Preprints.org

Article

Not peer-reviewed version

A Novel Position-Based Commitment Protocol for Secure Multi-Party Verification with Hydraulic-Inspired Mathematical Obfuscation

Manideep Thotakura *

Posted Date: 8 August 2025

doi: 10.20944/preprints202508.0584.v1

Keywords: cryptographic protocols; multi-party computation; commitment schemes; hydraulic inspired cryptography; supply chain security; consensus mechanisms; privacy-preserving verification

Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2025 doi:10.20944/preprints202508.0584.v1

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

A Novel Position-Based Commitment Protocol for Secure Multi-Party Verification with Hydraulic-Inspired Mathematical Obfuscation

Manideep Thotakura

Independent Researcher; manideep9949466287@gmail.com

Abstract

This work presents a cryptographic protocol for secure multi-party verification that achieves computational privacy while maintaining exceptional computational efficiency. The proposed Position-Based Commitment Protocol (PBCP) introduces a position-dependent nonce mechanism combined with cyclic verification architecture, enabling secure computation over private inputs without revealing individual parameters. Unlike existing commitment schemes that require complex cryptographic assumptions, computationally expensive zero-knowledge proofs, or extensive public key infrastructure, Fundamental innovation lies in adapting physical laws of fluid dynamics to create natural mathematical relationships where each verification equation contains multiple unknowns, making parameter extraction computationally infeasible while preserving verification integrity. The protocol preliminary analysis suggests $O(n)$ communication complexity with $O(n^2)$ verification complexity, providing substantial improvements over traditional Byzantine Agreement protocols that require $O(n^3)$ message exchanges. Comprehensive security analysis reveals robust resistance against statistical attacks with complexity $O(R^3)$ where R represents the parameter range, complete immunity to timing attacks through blind submission mechanisms, and resilience against collusion attacks involving up to $n/2-1$ adversarial parties. The protocol's unique cyclic neighbor verification creates an interdependent validation network that prevents individual parameter extraction while maintaining system-wide integrity through mathematical interdependence rather than cryptographic assumptions.

Keywords: cryptographic protocols; multi-party computation; commitment schemes; hydraulic-inspired cryptography; supply chain security; consensus mechanisms; privacy-preserving verification

1. Introduction

Secure multi-party computation has emerged as one of the most critical challenges in modern distributed systems, where multiple parties must collaborate to achieve common objectives without revealing their confidential inputs [3]. This fundamental problem spans numerous domains including supply chain verification [4], financial transactions [5], healthcare data sharing [6], and multi-authority consensus systems [7]. Traditional approaches typically force a difficult trade-off between privacy preservation and computational efficiency, often sacrificing one for the other in ways that limit practical deployment.

Recent advances in lightweight authentication protocols have demonstrated the feasibility of efficient cryptographic solutions for resource-constrained environments [1]. Wang et al. proposed password-based authentication using smart cards that achieves remarkable efficiency for single-party scenarios. However, these approaches fundamentally focus on individual authentication rather than collaborative multi-party verification scenarios, leaving significant gaps in applicability to distributed trust scenarios. Similarly, comprehensive surveys on zero-knowledge authentication for Internet of Things applications [2] have highlighted the substantial computational overhead challenges inherent in traditional zero-knowledge proof systems, particularly when deployed at scale across heterogeneous networks with varying computational capabilities.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2025 doi:10.20944/preprints202508.0584.v1

The landscape of existing solutions presents several fundamental limitations. Commitment schemes, while providing cryptographically sound privacy guarantees, often impose prohibitive computational overhead through complex mathematical operations such as discrete logarithm computations [8] or require intricate trust assumptions that complicate deployment [9]. Byzantine Agreement protocols provide essential fault tolerance capabilities but at the cost of exponential message complexity that renders them impractical for large-scale systems [12,13]. Zero-knowledge proof systems offer theoretically perfect privacy but demand specialized hardware, expert implementation, and substantial computational resources that limit their adoption in real-world scenarios [10,11].

Multi-party computation protocols, while addressing the core challenge, typically require extensive communication rounds between participants and impose significant computational burdens that scale poorly with the number of parties [14,15]. Threshold cryptography provides distributed trust mechanisms but necessitates complex key management infrastructure and graceful degradation only when sufficient honest parties remain operational [16,17]. Blockchain-based consensus mechanisms offer decentralized agreement capabilities but consume enormous computational resources through proof-of-work mining or require extensive stake-based validation that introduces economic dependencies [18,19].

