future internet

MDPI

Article

EPoLBFT: A Blockchain Consensus Algorithm for Enhancing Privacy, Invulnerability and Trust in IoT System

Yunus Kareem [1,*], Djamel Djenouri [1] and Essam Ghadafi [2]

[1] Department of Computer Science and Creative Technologies, University of the West of England, Bristol BS16 1QY, UK; djamel.djenouri@uwe.ac.uk
[2] School of Computing, Newcastle University, Newcastle-Upon-Tyne NE1 7RU, UK
* Correspondence: yunus.kareem@uwe.ac.uk; Tel.: +44-7502-583-020

Abstract

The rapid growth of Internet of Things (IoT) systems has introduced significant challenges related to privacy, trust, scalability, and attack resilience, particularly in resource-constrained and location-sensitive environments. Existing blockchain consensus mechanisms provide decentralised trust, but they often suffer from high communication overhead, weak physical-context awareness, and limited privacy protection when deployed in large-scale IoT networks. This paper proposes Elastic Proof-of-Location Byzantine Fault Tolerance (EPoLBFT), a privacy-preserving and location-aware blockchain consensus framework for IoT systems. The proposed design enables IoT nodes to prove regional eligibility without revealing exact coordinates while restricting consensus participation to trusted and geographically verified validators. EPoLBFT is evaluated using the Blockchain IoT Consensus Algorithm (BICA) simulator under normal, high-load, Byzantine, Sybil, location-spoofing, and denial-of-service scenarios. The evaluation considers latency, throughput, communication overhead, energy consumption, and attack resilience. The results show that EPoLBFT reduces communication overhead and improves consensus efficiency compared with conventional PBFT-based approaches while strengthening resilience against location-based and identity-based attacks. The study also discusses the privacy–latency trade-off introduced by zk-PoL, the assumptions related to trusted location anchors, and the limitations of simulation-based evaluation.

Keywords: Internet of Things (IoT); blockchain; Proof-of-Location; Byzantine Fault Tolerance; privacy preservation; trust management; consensus algorithm; resource-constrained systems

Academic Editor: Paolo Bellavista

Received: 22 May 2026
Revised: 3 July 2026
Accepted: 10 July 2026
Published: 15 July 2026

1. Introduction

The proliferation of Internet of Things (IoT) devices has fundamentally reshaped modern life. By 2027, the global IoT ecosystem is projected to exceed 22 billion connected devices [1]. However, this rapid adoption has also introduced a vast and complex attack surface [2]. Traditional IoT system architectures, predominantly based on centralised cloud-based models, present significant security and privacy risks [3]. A single breach of a central server can compromise entire networks, exposing sensitive user data and giving malicious actors control over critical physical infrastructure, such as smart locks, climate control systems, and medical devices.

To mitigate these risks, decentralised frameworks based on blockchain technology have emerged as a promising alternative [4,5]. By replacing a central point of failure with

a distributed and immutable ledger, blockchain provides a more resilient and trustworthy foundation for IoT interactions. Despite this potential, the application of traditional blockchain consensus mechanisms, such as Proof of Work (PoW) and Proof of Stake (PoS), to IoT environments remains challenging [6]. The high energy consumption and computational overhead associated with PoW make it unsuitable for resource-constrained IoT devices, while PoS-based systems may not align well with the dynamic and transient participation patterns typical of IoT nodes [7].

The Byzantine Fault Tolerance (BFT) algorithm, particularly the Practical Byzantine Fault Tolerance (PBFT) algorithm [8], offers a more suitable alternative due to its deterministic finality and ability to tolerate up to $f < \frac{n}{3}$ malicious nodes without requiring energy-intensive mining. However, conventional BFT protocols face several key limitations when applied to large-scale IoT systems:

1. Scalability bottleneck: PBFT exhibits $O(n^2)$ message complexity, which becomes prohibitive as the network size increases beyond approximately 50–100 nodes [9].
2. Resource constraints: IoT devices typically have limited computational power, memory, and energy resources, making frequent cryptographic operations and multiphase communication inefficient [10].
3. Network heterogeneity: IoT deployments consist of nodes with diverse capabilities, intermittent connectivity, and variable latency, which conventional BFT protocols do not adequately address [11].
4. Lack of physical awareness: Traditional BFT protocols rely solely on cryptographic authentication, ignoring the physical context of devices. In many IoT applications, such as smart homes or healthcare systems, the legitimacy of a command is inherently tied to its physical origin [12]. For example, a command to unlock a door should ideally originate from a verified device located within or near the premises, rather than from a remote but compromised node.

To address this limitation, location-aware security mechanisms such as Proof-of-Location (PoL) have been introduced. Protocols including Geographic Practical Byzantine Fault Tolerance (G-PBFT) [13] and Delegated Proof of Proximity (DPoP) [14] have demonstrated the potential of integrating geographic context into consensus processes. However, these approaches rely on static configurations and incur significant communication overhead, limiting their scalability in dynamic IoT environments. Other applications of PoL have been explored in blockchain-based tracking systems, such as supply chain logistics [15], but their integration into consensus mechanisms remains limited.

To overcome these challenges, this paper proposes Elastic Proof-of-Location Byzantine Fault Tolerance (EPoLBFT), an enhanced consensus algorithm designed to improve security, scalability, and trust in IoT systems. The proposed framework introduces two key innovations: (i) a dynamic beacon-based election mechanism that adaptively assigns location verification responsibilities, and (ii) an optimised zero-knowledge-based Proof-of-Location (zk-PoL) process that reduces communication overhead while strengthening resilience against denial-of-service and location-spoofing attacks.

EPoLBFT operates under a partially synchronous network model with a computationally bounded adversary capable of controlling up to $f < \frac{n-1}{3}$ Byzantine nodes. The adversary may delay or drop messages but cannot compromise standard cryptographic primitives or trusted location infrastructure. The system assumes the availability of reliable location data and supports dynamic node participation to maintain the required fault tolerance threshold.

To evaluate the effectiveness of the proposed approach, EPoLBFT is implemented and tested using the Blockchain IoT Consensus Algorithm (BICA) simulator [16]. Its performance is benchmarked against G-PBFT [13], Directed Acyclic Graph BFT (DAG-BFT) [17],

DCBFT [18], and standard PBFT [19]. Experimental results demonstrate that EPoLBFT improves security, reduces transaction latency, and lowers communication overhead, establishing it as a robust and practical solution for securing decentralised IoT systems. The main contribution of this work is not the isolated use of proof-of-location, Byzantine fault tolerance, or zero-knowledge proofs but their integration into a unified consensus framework for privacy-preserving and resilient IoT blockchain networks. EPoLBFT combines zk-PoL, BFT consensus, geo-sharded committee formation, reputation-aware validator selection, and elastic committee adaptation. This combination allows validators to prove regional eligibility without revealing precise coordinates, limits Sybil and spoofing attacks through physical-context verification, and reduces consensus overhead by confining BFT communication to trusted regional committees. The remainder of this paper is structured as follows. Section 2 provides an overview of related work on IoT blockchain consensus mechanisms and Proof-of-Location systems. Section 3 introduces the proposed EPoLBFT architecture and details the experimental methodology. Section 4 describes the simulation environment and presents the corresponding results and performance evaluation. Section 5 examines the security properties of the proposed scheme. Finally, Section 6 offers concluding remarks and outlines directions for future research.

2. Related Work

The development of the proposed EPoLBFT consensus framework is informed by extensive research in Byzantine Fault Tolerance (BFT) [5], zero-knowledge proof systems [20], and location-aware consensus mechanisms for IoT environments [21]. This section reviews related work across five major research directions: PBFT adaptations for IoT, reputation-based and scalable consensus mechanisms, zero-knowledge integration in distributed systems, location-based consensus protocols, and hybrid approaches for IoT security. The limitations of existing solutions are highlighted to motivate the design choices of EPoLBFT.

2.1. PBFT Enhancements for IoT Systems

Practical Byzantine Fault Tolerance (PBFT) has served as a foundational consensus protocol for permissioned blockchain systems due to its deterministic finality and tolerance of up to $f < \frac{n}{3}$ byzantine failures in partially synchronous networks [22]. Despite these advantages, the direct application of PBFT to IoT environments remains challenging. Its quadratic message complexity $O(n^2)$, reliance on multiple communication phases, and cryptographic overhead introduce significant performance and energy constraints when deployed on resource-limited IoT devices [10]. These limitations have motivated a wide range of PBFT enhancements tailored to IoT requirements. Several studies have investigated deployment-level optimisations to improve PBFT feasibility in IoT networks. Onireti et al. [23] introduced the concept of a viable deployment region, demonstrating that careful geographic placement of replicas can significantly reduce communication cost and energy consumption. Similarly, Meshcheryakov et al. [24] evaluated PBFT on constrained IoT hardware platforms, confirming its functional feasibility but highlighting scalability and latency bottlenecks as network size grows. While these works provide valuable insights into deployment efficiency, they do not fundamentally address PBFT's inherent communication complexity.

Leader-centric designs in PBFT can create bottlenecks and single points of performance degradation. To mitigate this issue, Misic et al. [8] proposed a multi-leader PBFT variant that allows concurrent proposal generation. Although this approach improves throughput, it introduces coordination challenges, such as proposal conflicts and increased synchronisation overhead. In contrast, EPoLBFT distributes leadership geographically across committees, reducing contention while embedding physical-location awareness

directly into the consensus process. Reputation-based consensus mechanisms aim to reduce overhead by prioritising nodes with reliable historical behaviour. Yuan et al. [25] and Kumar et al. [11] proposed reputation-driven PBFT variants that improve efficiency and reduce latency. Niu et al. [26] further refined reputation-based PBFT for IoT-specific contexts. However, these approaches rely primarily on logical trust metrics, which can be manipulated through collusion or strategic behaviour, and they lack cryptographic guarantees tied to physical node properties.

Scalability improvements have also been pursued through partitioning and grouping strategies. Ramos et al. [27] introduced reputation-based validator selection to reduce communication overhead, while score-based grouping PBFT approaches achieved up to 73% reduction in consensus latency [28]. Despite these gains, such solutions do not address physical-layer threats, such as location spoofing, which remain critical in IoT environments.

2.2. Zero-Knowledge Proofs in Consensus Mechanisms

Zero-knowledge proofs (ZKPs) enable one party (the prover) to demonstrate the validity of a statement to another party (the verifier) without revealing any additional information beyond the truth of the statement itself [20]. In blockchain-based IoT systems, ZKPs are increasingly employed to preserve privacy while enabling verifiable computation, authentication, and compliance [29]. Existing ZKP-based consensus approaches can be broadly categorised according to their underlying cryptographic mechanisms and operational trade-offs.

