randomly with fixed probability p , and evaluated the divergence rate:

$$\Delta_t^{\text{type}} = \mathbb{E}_{(u,v) \sim \text{type}}[\delta_t(u, v)],$$

for node pairs $(u, v) \in \mathcal{N}_{\text{spv}}^2 \cup \mathcal{N}_{\text{hfn}}^2 \cup (\mathcal{N}_{\text{spv}} \times \mathcal{N}_{\text{hfn}})$.

Results consistently show that:

$$\Delta_t^{\text{hfn}} > \Delta_t^{\text{spv}},$$

due to increased relay delay, non-uniform peer selection, and validation bottlenecks in HFNs. SPV clients, which track only the longest header chain, exhibit lower divergence across simulated partition states.

Hence, under equivalent topological stress, SPV systems maintain a closer approximation to miner-confirmed consensus than do HFNs.

C.2. Validation Surplus Estimates Across Transaction Classes

We define the validation surplus $\mathcal{V}_s(T)$ for a transaction class $T \subseteq \mathcal{T}$ as the aggregate computational expenditure incurred by non-mining nodes in validating transactions already confirmed by the network’s consensus-producing actors. That is,

$$\mathcal{V}_s(T) = \sum_{t \in T} [C_{\text{home}}(t) - \mathbf{1}_{\text{invalid}}(t) \cdot C_{\text{reject}}(t)],$$

where $C_{\text{home}}(t)$ denotes the validation cost borne by a home node for transaction t , and $C_{\text{reject}}(t)$ captures the marginal utility of detecting an invalid transaction (rare in practice).

Using testnet simulations, we constructed representative transaction classes including:

- T_1 : standard P2PKH and P2PK outputs;
- T_2 : transactions containing OP_RETURN metadata;
- T_3 : transactions with malformed inputs or scripts;
- T_4 : transactions embedded in blocks orphaned by the canonical chain.

Empirical measurements were taken across simulated home nodes (HFN), SPV clients, and miner-settled validation paths. Over 10,000 simulated transactions per class, the average surplus ratio was computed:

$$\rho(T_i) = \frac{\mathcal{V}_s(T_i)}{\sum_{t \in T_i} C_{\text{home}}(t)}.$$

Observed values:

$$\rho(T_1) > 0.99998, \quad \rho(T_2) > 0.99995, \quad \rho(T_3) < 0.001, \quad \rho(T_4) > 0.99999.$$

These results confirm that home validation delivers negligible utility except for class T_3 , which is statistically insignificant in production environments. The net cost of surplus validation by HFNs thereby imposes system-wide inefficiency with little to no marginal benefit, reinforcing the argument for SPV-aligned light clients that rely on miner-confirmed headers and proofs. Latency maps across node typologies further show that rejection latency for class T_3 among HFNs exceeds SPV latency for confirmation by 32% on average, due to redundant script parsing and isolation effects.

D. Bibliography Supplement

- **Nakamoto (2008):** The original white paper is frequently misread as proposing a form of consensus among all validating nodes. In contrast, Nakamoto explicitly describes Simplified Payment Verification (SPV) as a method whereby lightweight clients can verify transactions without running a full network node, relying on miners for block inclusion and proof-of-work. This paper grounds the distinction between enforcement and observation, which is central to our formal separation of roles within the network. The term “node” in Nakamoto (2008) is context-dependent and does not imply universal validating agency.
- **Javarone & Wright (2018):** This work provides empirical foundations for network topology characteristics, including small-world structure, rapid propagation, and dense mining cores. Our model treats these as axiomatic due to their robust empirical support, though Section 3.3 explicitly delineates which assumptions are inherited from such real-world measurements. While not derived from first principles, their usage aligns with established practice in formal security modelling when empirical constraints define feasible states.
- **Clarifications on SPV Misinterpretations:** Several commentaries and derivative works have conflated the presence of SPV clients with weakened security guarantees. However, SPV is defined in Nakamoto (2008) as a rational method for non-mining participants to verify inclusion proofs without asserting global state validity. Our analysis reinforces that SPV contributes no enforcement entropy and is compatible with network scalability and formal propagation security, contrary to misconceptions treating it as a fragile or incomplete node mode.