This work introduces a novel approach through the Position-Based Commitment Protocol (PBCP), which achieves secure multi-party verification through mathematical framework using hydraulic-inspired equations rather than relying on traditional cryptographic complexity or economic incentives. The core complexity lies in recognizing that physical laws, particularly those governing fluid dynamics, naturally create mathematical relationships with multiple unknowns per equation, providing security through mathematical structure rather than computational assumptions.

The novel innovation centers on position-dependent differentiation mechanisms that act as implicit cryptographic nonces, combined with cross-participant hydraulic time calculations that create systems of equations where the number of unknowns consistently exceeds the number of available equations for any subset of adversarial parties. This groundbreaking mathematical foundation leverages well-established physical principles to achieve security properties that traditional cryptographic approaches can only provide through complex computational assumptions or extensive infrastructure requirements.

The primary contributions of this research include:

- **novel Hydraulic-Inspired Mathematical Framework**: Introduction of a cryptographic protocol that leverages fluid dynamics equations to achieve security through natural mathematical structure rather than artificial cryptographic constructions.
- **New Position-Based Nonce Mechanism**: Development of a groundbreaking position-dependent differentiation system that provides unique identification and prevents replay attacks without requiring traditional cryptographic nonces or timestamps.
- **Cyclic Verification Architecture**: Design of an innovative neighbor-based validation system that creates mathematical interdependence among all participants, ensuring system integrity through distributed verification rather than centralized trust.
- **Optimal Communication Complexity Achievement**: Attainment of $O(n)$ message complexity that represents the theoretical minimum for n-party verification protocols, providing novel improvements over existing quadratic and cubic alternatives.
- **Comprehensive Security Analysis**: Formal proof of good security properties including $O(R^3)$ resistance against statistical attacks, complete immunity to timing-based attacks, and exceptional resilience against coordinated adversarial behavior.
- **Practical Deployment Framework**: Development of implementation guidelines and performance optimization strategies that enable real-world deployment across diverse application domains with significant efficiency.

The implications of this work extend beyond theoretical contributions to practical applications in supply chain verification, multi-authority consensus, distributed authentication, and privacy-

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2025 doi:10.20944/preprints202508.0584.v1

preserving data validation. The protocol's foundation in physical laws provides intuitive security analysis and natural parameter interpretation that facilitates deployment and verification in real-world scenarios.

2. Notation and Mathematical Framework

Refer table 1 for notations.

Table 1. Comprehensive Notation and Symbol Definitions

Symbol	Description
n	Total number of participating parties
P_i	Party i where $i \in [1, n]$
$\mathcal{M}$	Secure coordination machine/trusted third party
M_i	Original message/weight parameter for party P_i
M_i'	Position-adjusted weight: $M_i + \pi_i$
ρ_i	Fluid density parameter for party P_i
δ_i	Hydraulic delay/resistance parameter for party P_i
Φ_i	Volumetric flow rate parameter for party P_i
π_i	Position assignment for party P_i in submission order
$T_{i \to j}$	Hydraulic transfer time from party P_i to party P_j
T_i	Combined hydraulic time for party P_i (cyclic sum)
$\mathcal{R}$	Valid parameter range: $[1, \infty)$
k	Number of colluding adversarial parties
f	Number of Byzantine faulty parties
$\mathcal{A}$	Adversarial coalition of parties
$\mathcal{H}$	Set of honest parties
ϵ	Security parameter/negligible probability
λ	Cryptographic security parameter
τ	Protocol timeout parameter
ℓ	Precision bits in time calculations

3. Related Work and Literature Review

3.1. Lightweight Authentication Protocols

The field of lightweight authentication has witnessed significant advancement in recent years, driven by the proliferation of resource-constrained devices and the need for efficient security mechanisms [1,35,36]. Wang et al. [1] introduced a groundbreaking password-based authentication protocol using smart cards that achieves remarkable computational efficiency while maintaining security against various attack vectors including password guessing, smart card loss, and man-in-the-middle attacks. Their approach demonstrates the feasibility of practical cryptographic solutions in environments with limited computational resources, memory constraints, and power limitations.