2.2.1. Succinct Non-Interactive Proofs (zk-SNARKs)

zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge) rely on pairing-based elliptic curve cryptography and polynomial commitment schemes to generate compact proofs that can be verified efficiently. A trusted setup phase is typically required to generate public parameters. The main advantages of zk-SNARKs include: constant-size proofs, fast verification, and low on-chain overhead, making them suitable for resource-constrained IoT environments [30]. Wu et al. [20] demonstrated the feasibility of integrating zk-SNARK-based Proof-of-Location into blockchain systems, enabling nodes to prove geographic membership without revealing precise coordinates. Similarly, Alanazi et al. [31] incorporated lightweight zk-SNARK constructions into a DAG-based blockchain architecture to balance privacy and throughput. Despite their efficiency, zk-SNARK-based systems require a trusted setup and incur relatively high proof-generation costs, which must be carefully managed in IoT deployments.

2.2.2. Transparent Proof Systems (zk-STARKs)

zk-STARKs (Zero-Knowledge Scalable Transparent Arguments of Knowledge) eliminate the need for a trusted setup by leveraging hash-based cryptographic primitives and interactive oracle proofs [32]. These systems offer strong post-quantum security guarantees and transparency. However, zk-STARKs generate significantly larger proofs and require higher computational and communication overhead during verification, making them less suitable for low-power IoT devices and latency-sensitive consensus protocols.

2.2.3. Range Proofs and Bulletproofs

Bulletproofs are short non-interactive zero-knowledge proofs based on inner-product arguments that enable efficient range proofs without a trusted setup. They are particularly effective for proving numerical constraints, such as value bounds or resource limits, and are commonly used in privacy-preserving cryptocurrencies [33]. However, Bulletproofs require linear-time verification and are less expressive for complex statements such as multi-witness location verification, limiting their applicability in location-aware consensus mechanisms.

2.2.4. Secure Multiparty Computation-Based Proofs

Secure multiparty computation (MPC) techniques enable distributed parties to jointly compute a function while keeping inputs private. Geng et al. [34] proposed Delegated Proof of Secret Sharing (DPoSS), combining MPC with consensus protocols to preserve authentication privacy. While MPC-based approaches offer strong confidentiality guarantees, they introduce substantial communication rounds and computational overhead, which negatively impact scalability and latency in dynamic IoT networks.

2.2.5. Obfuscation and Encryption-Based Approaches

Obfuscation-based techniques rely on encrypted data aggregation and iterative consensus over masked values. Ambrosin et al. [35] proposed ODIN, which uses cryptographic obfuscation to prevent consensus participants from accessing raw data. Although effective in preserving confidentiality, such approaches incur additional computational cost and are less suitable for real-time consensus in resource-constrained environments.

2.3. Location-Based Consensus Protocols

Location-based consensus protocols aim to enhance trust and security by incorporating the physical location of participating nodes into the consensus process. The core intuition behind these approaches is that binding digital identities to verifiable physical presence can significantly reduce Sybil attacks and identity forgery, which are prevalent in IoT environments. However, existing solutions differ considerably in how location is obtained, verified, and integrated into consensus, leading to important trade-offs between privacy, scalability, and trust assumptions.

2.3.1. Geographic PBFT (G-PBFT)

Geographic Practical Byzantine Fault Tolerance (G-PBFT) [13] is one of the earliest attempts to integrate location awareness into a Byzantine fault-tolerant consensus protocol. In G-PBFT, nodes are deployed at fixed and known geographic locations, and consensus committees are selected based on spatial distribution. The protocol assumes that nodes within a geographic region can reliably verify each other's presence using proximity-based communication or trusted positioning services.

During consensus, only nodes located within predefined regions are allowed to participate, reducing the risk of Sybil attacks by imposing a physical deployment cost on adversaries. However, G-PBFT relies on static node locations and assumes that geographic information is publicly available and trusted. This exposes precise location data, creating significant privacy risks and making the protocol unsuitable for mobile or dynamic IoT environments. Additionally, the reliance on fixed infrastructure limits scalability and adaptability in real-world deployments.

2.3.2. Delegated Proof of Proximity (DPoP)

Delegated Proof of Proximity (DPoP) [14] extends the concept of location-aware consensus by introducing delegation mechanisms. In DPoP, nodes delegate their participation rights to nearby representatives that can prove proximity to a specific geographic area. Proximity is typically verified using short-range communication technologies such as Bluetooth, Wi-Fi, or signal strength measurements.

This approach reduces communication overhead by limiting active consensus participants while still leveraging physical proximity as a trust factor. However, DPoP relies heavily on signal-based proximity measurements, which are susceptible to spoofing, relay attacks, and environmental interference. Furthermore, the delegation process introduces additional trust assumptions, as compromised delegates can act maliciously on behalf of

multiple nodes. Privacy preservation is also limited, since proximity information may leak sensitive spatial data.

2.3.3. Infrastructure-Dependent Location Consensus

Several location-based consensus systems depend on external positioning infrastructures, such as GPS, cellular networks, or Wi-Fi fingerprinting services [36]. In these approaches, nodes submit location proofs generated by trusted location providers, which are then verified by consensus participants.

While infrastructure-assisted solutions offer high accuracy and ease of deployment, they introduce central points of failure and strong trust dependencies. A compromised or unavailable location service can undermine the entire consensus process. Moreover, continuous reliance on external services raises concerns regarding availability, scalability, and user privacy, making these approaches less suitable for decentralised IoT environments [29].

2.3.4. Privacy–Verification Trade-Off in Location Consensus

A fundamental challenge in location-based consensus protocols is balancing verification accuracy with privacy protection. Protocols that expose precise location data allow strong verification but create privacy risks, enabling targeted attacks and surveillance. Conversely, privacy-preserving approaches often weaken verifiability, as limited information is available to confirm physical presence [37].

Some systems attempt to obfuscate location data or use coarse-grained regions, but this typically introduces additional communication overhead and reduces resistance to spoofing attacks [20]. As a result, most existing solutions fail to simultaneously achieve strong privacy guarantees, cryptographic verification, and efficient consensus.

2.3.5. Scalability Limitations in Location-Aware Consensus

Scalability remains a critical limitation of current location-based consensus protocols. Most proposed systems have only been evaluated in small-scale deployments or controlled simulation environments [38]. As the network size increases, the overhead associated with location verification, witness coordination, and trust management grows significantly. In addition, dynamic IoT environments, where nodes frequently join, leave, or move, exacerbate these challenges. Existing protocols often lack mechanisms for dynamic committee reconfiguration or adaptive location verification, limiting their applicability to large-scale real-world IoT systems.

The reviewed location-based consensus protocols demonstrate the potential benefits of integrating physical context into distributed trust mechanisms. However, existing solutions suffer from one or more of the following limitations: reliance on trusted infrastructure, exposure of sensitive location data, susceptibility to spoofing attacks, static deployment assumptions, and poor scalability. These limitations motivate the design of EPoLBFT, which integrates a cryptographically verifiable and privacy-preserving Proof-of-Location mechanism directly into the consensus process. By combining zero-knowledge location proofs with dynamic committee selection and Byzantine fault tolerance, EPoLBFT overcomes the privacy, scalability, and trust limitations of prior location-aware consensus approaches.

2.4. Hybrid Consensus Approaches for IoT Security

Hybrid consensus mechanisms aim to address the limitations of individual consensus paradigms by combining multiple techniques, such as Byzantine Fault Tolerance, delegation, reputation, artificial intelligence, and resource-aware optimisation. In IoT environments, these hybrid approaches seek to balance security, scalability, and efficiency while accommodating heterogeneous device capabilities. Although hybrid designs offer

notable improvements over single-mechanism solutions, they introduce new trade-offs and unresolved challenges.

2.4.1. Multi-Phase and Hierarchical Consensus Designs

Multi-phase and hierarchical consensus mechanisms decompose the consensus process into multiple stages or layers to reduce communication overhead and improve scalability. Pelekoudas-Oikonomou et al. [37] proposed a scalable PBFT-based framework for Internet of Medical Things (IoMT) applications, where nodes are organised into hierarchical groups that perform local consensus before forwarding results to higher-level aggregators. This design reduces network-wide communication and improves latency in domain-specific deployments.

However, such hierarchical approaches introduce implicit trust assumptions between layers and may become vulnerable if higher-level aggregators are compromised. Furthermore, these systems typically rely solely on cryptographic authentication and do not incorporate physical-context validation, such as device location, which is critical in many IoT scenarios involving physical actuation and access control.

2.4.2. AI-Enhanced and Learning-Based Consensus Mechanisms

Artificial intelligence has been integrated into consensus protocols to dynamically adapt parameters and improve resilience under adversarial conditions. Haider et al. [38] proposed a trust-based delegated consensus mechanism that employs deep reinforcement learning to optimise validator selection and trust thresholds. By learning from historical behaviour, the system can adapt to changing network conditions and improve throughput and fault tolerance.

Despite these advantages, AI-enhanced consensus introduces new attack surfaces. Learning models may be vulnerable to adversarial manipulation, poisoning attacks, or concept drift, particularly in dynamic IoT environments. Additionally, such approaches incur training and inference overhead, which may not be suitable for resource-constrained devices. Importantly, AI-based mechanisms do not provide cryptographic guarantees of physical presence, limiting their effectiveness against location spoofing and Sybil attacks.

2.4.3. Resource-Aware and Delegation-Based Consensus Protocols

Resource-aware hybrid consensus protocols aim to optimise performance by delegating consensus responsibilities to nodes with higher computational or energy capacity. Heo et al. [39] proposed a delegation-based consensus model that selects validators based on resource availability, thereby reducing the burden on low-power IoT devices.

While delegation improves efficiency, it introduces centralisation risks and increases reliance on a small subset of powerful nodes. Compromise of delegated validators can have a disproportionate impact on system security. Moreover, these approaches typically lack integrated privacy-preserving mechanisms and do not verify the physical legitimacy of participating nodes.

2.4.4. Trust and Reputation-Augmented Hybrid Consensus

Several hybrid approaches incorporate reputation systems alongside traditional consensus mechanisms to improve trustworthiness and reduce malicious participation. By combining reputation scores with committee selection or delegation, these systems attempt to limit the influence of misbehaving nodes. However, reputation-based hybrids rely on historical behaviour, making them susceptible to long-term manipulation, collusion, and strategic attacks. Without cryptographic enforcement of physical attributes, such as location, reputation-based hybrids cannot effectively prevent Sybil attacks in large-scale IoT deployments, where adversaries can repeatedly join the network using new identities.

Hybrid consensus mechanisms represent a significant step toward addressing IoT-specific challenges by combining complementary techniques. However, existing hybrid designs suffer from several common limitations: reliance on implicit trust assumptions, lack of privacy-preserving physical verification, increased system complexity, and vulnerability to multi-vector attacks that exploit gaps between combined mechanisms.

None of the reviewed hybrid approaches simultaneously provides (i) cryptographically verifiable and privacy-preserving location assurance, (ii) strong Byzantine fault tolerance, and (iii) scalable performance optimised for resource-constrained IoT environments. These limitations motivate the design of EPoLBFT, which adopts a hybrid strategy that tightly integrates zero-knowledge Proof-of-Location with an optimised Byzantine Fault-Tolerant consensus protocol. By embedding cryptographic location verification directly into committee selection and consensus execution, EPoLBFT achieves a unified defence against Byzantine, Sybil, and location-spoofing attacks while maintaining scalability and efficiency.

2.5. Research Gaps and Motivation

The reviewed literature reveals several unresolved challenges: (i) the absence of privacy-preserving and decentralised location verification, (ii) limited integration of physical context into scalable BFT consensus, (iii) incomplete protection against multi-vector attacks, including Sybil, Byzantine, and location-spoofing attacks, and (iv) insufficient optimisation for resource-constrained IoT devices.

To address these gaps, this paper proposes EPoLBFT, which integrates a succinct zero-knowledge Proof-of-Location (zk-PoL) mechanism with an optimised PBFT protocol. By embedding location verification directly into consensus formation, EPoLBFT enables secure and decentralised validator selection while preserving location privacy and improving scalability and energy efficiency for IoT environments [20].

3. Elastic Proof-of-Location Byzantine Fault Tolerance Model

This section presents the proposed Elastic Proof-of-Location Byzantine Fault Tolerance (EPoLBFT) model. EPoLBFT is designed for privacy-preserving, resilient, and scalable IoT blockchain environments where consensus decisions must consider both cryptographic identity and physical context. The model integrates five components: zero-knowledge Proof-of-Location, trust-aware validator selection, geo-sharded committee formation, local Byzantine Fault-Tolerant consensus, and elastic consensus adaptation.

The objective of EPoLBFT is to ensure that only geographically valid and behaviourally trusted nodes participate in consensus while preventing validators from learning the exact physical coordinates of participating IoT devices. This is achieved through zk-PoL certificates, reputation filtering, and regional committees. Compared with conventional PBFT, which requires all validators to participate in global consensus, EPoLBFT reduces communication overhead by limiting BFT message exchange to smaller location-aware committees.

EPoLBFT is designed for consortium and permissioned IoT blockchain deployments, such as smart cities, industrial IoT, smart buildings, healthcare IoT, and critical infrastructure monitoring. These environments typically contain registered devices, gateways, validators, and location-verification entities. Permissionless deployment is possible only with additional mechanisms such as decentralised identity, stake-based admission, or verifiable credentials and is therefore considered outside the main scope of this paper.

3.1. System Architecture and Network Model

EPoLBFT operates on a permissioned IoT blockchain network composed of a set of nodes $N = \{n_1, n_2, \ldots, n_N\}$. The network follows a partially synchronous communication model, consistent with the assumptions of Practical Byzantine Fault Tolerance (PBFT).

The nodes are categorised into the following roles:

- IoT Nodes: Devices that are resource-constrained and generate transactions and location claims.
- Validator Nodes: Capable devices responsible for participating in consensus.
- Witness Nodes: Nearby nodes that assist in validating physical location claims.
- Consensus Committee: A dynamically selected subset of validator nodes responsible for executing the consensus protocol.

The adversary model assumes a computationally bounded attacker capable of controlling up to $f < \frac{n-1}{3}$ Byzantine nodes. The adversary may delay, replay, or drop messages but cannot break standard cryptographic primitives or forge zero-knowledge proofs. The protocol operates in iterative epochs, each consisting of four phases: (1) Location Attestation, (2) Committee Formation, (3) Intra-committee Consensus, and (4) Global Finalisation, as described in Figure 1. Location attestation leverages zk-PoL to verify node presence without revealing sensitive geographic information, while committee formation ensures decentralised and geographically aware validator selection.

The framework operates across diverse IoT environments, enabling secure, privacy-preserving consensus through integrated location verification and Byzantine fault-tolerant coordination.

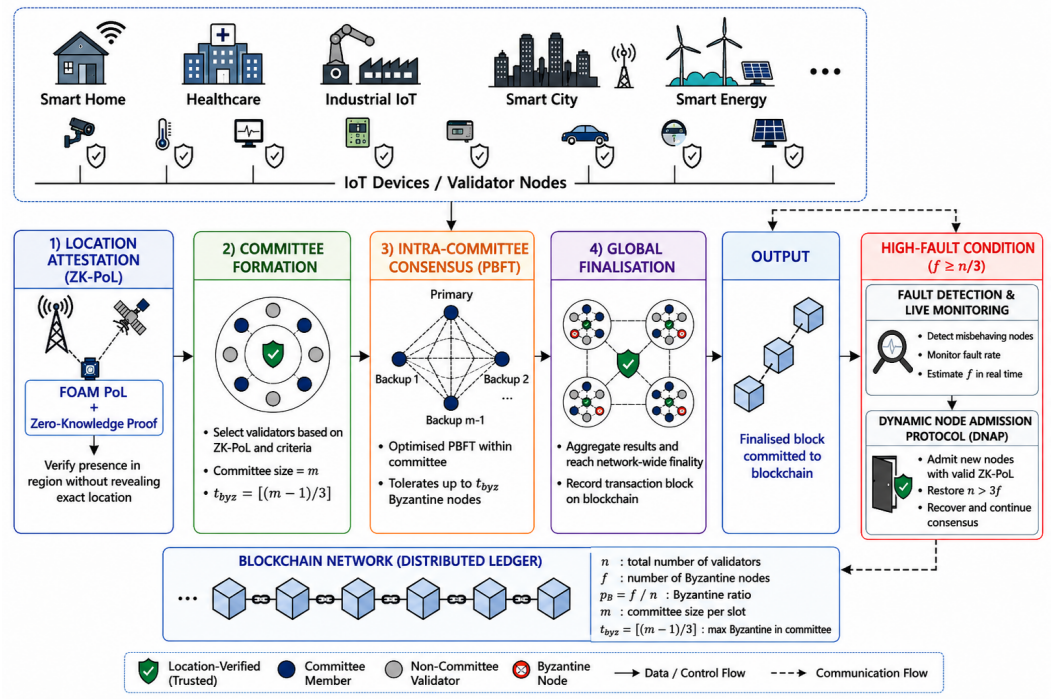

Figure 1. EPoLBFT System Architecture.

3.2. Elasticity Mechanism

In EPoLBFT, elasticity refers to the adaptive adjustment of consensus parameters according to network scale, geographic density, validator trust, mobility, and detected attack intensity. The elasticity mechanism consists of three main components: dynamic committee resizing, adaptive epoch duration, and variable geographic granularity.

The Dynamic committee resizing adjusts the committee size k according to the number of available trusted validators and the estimated Byzantine risk in a region. When validator churn, mobility, or attack intensity increases, the committee size may be increased to preserve fault tolerance. When the network remains stable, the committee size may be reduced to lower latency, energy consumption, and communication overhead.

Adaptive epoch duration controls how frequently location certificates, reputation scores, and committee membership are refreshed. Shorter epochs are used in high-mobility or high-risk conditions to prevent stale location proofs and compromised validators from remaining active for long periods. Longer epochs are used in stable environments to reduce re-attestation and committee reconfiguration overhead.

Variable geographic granularity allows regions to be split or merged according to node density and validator availability. Dense regions may be divided into smaller consensus zones to reduce local communication overhead, while sparse regions may be merged to ensure that each committee contains enough validators to satisfy the BFT requirement $k \geq 3f_c + 1$.

To improve reproducibility, EPoLBFT employs threshold-based adaptation rules for committee resizing, epoch adjustment, and geographic reconfiguration. Let $\mu \in [0, 1]$ denote the observed node mobility rate, $\beta \in [0, 1]$ the estimated Byzantine-risk score, and ρ the regional node density.

The committee size for epoch $e + 1$ is updated as

$$k_{e+1} = \begin{cases} \min(k_e + \Delta_k, k_{\max}), & \text{if } \mu > \mu_{th} \text{ or } \beta > \beta_{th}, \\ \max(k_e - \Delta_k, k_{\min}), & \text{if } \mu < \mu_{low} \text{ and } \beta < \beta_{low}, \\ k_e, & \text{otherwise.} \end{cases} \quad (1)$$

Similarly, the epoch duration is adapted according to network stability:

$$E_{e+1} = \begin{cases} \max(E_e/2, E_{\min}), & \text{if } \mu > \mu_{th} \text{ or } \beta > \beta_{th}, \\ \min(1.5E_e, E_{\max}), & \text{if } \mu < \mu_{low} \text{ and } \beta < \beta_{low}, \\ E_e, & \text{otherwise.} \end{cases} \quad (2)$$

In the default configuration, $\mu_{th} = 0.3$, $\beta_{th} = 0.25$, $\mu_{low} = 0.1$, and $\beta_{low} = 0.1$. Geographic regions are dynamically split when $\rho > \rho_{\max}$ and merged when the number of eligible validators falls below the BFT safety requirement $k < 3f_c + 1$. These rules allow EPoLBFT to maintain fault tolerance under adverse conditions while reducing communication and reconfiguration overhead during stable operation. Therefore, elasticity in EPoLBFT is not limited to leader replacement or view change. It refers to the protocol's ability to adapt its committee size, epoch length, and geographic consensus boundary according to changing IoT network conditions.

3.3. Location Attestation

At the beginning of each epoch, the designated primary node initiates the verification process by re-validating the signatures associated with the location certificates submitted by all participating validator nodes in the network. The certificate generation proceeds through a sequence of coordinated steps to demonstrate location authenticity without revealing its exact coordinates.

Step 1: Distance-Bounding and Location Estimation. Beacon B engages in a challenge–response protocol with nearby anchors. Each anchor A_i measures round-trip time or signal characteristics to derive a distance bound d_i. Using multilateration, anchors collaboratively verify that B's location, ℓ, satisfies geometric constraints relative to their positions. Instead of disclosing ℓ, anchors agree on a predicate $\mathcal{P}(\ell)$, e.g., $\ell \in Z$, where Z is a predefined zone.