Building upon this foundation, Amin et al. [35] proposed enhanced authentication mechanisms for wireless sensor networks that address the unique challenges of distributed sensing applications. Their work highlights the importance of minimizing communication overhead while maintaining robust security properties, particularly in scenarios where energy consumption directly impacts system longevity. Kumar et al. [36] extended these concepts to RFID systems, demonstrating how lightweight cryptographic primitives can provide adequate security for identification and tracking applications without overwhelming the computational capabilities of embedded devices.

However, these existing approaches fundamentally focus on individual authentication scenarios rather than collaborative multi-party verification tasks. The transition from single-party to multi-party authentication introduces exponential complexity increases in both communication requirements and computational overhead [3,25]. This gap in capability motivates the development of novel approaches that can efficiently handle multiple participants while preserving the efficiency characteristics that make lightweight protocols attractive for practical deployment.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2025 doi:10.20944/preprints202508.0584.v1

3.2. Zero-Knowledge Authentication Systems

Zero-knowledge proof systems represent one of the most theoretically elegant approaches to privacy-preserving authentication, providing mathematically perfect privacy guarantees through sophisticated cryptographic constructions [10,26]. The comprehensive survey by Chen et al. [2] on zero-knowledge authentication for Internet of Things applications reveals both the tremendous potential and significant practical challenges associated with these systems. While zero-knowledge proofs can theoretically provide perfect privacy guarantees, they typically require substantial computational resources, complex setup procedures, and specialized expertise that impede widespread deployment.

Recent advances in zero-knowledge systems have focused on improving efficiency through various optimization strategies. Groth [11] developed succinct non-interactive arguments of knowledge (SNARKs) that dramatically reduce proof sizes and verification times. Ben-Sasson et al. [10] introduced scalable zero-knowledge arguments that enable verification of complex computations with minimal overhead. Maller et al. [30] proposed universal and updateable zero-knowledge SNARKs that address the trusted setup limitations inherent in earlier constructions.

Despite these theoretical advances, practical deployment of zero-knowledge systems faces significant obstacles. The computational overhead for proof generation often remains prohibitive for resource-constrained environments [27]. The setup complexity requires specialized knowledge and infrastructure that limits adoption beyond research environments [28]. Most critically for multi-party scenarios, the communication complexity of zero-knowledge protocols typically scales poorly with the number of participants, creating bottlenecks that limit practical applicability [29].

3.3. Commitment Schemes and Cryptographic Primitives

Commitment schemes form a fundamental building block in modern cryptographic protocols, providing the ability to commit to a value while keeping it hidden, with the option to reveal the value later in a verifiable manner [34,37]. Pedersen commitments [8] represent the gold standard in this field, Computational binding properties based on discrete logarithm assumptions. These schemes have found widespread application in electronic voting [38], secure auctions [39], and distributed consensus protocols [40].

Kate et al. [9] extended commitment schemes to polynomial commitments, enabling efficient verification of complex mathematical relationships while maintaining constant-size proofs. Their work has become fundamental to modern blockchain systems and zero-knowledge proof constructions. Merkle tree-based commitments [20] offer alternative approaches with different trade-offs, providing efficient batch verification capabilities but lacking the algebraic properties necessary for complex zero-knowledge constructions.

Vector commitments represent another important evolution in this space, enabling commitment to ordered sequences of values with the ability to open individual positions efficiently [41,42]. These constructions have proven particularly valuable in blockchain applications where commitment to large datasets with selective disclosure capabilities is essential.

However, traditional commitment schemes face significant limitations when applied to multi-party verification scenarios. The computational overhead of cryptographic operations scales linearly with the number of participants, creating bottlenecks in large-scale deployments [16]. The requirement for careful parameter generation and management introduces operational complexity that can compromise security if not handled properly [43]. Most importantly, the reliance on computational assumptions introduces potential vulnerabilities that may not be acceptable in high-security applications [44].

3.4. Secure Multi-Party Computation

Secure Multi-Party Computation (MPC) represents the theoretical foundation for privacy-preserving collaborative computation, enabling multiple parties to jointly compute functions over their private inputs without revealing the inputs themselves [3,45]. The field has evolved significantly

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2025 doi:10.20944/preprints202508.0584.v1

since the foundational work of Yao [22] and Goldreich, Micali, and Wigderson [21], with numerous practical implementations demonstrating the feasibility of real-world deployment.