Step 2: Anchor Attestation. Each anchor A_i produces a signed statement:

$$\sigma_i = \text{Sign}_{sk_i}(B \parallel Z \parallel T \parallel d_i),$$

where T is a timestamp. A threshold subset of anchors (e.g., $t \geq 2f + 1$) is required to ensure robustness. These signatures collectively form a location attestation set $\Sigma = \{\sigma_1, \ldots, \sigma_t\}$.

Step 3: Location Commitment. The beacon computes a cryptographic commitment to its location:

$$C = \text{Com}(\ell, r),$$

where r is randomness, ensuring both binding and hiding of the actual coordinates.

Step 4: Zero-Knowledge Proof Generation. The beacon constructs a ZK proof π attesting that:

- The committed value, C, opens to a location ℓ,
- ℓ satisfies the predicate $\mathcal{P}(\ell)$,
- ℓ is consistent with the anchor-derived distance bounds $\{d_i\}$,
- The attestation set Σ contains valid signatures from registered anchors.

Formally,

$$\pi = \text{ZK-Prove}(\ell, r, \Sigma : C = \text{Com}(\ell, r) \wedge \mathcal{P}(\ell) \wedge \text{Verify}(\Sigma))$$

Step 5: Certificate Issuance. The beacon submits (C, π, Σ) to a verifier set (e.g., blockchain validators or a distributed authority). Upon successful verification, a location certificate is issued:

$$\text{Cert}_B = \text{Sign}_{sk_V}(B \parallel C \parallel Z \parallel T).$$

This certificate serves as a publicly verifiable proof that node B was present in zone Z at time T, without revealing its exact coordinates as described in Algorithm 1. Each validator node provides an issued location certificate, which is verified to ensure authenticity, integrity, and consistency with the claimed geographic region. The primary node verifies that the signatures are valid and originate from registered validators, and a sufficient threshold of consistent attestations is satisfied.

Algorithm 1 Region Formation and Node Assignment

Require: Set of IoT nodes N **Ensure:** Partitioned regions $\mathcal{R} = \{R_1, R_2, \ldots, R_m\}$

```
for each node n_i ∈ N do
    Generate zk-PoL proof P_i
    Verify P_i using validator nodes
    if valid then
        Assign node n_i to region R_j based on geohash/location
    else
        Reject node or mark as untrusted
    end if
end for
    return Regional node sets R
```

Upon successful validation, the region location certificate R_lv is confirmed, establishing the node's eligibility within the designated zone for the current epoch. This process ensures the integrity of location claims while preserving privacy and making large-scale spoofing computationally infeasible.

3.4. Committee Formation

The network is partitioned into M geographically defined regions, referred to as committees $C_1, C_2, \ldots, C_M$, based on the validated locations of participating nodes. Nodes within a predefined physical proximity are grouped into the same committee. This geo-sharding strategy leverages the natural locality of IoT communications, ensuring that

the majority of communication occurs within a committee, thereby reducing latency and bandwidth consumption.

Within each committee, a leader (primary) is selected using a verifiable random function (VRF) that takes as input the node's encrypted Proof of Location (PoL) certificate and its historical reliability score. To prevent centralisation of authority and mitigate leader-based attacks, the role of the primary is rotated at each view change, ensuring fairness and fault tolerance over time.

Let n denote the size of a given committee and t the maximum number of Byzantine nodes it can tolerate, defined as $t = \lfloor (n-1)/3 \rfloor$. To guarantee safety and liveness under standard Byzantine fault assumptions, the committee size must satisfy:

$$n \geq 3t + 1. \tag{3}$$

3.5. Intra-Committee Consensus

After location verification and committee formation, each regional committee executes a local Byzantine Fault-Tolerant consensus process to order transactions generated within its geographic region. The purpose of this phase is to preserve the safety and finality properties of PBFT while reducing the communication cost associated with global consensus. Unlike classical PBFT, where all validators exchange messages across the entire network, EPoLBFT confines the pre-prepare, prepare, and commit phases to a smaller set of geographically verified and reputation-filtered validators. Within each committee, C_m, a modified three-phase PBFT protocol is run to order a block of transactions. The key modification is that the messages in the prepare and commit phases are only broadcast to members of C_m, not to the entire network, as described in Algorithm 2. This reduces the communication cost associated with local consensus from approximately $O(N^2)$ to $O(N^2/M)$ when the validators are distributed between M committees. Consequently, the overall communication complexity can be approximated as

$$C_{total} = O\left(\frac{N^2}{M}\right) + O(G^2), \tag{4}$$

where G denotes the size of the Global Finalisation Committee (GFC). Since $G \ll N$ in typical deployments, the additional overhead remains substantially lower than the cost of network-wide PBFT consensus. However, EPoLBFT also incurs an additional communication cost during global finalisation, where regional representatives participate in a secondary PBFT instance through the GFC. Consequently, the overall communication complexity can be approximated as

$$C_{total} = O\left(\frac{N^2}{M}\right) + O(G^2), \tag{5}$$

where G denotes the size of the Global Finalisation Committee. Since $G \ll N$ in typical deployments, the additional overhead remains substantially lower than the cost of network-wide PBFT consensus.

The protocol for a single committee includes the following steps:

1. Request: A client sends a transaction to the primary of a committee.
2. Pre-Prepare: The primary assigns a sequence number and multicasts a PRE-PREPARE message to all committee members.
3. Prepare: Upon receiving a valid PRE-PREPARE, each backup node multicasts a PREPARE message to all other nodes in the committee. After receiving $2f$ matching PREPARE messages (including its own), the node enters the prepared state.

4. Commit: Each node then multicasts a COMMIT message to the committee. Upon receiving $2f + 1$ matching COMMIT messages, the node commits the block, executes the transaction, and replies to the client. The expected latency is modelled with Equation (6):

$$T_{\text{intra}} = 2\Delta + c_m t_\sigma, \tag{6}$$

where Δ denotes the upper bound on one-way message delay within a committee. The term 2Δ accounts for the PREPARE and COMMIT communication rounds, while $c_m t_\sigma$ captures the local message processing overhead, where c_m is the number of committee messages processed and σ is the per-message verification cost.

Algorithm 2 Intra-Region Consensus (Local PBFT)

Require: Region R_i, node set N_i **Ensure:** Region-validated block B_i
Select regional committee $C_i \subseteq R_i$
Elect primary node P_i from C_i
for each transaction T in region R_i **do**
    Client sends request to P_i
    P_i broadcasts PRE-PREPARE message
    Replicas broadcast PREPARE messages
    if received $2f + 1$ PREPARE messages **then**
        Broadcast COMMIT messages
    end if
    if received $2f + 1$ COMMIT messages **then**
        Append T to region block B_i
    end if
end for
Cryptographically sign B_i using a committee **return** Region-validated block B_i

3.6. Global Finalisation

Once a committee commits a block, the committee leader creates a compact commitment for the block and broadcasts it to a designated GFC. Here, a *hash function* maps arbitrary-length input to a fixed-length digest and is assumed to be collision-resistant; a *Merkle tree* arranges hashes of the block's transactions in a binary hash tree so that the *Merkle root* succinctly commits to the full transaction set and supports efficient inclusion proofs as shown in Algorithm 3.

In this context, a *compact commitment* is a short digest (e.g., the block header together with the Merkle root) that binds to the block contents without transmitting the entire block. The GFC, composed of randomly selected leaders from different geographic regions, runs a standard PBFT instance to order these commitments into a final global ledger. This provides a verifiable audit trail and ensures global consistency across all shards. The final latency, T_{EPoLBFT}, is given by Equation (7)

$$T_{\text{EPoLBFT}} = T_{\text{att}} + T_{\text{form}} + T_{\text{intra}} + T_{\text{final}}, \tag{7}$$

where T_{att} is the time to generate and validate location attestations (node/region certificates), T_{form} is the committee-formation time (geo-sharding and leader selection), T_{intra} is the intra-committee PBFT latency defined above, and T_{final} is the global finalisation latency at the GFC.

Algorithm 3 Inter-Region Global Finalisation

Require: Set of region blocks $\{B_1, B_2, \ldots, B_m\}$ **Ensure:** Globally agreed block B_g
Form global committee C_g from regional representatives
for each region block B_i **do**
    Submit B_i to global committee
end for
Global committee verifies validity of B_i
Execute PBFT consensus among C_g:
    PRE-PREPARE → PREPARE → COMMIT
if consensus achieved **then**
    Aggregate region blocks into B_g
    Broadcast B_g to all nodes
end if
    return Finalised global block B_g

3.7. Fault Detection and Liveness Monitoring

Replicas use timers to detect liveness failures. If a block is not committed within a timeout period, the percentage of compromise can be modelled. Let n denote the total number of validators and f the number of Byzantine nodes, with the Byzantine ratio defined as $p_B = f/n$. For a committee of size m, the maximum number of tolerable Byzantine nodes is given by $t_{\text{byz}} = \lfloor (m-1)/3 \rfloor$. The probability of committee compromise under adversarial conditions is formally analysed in Equation (8). Equation (8) is based on the classical hypergeometric committee-selection model and assumes a uniform random selection of validators from the global validation pool.

$$\mathbb{P}[\text{compromise}] = \sum_{i=\lfloor (m-1)/3 \rfloor + 1}^{m} \frac{\binom{f}{i}\binom{n-f}{m-i}}{\binom{n}{m}}. \tag{8}$$

At this point, a VIEW-CHANGE protocol is triggered to select a new leader within the committee. If a predefined number of consecutive view changes fail, the protocol deduces that the committee is unable to form a quorum, indicating $f \geq n/3$. The node then transits into a "High-Fault State" and initiates the Dynamic Node Admission (DNAP) Process.

3.8. The DNAP Process

When a committee enters the High-Fault State due to repeated view-change failures or the inability to maintain a valid Byzantine quorum, EPoLBFT activates the Dynamic Node Admission Process (DNAP). The objective of DNAP is to restore the required fault-tolerance level by securely admitting new validators while preserving location integrity, trustworthiness, and consensus safety. The process consists of five phases.

1. Establishment of mutually authenticated connections: The establishment of secure communication channels involves active nodes forming mutually authenticated connections with pre-authorised "standby nodes". This process may utilise Trusted Execution Environments (TEE) to perform remote attestation, thereby assuring the integrity of the IoT systems involved.
2. Node Vetting and Admission: A standby node submits a join request. Crucially, this request includes a proof-of-Location claim that is verified by the active network members. Admission is granted only if a quorum of honest, active nodes approves the request and its location proof.
3. Reputation Score: Each standby node, i, maintains a local reputation score that summarises its recent behaviour as observed by the active committee. The score is used as an admission signal during vetting (in addition to identity checks, TEE

attestation, and Proof-of-Location). Concretely, a node is only eligible for admission if its reputation exceeds a minimum threshold $Rep_{\min}$ and a quorum of active nodes agrees on the reported value (e.g., by exchanging signed reputation reports over the same observation window). The reputation score is evaluated over a sliding observation window T (or the most recent k epochs) and is computed as

$$Rep_i = w_1 Uptime_i + w_2 Honesty_i - w_3 Penalty_i, \tag{9}$$

where $Uptime_i$ denotes the fraction of time that node i remains reachable and responsive within protocol deadlines, $Honesty_i$ represents the ratio of valid protocol actions successfully verified by other validators, and $Penalty_i$ captures detected misbehaviour, including contradictory votes, invalid signatures, failed location attestations, repeated timeout events, or other protocol violations. The weighting parameters satisfy

$$w_1 + w_2 + w_3 = 1, \tag{10}$$

where $w_1, w_2, w_3 \geq 0$. To account for temporal changes in behaviour and prevent permanent blacklisting due to transient faults, reputation values are updated at the end of each epoch using an Exponentially Weighted Moving Average (EWMA):

$$Rep_i^{(e+1)} = \alpha Rep_i^{(e)} + (1 - \alpha)\hat{Rep}_i^{(e)}, \tag{11}$$

where $\hat{Rep}_i^{(e)}$ is the reputation value calculated during epoch e, and $\alpha \in [0, 1]$ controls the influence of historical behaviour. Larger values of α favour long-term stability, whereas smaller values enable faster adaptation to recent node behaviour.

A standby node is eligible for admission only if its reputation exceeds a minimum threshold,

$$Rep_i \geq Rep_{\min}, \tag{12}$$

and a quorum of active validators agrees on the reported reputation value. Severe protocol violations trigger immediate penalties according to

$$Rep_i^{(e+1)} = \max(0, Rep_i^{(e)} - \delta), \tag{13}$$

where δ denotes the penalty factor associated with the violation. Nodes whose reputation falls below $Rep_{\min}$ are temporarily excluded from committee selection until sufficient compliant behaviour is observed.

The reputation score is subsequently used during validator admission, committee formation, and leader selection, ensuring that consensus participation is biased towards consistently honest and reliable nodes while limiting the influence of malicious or unstable validators.

4. Secure State Synchronisation: The new node synchronises its blockchain state by requesting the latest state from multiple existing nodes and verifying it via a quorum of cryptographically signed checkpoint hashes.
5. Network Re-initialisation: This process repeats until enough new, location-verified nodes have been admitted so that the post-recovery committee size n', i.e., the number of active replicas after admission, satisfies the standard BFT safety bound against up to f Byzantine (arbitrarily faulty) replicas, namely $n' \geq 3f + 1$. The network then exits the High-Fault State and resumes the standard consensus protocol.

4. Performance Evaluation

This section evaluates the performance of the proposed EPoLBFT consensus framework in comparison with representative state-of-the-art consensus mechanisms for IoT environments. The evaluation focuses on scalability, latency, throughput, communication overhead, and security resilience, which are critical performance indicators for resource-constrained and dynamic IoT systems.

4.1. Experimental Setup

The performance evaluation was conducted using the Blockchain IoT Consensus Algorithm (BICA) simulator, which enables controlled experimentation of blockchain consensus protocols under heterogeneous IoT network conditions. The evaluation compares the proposed EPoLBFT framework with PBFT, G-PBFT, Directed Acyclic Graph Byzantine Fault Tolerance (DAG-BFT), and Distributed Committee Byzantine Fault Tolerance (DCBFT) using identical workload, network, and adversarial assumptions. The objective of the evaluation is to measure the effect of location-aware committee formation, zk-PoL verification, and elastic validator selection on latency, throughput, communication overhead, energy consumption, scalability, and attack resilience.

The simulated network consists of heterogeneous IoT devices, edge/fog gateways, and validator nodes operating under a partially synchronous communication model. IoT devices generate transactions and location-attestation requests, edge/fog gateways support computationally expensive cryptographic operations where required, and validator nodes execute proof verification and BFT consensus. The main simulation experiments vary the number of nodes from 200 to 1000. For committee-based protocols, the default committee size is set to $k = 21$. According to the standard BFT bound, each committee can tolerate up to

$$f_c = \left\lfloor \frac{k-1}{3} \right\rfloor = 6$$

Byzantine validators while preserving safety and liveness. This corrects the committee fault-tolerance threshold and ensures consistency with the condition $k \geq 3f_c + 1$. Sensitivity experiments are also conducted using different committee sizes to evaluate the effect of committee configuration on latency, throughput, and security resilience.

Each experiment was repeated over 30 independent simulation runs using fixed random seeds to improve reproducibility. Results are reported as mean values with standard deviations and 95% confidence intervals. Statistical significance between EPoLBFT and each baseline protocol was assessed using paired Student's t-tests across the 30 runs. Differences were considered statistically significant at the 95% confidence level ($p < 0.05$). Error bars in the latency, throughput, communication overhead, and energy consumption figures represent the variability observed across simulation runs.

The simulation workload consists of transaction generation, location verification, validator selection, committee consensus, and block finalisation. Transaction generation rates are varied to represent normal and high-load IoT conditions. Network delay is modelled using a partially synchronous communication assumption, where messages may be delayed but are eventually delivered within a bounded delay under normal operation. Byzantine behaviour, Sybil identities, location-spoofing attempts, and denial-of-service conditions are introduced in dedicated security scenarios.

Table 1 summarises the main simulation parameters used in the evaluation.

Table 1. Simulation parameters used in the performance evaluation.

Parameter	Value/Description
Simulator	BICA simulator
Network size	200, 400, 600, 800, 1000 nodes
Consensus protocols	PBFT, G-PBFT, DAG-BFT, DCBFT, EPoLBFT
Committee size	$k = 21$ by default
Byzantine tolerance	$f_c = \lfloor (k-1)/3 \rfloor = 6$
Network model	Partially synchronous
Number of runs	30 independent runs
Random seeds	Fixed seed set for reproducibility
Reported values	Mean, standard deviation, 95% confidence interval
Significance level	$p < 0.05$
Attack scenarios	Byzantine, Sybil, spoofing, DoS
Mobility scenarios	Low, medium, and high mobility
Churn scenarios	Node joining, leaving, and temporary failure

4.1.1. Hardware and Deployment Assumptions

The evaluation assumes three classes of devices: constrained IoT nodes, edge/fog gateways, and validator nodes. Constrained IoT nodes are responsible for transaction generation, sensing, and location-attestation requests. Edge/fog gateways provide optional support for computationally expensive operations such as zk-proof generation. Validator nodes perform zk-PoL verification, reputation evaluation, committee participation, and block finalisation.

The resource-constrained assumption applies mainly to IoT sensing devices, which are not expected to execute all cryptographic and consensus operations locally. Instead, EPoLBFT supports edge-assisted proof generation, while validators and gateways perform heavier verification and consensus tasks. This deployment model reflects practical IoT environments, where constrained devices are commonly supported by nearby gateways or edge servers.

Table 2 summarises the hardware assumptions used in the evaluation.

Table 2. Hardware assumptions for simulated IoT and validator entities.

Entity	Role	Assumption
IoT device	Transaction generation and sensing	Low CPU, low memory, battery-powered
Edge/fog gateway	zk-proof support and aggregation	Moderate CPU and memory
Validator node	zk-PoL verification and consensus	Higher compute capacity than IoT nodes
Location anchor	Location attestation	Fixed or semi-fixed trusted/verifiable node

4.1.2. zk-PoL Implementation and Overhead

The zk-PoL component is modelled using a zk-SNARK. This is chosen over other zero-knowledge algorithms, such as zk-SHARK, Bullet Proof, etc., because it offers very short proofs and fast verification times, making it highly efficient and scalable for real-world applications. zk-SNARK with Groth16 is selected because it provides compact proofs and efficient verification, making it suitable for bandwidth-constrained IoT and edge-assisted environments. The proof circuit verifies that a node satisfies a regional location predicate without revealing its exact coordinates.

The public inputs to the zk-PoL verifier include the region identifier, epoch identifier, proof commitment, and validator challenge. The private witness includes the node's location evidence, anchor attestations, timestamp, nonce, and location certificate. A proof is accepted only if the node can demonstrate that it belongs to the permitted region during the valid epoch.

The zk-PoL overhead is reported separately from ordinary consensus latency. This prevents the cost of privacy-preserving location verification from being hidden inside general consensus processing. In the evaluation, Table 3 captures zk-PoL overhead metrics adopted and their value.

Since proof generation may be expensive for highly constrained IoT devices, EPoLBFT permits edge-assisted proof generation. However, proof verification remains lightweight enough to be performed by validator nodes during committee admission. Valid location proofs are reused within a bounded epoch to avoid repeated proof generation for every transaction.

Table 3. zk-PoL overhead metrics.

Metric	Reported Value
Proof system	Groth16 zk-SNARK
Curve	BN254
Proof size	0.2 KB
Proof generation time	12.5 ms
Proof verification time	5.8 ms
Memory usage	1 KB
Proof validity	Epoch-bound
Reverification trigger	Epoch expiry, mobility threshold, or trust-risk event

4.2. Baseline and Comparative Protocols

To evaluate the performance of EPoLBFT, the proposed framework is compared with four baseline consensus protocols: PBFT, G-PBFT, DAG-BFT and DCBFT. These protocols were selected because they represent different approaches to Byzantine fault tolerance, location-aware consensus, graph-based ordering, and distributed committee-based consensus in IoT blockchain environments. The same workload, network conditions, transaction generation model, and attack assumptions are applied to all protocols to ensure a fair comparison.

4.2.1. PBFT

Practical Byzantine Fault Tolerance (PBFT) is used as the classical BFT baseline. In PBFT, all validators participate in the pre-prepare, prepare, and commit phases. A block is finalised only when at least $2f + 1$ matching commit messages are received, where f is the maximum number of Byzantine validators tolerated by the network. Although PBFT provides deterministic finality and strong safety guarantees under the condition $n \geq 3f + 1$, its communication complexity increases quadratically with the number of validators. This makes PBFT expensive for large-scale IoT environments with many resource-constrained devices.

4.2.2. G-PBFT

Geographic PBFT (G-PBFT) is included as a location-aware baseline. G-PBFT improves classical PBFT by using geographic grouping or location-based validator organisation to reduce global communication overhead. It is relevant to EPoLBFT because both protocols exploit the physical distribution of IoT nodes. However, G-PBFT does not provide privacy-preserving location verification using zero-knowledge proofs and does not explicitly integrate elastic committee resizing, adaptive epoch duration, or reputation-aware validator filtering.