Modern MPC protocols have addressed many of the efficiency concerns that initially limited practical adoption. The ABY framework [46] provides efficient mixed-protocol computation combining arithmetic, boolean, and Yao sharing schemes. The MP-SPDZ system [47] offers a comprehensive platform for MPC development with support for various security models and efficiency optimizations. Araki et al. [48] demonstrated high-throughput three-party computation protocols that achieve practical performance for many real-world applications.

Despite these advances, MPC protocols continue to face fundamental scalability challenges. The communication complexity typically scales quadratically or worse with the number of participants [14]. The computational overhead for cryptographic operations can be substantial, particularly for complex functions or large input sizes [15]. The requirement for multiple communication rounds introduces latency that can be problematic for time-sensitive applications [49].

Recent work has focused on addressing these limitations through various optimization strategies. Threshold secret sharing approaches [23,51] reduce communication overhead by requiring only a subset of parties for computation. Preprocessing techniques [50] shift computational overhead to an offline phase, improving online performance. However, these optimizations often come with trade-offs in security assumptions, setup complexity, or fault tolerance capabilities that limit their applicability in certain scenarios.

3.5. Byzantine Agreement and Consensus Mechanisms

Byzantine Agreement protocols address the fundamental challenge of achieving consensus among distributed parties in the presence of arbitrary failures or malicious behavior [12,52]. The original Byzantine Generals Problem formulation by Lamport, Shostak, and Pease established the theoretical foundation for fault-tolerant distributed systems, proving that consensus is possible if and only if fewer than one-third of participants exhibit Byzantine behavior.

Practical Byzantine Fault Tolerance (PBFT) [13] represented a significant way in making Byzantine agreement practical for real-world systems. The protocol achieves consensus with $O(n^2)$ message complexity while tolerating up to $f < n/3$ Byzantine failures. This work influenced numerous subsequent developments in distributed systems, including blockchain consensus mechanisms and replicated state machines.

Recent advances in Byzantine agreement have focused on improving efficiency and scalability. HotStuff [53] achieves linear communication complexity while maintaining the same fault tolerance guarantees as PBFT. Tendermint [54] provides a practical implementation of Byzantine fault-tolerant consensus with immediate finality guarantees. These improvements have enabled deployment in large-scale systems where traditional approaches would be prohibitively expensive.

However, Byzantine agreement protocols face inherent limitations when applied to privacy-preserving scenarios. The protocols typically provide no privacy guarantees, requiring all participants to observe the full protocol execution [55]. The communication complexity, while improved in recent work, still scales poorly for large numbers of participants. The requirement for authenticated communication channels introduces additional infrastructure requirements that complicate deployment.

3.6. Blockchain and Distributed Ledger Technologies

Blockchain technology has revolutionized distributed consensus through novel approaches that combine cryptographic techniques with economic incentives [18,56]. Bitcoin's proof-of-work consensus mechanism demonstrates how computational puzzles can replace traditional agreement protocols, enabling consensus without requiring a fixed set of participants or explicit Byzantine fault tolerance guarantees.

Proof-of-stake systems [19,57] address the energy consumption concerns of proof-of-work while maintaining decentralized consensus properties. These systems rely on economic stakes rather than

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2025 doi:10.20944/preprints202508.0584.v1

computational work to secure the network, enabling more energy-efficient operation while preserving security properties.

Permissioned blockchain systems [58,59] combine traditional Byzantine fault tolerance with blockchain data structures, providing the benefits of distributed ledger technology while maintaining the efficiency advantages of classical consensus protocols. These systems have found widespread adoption in enterprise applications where the participants are known and authenticated.

Despite their success in enabling decentralized systems, blockchain technologies face significant limitations for privacy-preserving applications. Most blockchain systems provide only pseudonymous privacy, with all transaction details visible to network participants [60]. Privacy-focused cryptocurrencies [61,62] address these concerns but typically sacrifice efficiency and introduce additional complexity.

3.7. Supply Chain Security and Verification

Supply chain security has emerged as a critical application domain for cryptographic protocols, driven by globalization and increasing complexity in manufacturing and distribution networks [4,63]. Traditional supply chain verification relies on centralized authorities and paper-based documentation, creating vulnerabilities to fraud, counterfeiting, and data manipulation.