4.2.3. DAG-BFT

DAG-BFT represents graph-based BFT consensus, where transactions or blocks are structured as a directed acyclic graph rather than a single linear chain. DAG-based

consensus can improve concurrency and throughput by allowing multiple proposals or transaction paths to progress in parallel. However, DAG-BFT does not directly address privacy-preserving location validation or physical-context-aware validator selection. It is therefore included to compare EPoLBFT against a high-throughput non-location-aware BFT approach.

4.2.4. DCBFT

DCBFT is included as a distributed or committee-based BFT baseline. It reduces the cost of classical PBFT by distributing consensus responsibilities across smaller groups of validators. This makes it relevant for evaluating the effect of committee partitioning on latency, throughput, and communication overhead. However, unlike EPoLBFT, DCBFT does not integrate zk-PoL, location-bound validator eligibility, mobility-aware reattestation, or reputation-based location trust.

4.2.5. EPoLBFT

EPoLBFT is the proposed protocol. It combines zero-knowledge Proof-of-Location, reputation-aware validator selection, geo-sharded committee formation, elastic committee adaptation, and local BFT consensus. In EPoLBFT, validators must prove regional eligibility without revealing exact coordinates. Consensus is then executed within smaller verified committees rather than across the entire network. This design aims to reduce communication overhead, improve scalability, preserve location privacy, and increase resilience against Sybil, Byzantine, location-spoofing, and denial-of-service attacks.

Table 4 summarises the main features of the protocols used in the experimental evaluation.

Table 4. Baseline protocols used in the experimental evaluation.

Protocol	BFT	Location-Aware	zk-PoL	Committee-Based	Elastic
PBFT	Yes	No	No	No	No
G-PBFT	Yes	Yes	No	Partial	No
DAG-BFT	Yes	No	No	No	No
DCBFT	Yes	No	No	Yes	Partial
EPoLBFT	Yes	Yes	Yes	Yes	Yes

4.3. Evaluation Metrics

The following metrics are used to evaluate performance:

- Consensus Latency: Average time required to finalise a transaction.
- Throughput: Number of transactions processed per second (TPS).
- Communication Overhead: Number of consensus-related messages exchanged.
- Energy Efficiency: Estimated computational and communication energy cost.
- Security Resilience: Ability to maintain performance under Byzantine, Sybil, and location-spoofing attacks.

4.4. Scalability Analysis

Figure 2 illustrates the variation of consensus latency as the network size increases. Standard PBFT exhibits a rapid increase in latency due to its $O(N^2)$ communication complexity, becoming impractical beyond moderate network sizes. In contrast, EPoLBFT maintains stable latency as network size grows, with little changes arising from the global finalisation of the consensus, since consensus communication is confined to a small committee.

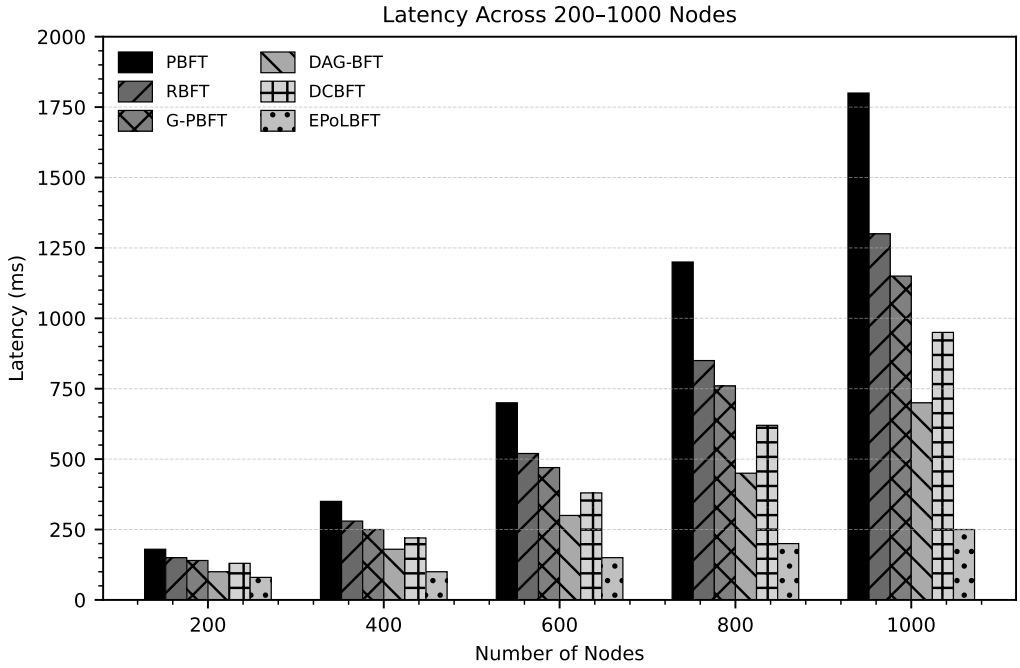

Figure 2. Latency comparison of EPoLBFT and selected BFT algorithms under increasing network sizes.

Similarly, throughput results in Figure 3 demonstrate that EPoLBFT achieves consistently higher transaction throughput compared to PBFT and G-PBFT. DAG-BFT shows competitive throughput at large scales; however, it exhibits increased variance due to its asynchronous confirmation structure.

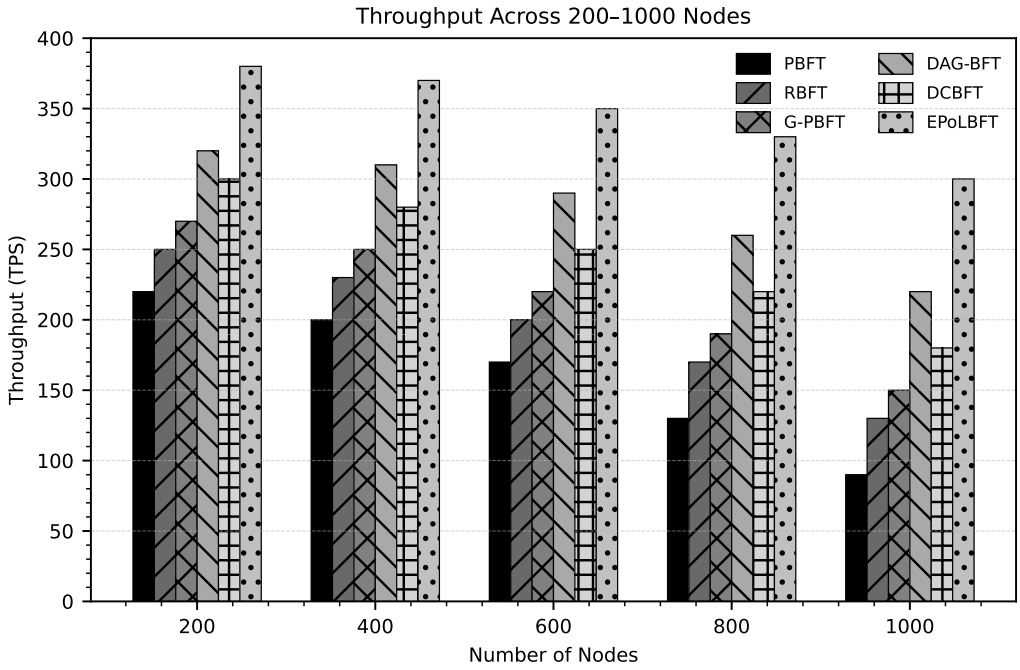

Figure 3. Throughput comparison of EPoLBFT and selected BFT algorithms under increasing network sizes.

4.5. Communication Overhead

The communication overhead of EPoLBFT is significantly lower than that of standard PBFT. By limiting consensus participation to a committee of size k, EPoLBFT reduces message complexity from $O(N^2)$ to $O(k^2)$. Although zk-PoL introduces additional verification

messages during committee formation, this overhead is amortised across epochs and does not impact per-transaction consensus performance.

4.6. Energy Efficiency

Energy consumption is estimated using a standard IoT energy model based on the number of cryptographic operations and communication messages exchanged during consensus execution. Specifically, the model accounts for the energy cost of hash computations, digital signature verification, zero-knowledge proof verification, and message transmission, following widely adopted assumptions in IoT consensus evaluation. While zk-PoL proof generation incurs a higher computational cost, this operation is performed infrequently during epoch transitions and does not impact per-transaction consensus latency.

In general, EPoLBFT shows lower energy consumption, as shown in Figure 4. The blue shaded region (600–1000 nodes) marks the large-scale range where differences in energy usage become clearer—PBFT and G-PBFT rise more sharply due to extra communication rounds and near-quadratic message growth, whereas EPoLBFT grows more gradually because committee-based consensus limits communication and signature/verification overhead at scalegroups.

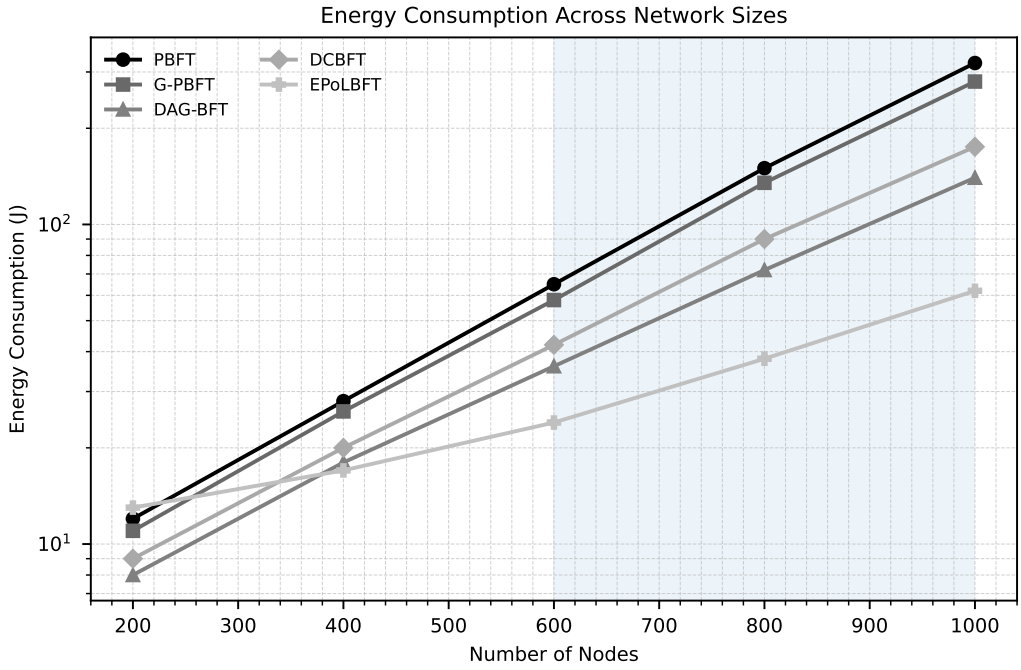

Figure 4. Energy consumption comparison for EPoLBFT and selected BFT algorithms.

5. Security Analysis

This section analyses the security properties of the proposed EPoLBFT framework. The analysis focuses on resistance to Byzantine behaviour, Sybil attacks, location spoofing, and denial-of-service (DoS) attacks, as well as the preservation of location privacy. The security guarantees of EPoLBFT are derived from the integration of Byzantine Fault Tolerance (BFT), region-based consensus partitioning, and cryptographically verifiable privacy-preserving Proof-of-Location (zk-PoL). The analytical findings are further validated through experimental results, as illustrated in Figure 5, which presents a comparative evaluation of security performance across multiple consensus protocols.

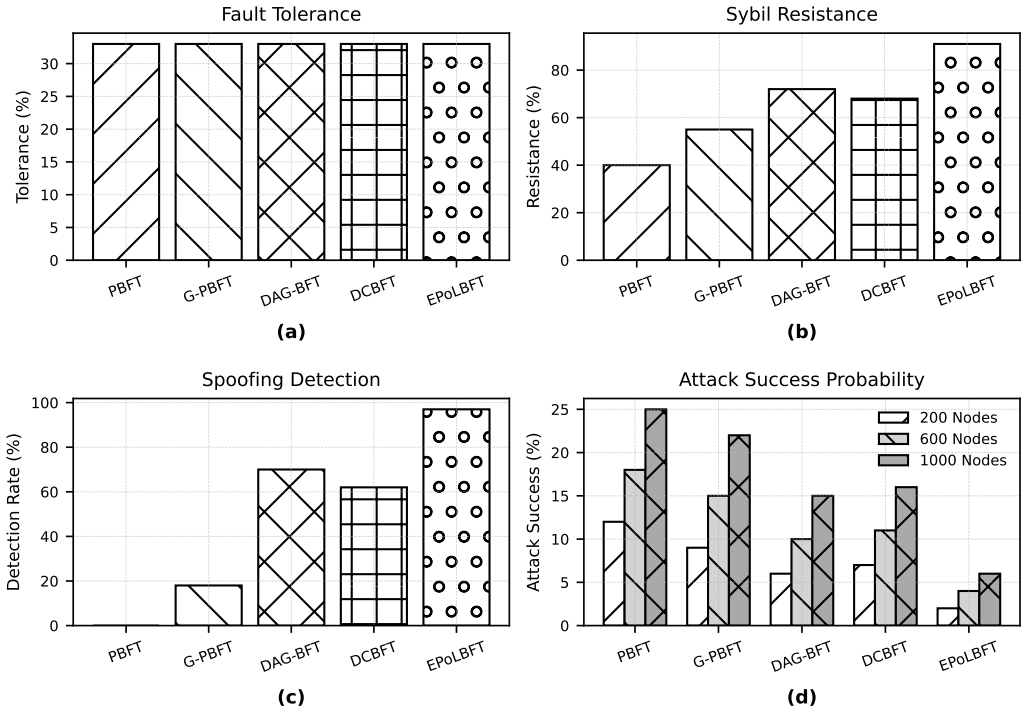

Figure 5. Security performance comparison across consensus protocols showing (**a**) theoretical Byzantine fault tolerance, (**b**) Sybil resistance, (**c**) spoofing detection, and (**d**) attack success probability for different network sizes.

5.1. Byzantine Fault Tolerance

EPoLBFT preserves the classical Byzantine fault tolerance guarantees of Practical Byzantine Fault Tolerance (PBFT). Within each consensus committee of size k, the protocol tolerates up to $f_c < \frac{k-1}{3}$ Byzantine nodes without compromising safety or liveness. This ensures that all honest nodes agree on the same sequence of transactions. Unlike traditional PBFT, which relies on a single global committee, EPoLBFT introduces a hierarchical consensus structure consisting of intra-region consensus and inter-region global finalisation. Each region independently satisfies the BFT constraint, thereby localising adversarial influence.

It is important to emphasise that EPoLBFT does not increase the theoretical Byzantine threshold beyond $\frac{n}{3}$. Instead, its enhanced security arises from fault isolation across regions. Adversaries must simultaneously compromise multiple independent regional committees to affect global consensus, which significantly increases the difficulty of coordinated attacks. As shown in Figure 5a, all evaluated protocols adhere to the classical Byzantine fault tolerance bound of $f < \frac{n}{3}$. EPoLBFT maintains this theoretical guarantee while improving practical resilience through its hierarchical and region-based consensus structure.

5.2. Sybil Attack Resistance

EPoLBFT mitigates Sybil attacks by binding node participation to cryptographic Proof-of-Location (zk-PoL). Each node must prove physical presence within a valid geographic region before being admitted into consensus. Since zk-PoL requires verifiable location evidence and witness attestation, creating multiple identities requires proportional physical deployment effort. This introduces a strong economic and spatial constraint on adversaries.

Furthermore, regional partitioning limits the concentration of identities within a single consensus group. Even if multiple Sybil identities are created, their influence is restricted to specific regions, preventing global dominance. This improvement is reflected in Figure 5b,

where EPoLBFT demonstrates significantly higher Sybil resistance compared to baseline protocols due to the integration of zk-PoL and physical verification constraints.

5.3. Resistance to Location Spoofing

Location-spoofing attacks are mitigated through the zk-PoL mechanism, which enables nodes to prove membership within a geographic region without revealing exact coordinates. Unlike traditional systems relying on raw location data, zk-PoL encodes spatial constraints into a cryptographic proof, ensuring authenticity and non-forgeability.

Additionally, proofs are time-bound and tied to witness validation, preventing replay attacks and impersonation. The region-based architecture further strengthens this protection, as adversaries must forge valid proofs across multiple regions to influence global consensus. Figure 5c shows that EPoLBFT achieves substantially higher spoofing detection rates, highlighting the effectiveness of zk-PoL and witness-based verification in preventing falsified location claims.

5.4. Privacy Preservation

EPoLBFT ensures strong location privacy through zero-knowledge verification. Nodes prove compliance with regional constraints without disclosing sensitive spatial data such as exact coordinates or movement history.

This guarantees:

- **Confidentiality:** Precise location information is never revealed.
- **Unlinkability:** Repeated participation does not expose movement patterns.
- **Protection against targeting:** Adversaries cannot identify specific device locations.

These properties are essential in IoT scenarios such as smart homes and healthcare systems, where location data is highly sensitive.

5.5. Denial-of-Service Attack Mitigation

EPoLBFT mitigates denial-of-service attacks through both structural and protocol-level mechanisms. By partitioning the network into regions, consensus communication is localised, reducing the impact of traffic flooding and message amplification attacks.

The dynamic beacon mechanism further enhances robustness by rotating validation roles among nodes, preventing adversaries from targeting fixed participants. Additionally, the use of a two-stage consensus process ensures that disruptions in one region do not halt system-wide progress, preserving overall availability.

5.6. Committee Capture and Collusion Resistance

Committee capture and collusion are mitigated through dynamic and location-aware committee selection. At the end of each epoch:

- zk-PoL proofs are refreshed,
- reputation scores are updated,
- committee membership is re-evaluated.

Geographic diversity constraints ensure that committee members are distributed across regions, reducing the feasibility of coordinated collusion. Since both intra-region and inter-region consensus steps are required for finalisation, adversaries must control multiple independent committees simultaneously, which is significantly more difficult than compromising a single committee.

6. Conclusions and Future Work

This paper proposes an enhanced consensus mechanism designed to address the limitations of conventional BFT protocols in a resource-constrained Internet-of-Things

(IoT) environment. Unlike traditional PBFT-based schemes, EPoLBFT introduces an elastic quorum adjustment and zk-PoL-based reputation system that dynamically optimises consensus performance while maintaining strong Byzantine resilience. Extensive simulations across IoT nodes demonstrate that EPoLBFT achieves a superior trade-off between latency, throughput, and security fault tolerance. The elastic view-change mechanism of the algorithm effectively minimises leader re-election overhead, while PoL-based trust scoring prevents the participation of unverified or malicious nodes, thereby enhancing consensus stability. Though the proposed EPoLBFT framework demonstrates promising performance through simulation-based evaluation, several aspects require further investigation. First, the current study assumes a trusted initialisation authority for node identity verification and reputation bootstrapping. Also, the simulation-based validation provides a strong preliminary assessment of the proposed framework. Future work will focus on implementing and prototyping EPoLBFT in real-world IoT environments to further evaluate its practical feasibility and performance. In particular, experimental deployment on heterogeneous IoT devices and edge-computing platforms will be conducted to assess scalability, communication overhead, energy consumption, and resilience under realistic operating conditions. Future research will also investigate adaptive consensus parameter optimisation, enhanced reputation management mechanisms, and integration with emerging IoT security frameworks to further improve the efficiency, robustness, and applicability of EPoLBFT in large-scale decentralised IoT ecosystems.

Author Contributions: Conceptualization, Y.K., D.D. and E.G.; methodology, Y.K. and D.D.; validation, D.D. and E.G.; formal analysis, Y.K., D.D. and E.G.; investigation, Y.K., D.D. and E.G.; resources, Y.K., D.D. and E.G.; data curation, Y.K.; writing—original draft preparation, Y.K.; writing—review and editing, Y.K., D.D. and E.G.; visualization, Y.K. and D.D.; supervision, D.D. and E.G.; project administration, D.D.; funding acquisition, D.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the University of the West of England, Bristol Studentship.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: During the preparation of this manuscript, the authors used GPT 5.6 by OpenAI for language refinement, grammar checking, and academic editing support. Sincere appreciation to the reviewers for their insightful comments.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

IoT	Internet of Things
BC	Blockchain
PoW	Proof of Work
PoS	Proof of Stake
PoL	Proof of Location
BFT	Byzantine Fault Tolerance
PBFT	Practical Byzantine Fault Tolerance
TPS	Throughput Per seconds
DoS	Denial of Service
ECC	Elliptic Curve Cryptography
EPoLBFT	Elastic Proof of Location Byzantine Fault Tolerance
zK	Zero Knowledge

SNARK	Succinct Non-Interactive Arguments of Knowledge
G-PBFT	Geographic Practical Byzantine Fault Tolerance
DAG-BFT	Directed Acyclic Graph Byzantine Fault Tolerance
DCBFT	Distributed Committee Byzantine Fault Tolerance
BICA	Blockchain IoT Consensus Algorithm
VRF	Verifiable Random Function
GFC	Global Finalisation Committee
TEE	Trusted Execution Environment
DNAP	Dynamic Node Admission Process
EWMA	Exponentially Weighted Moving Average

References

1. Bataineh, M.R.; Mardini, W.; Khamayseh, Y.M.; Yassein, M.M.B. Novel and Secure Blockchain Framework for Health Applications in IoT. *IEEE Access* **2022**, *10*, 14914–14926. [CrossRef]
2. Bourian, I.; Sebbar, A.; Chougdali, K.; Amhoud, E.M. SSHCEth: Secure Smart Home Communications based on Ethereum Blockchain and Smart Contract. In *Proceedings of the GLOBECOM 2023—2023 IEEE Global Communications Conference*; IEEE: Piscataway, NJ, USA, 2023; pp. 2674–2679. [CrossRef]
3. Azizi, H. A Brief Review on Wireless Networks. *Int. Res. J. Eng. Technol.* **2017**, *4*, 329–333. [CrossRef]
4. Alshehri, S.; Bamasag, O. AAC-IoT: Attribute Access Control Scheme for IoT Using Lightweight Cryptography and Hyperledger Fabric Blockchain. *Appl. Sci.* **2022**, *12*, 8111. [CrossRef]
5. Kareem, Y.; Djenouri, D.; Ghadafi, E. A Survey on Emerging Blockchain Technology Platforms for Securing the Internet of Things. *Future Internet* **2024**, *16*, 285. [CrossRef]
6. Singh, M.; Singh, A.; Kim, S. Blockchain: A game changer for securing IoT data. In *Proceedings of the IEEE World Forum on Internet of Things, WF-IoT 2018—Proceedings*; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2018; Volume 2018, pp. 51–55. [CrossRef]
7. Pan, J.; Song, Z.; Hao, W. Development in Consensus Protocols: From PoW to PoS to DPoS. In Proceedings of the 2021 2nd International Conference on Computer Communication and Network Security (CCNS), Xining, China, 30 July–1 August 2021; pp. 59–64. [CrossRef]
8. Misic, J.; Misic, V.B.; Chang, X.; Ali, S.; Kulkarni, J. Multiple entry point PBFT for IoT systems. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), Taipei, Taiwan, 7–11 December 2020; pp. 1–6. [CrossRef]
9. Gao, N.; Huo, R.; Wang, S.; Liu, J.; Huang, T.; Liu, Y. SBFT: A BFT consensus mechanism based on DQN algorithm for industrial Internet of Thing. *China Commun.* **2023**, *20*, 185–199. [CrossRef]
10. Xu, G.; Bai, H.; Xing, J.; Luo, T.; Xiong, N.N.; Cheng, X.; Liu, S.; Zheng, X. SG-PBFT: A secure and highly efficient distributed blockchain PBFT consensus algorithm for intelligent Internet of vehicles. *J. Parallel Distrib. Comput.* **2022**, *164*, 1–11. [CrossRef]
11. Kumar, A.; Vishwakarma, L.; Das, D. R-PBFT: A secure and intelligent consensus algorithm for Internet of vehicles. *Veh. Commun.* **2023**, *41*, 100609. [CrossRef]
12. Liu, X.; Liu, Y.; Li, X.; Cao, H.; Wang, Y. FP-BFT: A fast pipeline Byzantine consensus algorithm. *IET Blockchain* **2023**, *3*, 123–135. [CrossRef]
13. Lao, L.; Dai, X.; Xiao, B.; Guo, S. G-PBFT: A Location-based and Scalable Consensus Protocol for IoT-Blockchain Applications. In *Proceedings of the Proceedings—2020 IEEE 34th International Parallel and Distributed Processing Symposium, IPDPS 2020*; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2020; pp. 664–673. [CrossRef]
14. Ledwaba, L.P.I.; Hancke, G.P.; Mitrokotsa, A.; Isaac, S.J. A Delegated Proof of Proximity Scheme for Industrial Internet of Things Consensus. In Proceedings of the IECON 2020 the 46th Annual Conference of the IEEE Industrial Electronics Society, Singapore, 18–21 October 2020; pp. 4441–4446. [CrossRef]
15. Brambilla, G.; Amoretti, M.; Zanichelli, F. Using Block Chain for Peer-to-Peer Proof-of-Location. *arXiv* **2016**. [CrossRef]
16. Kareem, Y.; Djenouri, D.; Ghadafi, E. Blockchain Simulator for Consensus Algorithms and Security Testing in Future IoT. In Proceedings of the 2025 IEEE 36th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Istanbul, Turkey, 1–4 September 2025; pp. 1–6. [CrossRef]
17. Chen, Y.; Zhang, Y.; Zhuang, Y.; Miao, K.; Pouriyeh, S.; Han, M. Efficient and Secure Blockchain Consensus Algorithm for Heterogeneous Industrial Internet of Things Nodes Based on Double-DAG. *IEEE Trans. Ind. Inform.* **2024**, *20*, 6300–6312. [CrossRef]
18. Guo, R.; Guo, Z.; Lin, Z.; Jiang, W. A hierarchical byzantine fault tolerance consensus protocol for the Internet of Things. *High-Confid. Comput.* **2024**, *4*, 100196. [CrossRef]

19. Sukhwani, H.; Martínez, J.M.; Chang, X.; Trivedi, K.S.; Rindos, A. Performance modeling of PBFT consensus process for permissioned blockchain network (hyperledger fabric). In *Proceedings of the IEEE Symposium on Reliable Distributed Systems*; IEEE Computer Society: Piscataway, NJ, USA, 2017; Volume 2017, pp. 253–255. [CrossRef]
20. Wu, W.; Liu, E.; Gong, X.; Wang, R. *Blockchain Based Zero-Knowledge Proof of Location in IoT*; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2020.
21. Brambilla, G.; Amoretti, M.; Medioli, F.; Zanichelli, F. Blockchain-based Proof of Location. In *Proceedings of the 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*; IEEE: Piscataway, NJ, USA, 2020. [CrossRef]
22. Tang, S.; Wang, Z.; Jiang, J.; Ge, S.; Tan, G.F. Improved PBFT algorithm for high-frequency trading scenarios of alliance blockchain. *Sci. Rep.* **2022**, *12*, 4426. [CrossRef] [PubMed]
23. Onireti, O.; Zhang, L.; Imran, M. On the Viable Area of Wireless Practical Byzantine Fault Tolerance (PBFT) Blockchain Networks. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6. [CrossRef]
24. Meshcheryakov, Y.; Melman, A.; Evsutin, O.; Morozov, A. On Performance of PBFT Blockchain Consensus Algorithm for IoT-Applications with Constrained Devices. *IEEE Access* **2021**, *9*, 80559–80570. [CrossRef]
25. Yuan, X.; Luo, F.; Haider, M.Z.; Buchanan, C. Efficient Byzantine Consensus Mechanism Based on Reputation in IoT Blockchain. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 9952218. [CrossRef]
26. Niu, K.; Yao, Z.; Si, X. Improved PBFT Consensus Algorithm Based on Reputation Value for IoT. In Proceedings of the IEEE International Conference Blockchain Cryptocurrency Intelligent System (ICBCTIS), Copenhagen, Denmark, 19–22 August 2024; pp. 164–168. [CrossRef]
27. Ramos, G.; Pequito, S.; Silvestre, D. Reputation-based resilient consensus with privacy guarantees. *IEEE Trans. Control Netw. Syst.* **2025**, *12*, 2381–2391. [CrossRef]
28. Fang, Y.; Pan, H.; Wang, X.; Wang, L.; Li, M. A secure and highly efficient blockchain PBFT consensus algorithm for microgrid power trading. *Sci. Rep.* **2024**, *14*, 8065. [CrossRef] [PubMed]
29. Pathak, A.; Al-Anbagi, I.; Hamilton, H.J. Blockchain-Enhanced Zero Knowledge Proof-Based Privacy-Preserving Mutual Authentication for IoT Networks. *IEEE Access* **2024**, *12*, 118618–118636. [CrossRef]
30. Chen, T.; Lu, H.; Kunpittaya, T.; Luo, A. A Review of zk-SNARKs. *arXiv* **2023**, arXiv:2202.06877.
31. Alanazi, F.; Zareei, M.; Arreola, A.R. PRIVOT: Privacy-Resilient Intelligent DAG Blockchain Architecture for IoT. *IEEE Access* **2025**, *13*, 8493–8506. [CrossRef]
32. Hamza, R.; Alotaibi, A.; Muhammad, K. A practical multi-layered framework for post-quantum secure machine learning. *Eng. Appl. Artif. Intell.* **2026**, *163*, 113044. [CrossRef]
33. Oude Roelink, B.; El-Hajj, M.; Sarmah, D. Systematic review: Comparing zk-SNARK, zk-STARK, and bulletproof protocols for privacy-preserving authentication. *Secur. Priv.* **2024**, *7*, e401. [CrossRef]
34. Geng, T.; Njilla, L.; Huang, C.T. Delegated Proof of Secret Sharing: A Privacy-Preserving Consensus Protocol Based on Secure Multiparty Computation for IoT Environment. *Network* **2022**, *2*, 66–80. [CrossRef]
35. Ambrosin, M.; Braca, P.; Conti, M.; Lazzeretti, R. ODIN: Obfuscation-based privacy preserving consensus algorithm for Decentralized Information fusion in smart device Networks. *arXiv* **2016**, arXiv:1610.07046.
36. Thakker, J.; Park, Y. Resilient and Efficient Blockchain Consensus Protocol for Internet-of-Things. In Proceedings of the IEEE International Conference Consumer Electronics (ICCE), Las Vegas, NV, USA, 4–6 January 2020; pp. 1–4. [CrossRef]
37. Pelekoudas-Oikonomou, F.; Zachos, G.; Mantas, G.; Komninos, D.; Rodriguez, J. A Scalable Approach of Practical Byzantine Fault Tolerance Algorithms for IoMT Blockchains. In Proceedings of the IEEE International Mediterranean Conference on Communications and Networking (MeditCom), Athens, Greece, 5–8 September 2022; pp. 156–161. [CrossRef]
38. Haider, M.Z.; Saeed, M.; Mostafa, A.; Mushtaq, M.F.; Chaudhry, U.; Ul-Hasan, A. Secure Trust-Based Delegated Consensus for Blockchain Frameworks Using Deep Reinforcement Learning. *IEEE Access* **2022**, *10*, 121406–121421. [CrossRef]
39. Heo, G.; Yang, D.; Doh, I.; Chae, K. Efficient and Secure Blockchain System for Digital Content Trading. *IEEE Access* **2021**, *9*, 73220–73228. [CrossRef]