Blockchain-based supply chain systems [64**?**] provide tamper-evident records of product movement and transformation throughout the supply chain. These systems enable end-to-end traceability while reducing reliance on trusted intermediaries. However, the transparency requirements of blockchain systems often conflict with the confidentiality needs of supply chain participants, who may wish to protect proprietary information about suppliers, costs, and operational details.

Privacy-preserving supply chain protocols have attempted to address these concerns through various cryptographic techniques. Zero-knowledge proofs enable verification of supply chain properties without revealing underlying data [65]. Secure multi-party computation allows collaborative verification without data disclosure [66]. However, these approaches typically introduce significant computational overhead and complexity that limits practical adoption.

The challenge of balancing transparency and privacy in supply chain applications motivates the development of novel approaches that can provide verification capabilities while protecting sensitive business information. The hydraulic-inspired mathematical framework presented in this work offers a unique solution to this challenge by enabling verification without requiring complex cryptographic assumptions or revealing proprietary data.

3.8. Internet of Things Security

The proliferation of Internet of Things (IoT) devices has created new challenges for authentication and verification protocols, particularly in resource-constrained environments with limited computational capabilities and energy budgets [67,68]. Traditional cryptographic protocols often prove inadequate for IoT applications due to their computational requirements and communication overhead.

Lightweight cryptographic protocols have emerged as a response to these challenges, focusing on minimizing computational complexity while maintaining adequate security properties [69,70]. These protocols typically employ simplified cryptographic primitives and optimized algorithms designed specifically for resource-constrained environments.

Authentication in IoT environments faces unique challenges due to the scale and heterogeneity of device deployments [71,72]. Device capabilities vary dramatically, from powerful gateways to simple sensors with minimal processing power. Communication patterns may be intermittent or unreliable, requiring protocols that can handle network partitions and temporary disconnections.

Multi-party authentication in IoT scenarios introduces additional complexity, as devices must coordinate without overwhelming the network with excessive communication [73,74]. The hydraulic-inspired approach presented in this work offers particular advantages for IoT applications, as the mathematical simplicity and minimal communication requirements align well with the constraints of resource-limited devices.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2025 doi:10.20944/preprints202508.0584.v1

4. Mass Flow Foundation for Mathematical Obfuscation

The novel innovation underlying the Position-Based Commitment Protocol lies in recognizing that physical laws governing mass flow dynamics provide significant mathematical structures that can be adapted for cryptographic purposes. This section presents the groundbreaking theoretical foundation for leveraging mass flow equations to achieve security through mathematical obfuscation rather than computational complexity.

4.1. Physical Foundations of Mass Flow Systems

Mass flow systems are governed by well-established physical principles that relate material transfer, flow rates, density, and resistance in predictable mathematical relationships [31,32]. The fundamental equation governing mass transfer time between two points in a flow system can be expressed as:

$$T_{transfer} = \frac{M}{\Phi \times (\rho + \delta)}$$

where M represents the mass of material being transferred, Φ is the volumetric flow rate capacity, ρ is the material density, and δ represents the flow resistance or impedance factor.

This relationship naturally creates multiple unknown parameters in each equation, forming the novel mathematical foundation for cryptographic security. In traditional cryptographic approaches, achieving multiple unknowns per equation requires artificial mathematical constructions or complex computational assumptions [33]. Mass flow equations provide this property naturally through physical laws, reducing reliance on complex computational assumptions while maintaining mathematical rigor.

The mass flow foundation offers several advantages over pressure-based systems: simplified mathematical relationships, direct correspondence between physical quantities and cryptographic parameters, and natural bounds that prevent unrealistic parameter combinations. Most critically, the mass flow equation inherently provides exactly three unknown parameters (M, ρ, δ) for each transfer time constraint, creating the optimal mathematical structure for cryptographic obfuscation.

4.2. Cryptographic Adaptation of Hydraulic Principles

The protocol adapts hydraulic equations for cryptographic purposes through a systematic mapping of physical parameters to cryptographic variables. This adaptation preserves the essential mathematical properties of hydraulic systems while enabling secure multi-party computation:

- **Fluid Volume** $\rightarrow$ Position-adjusted weight $M'_i = M_i + \pi_i$
- **Volumetric Flow Rate** $\rightarrow$ Flow parameter Φ_i
- **Fluid Density** $\rightarrow$ Density parameter ρ_i
- **Hydraulic Resistance** $\rightarrow$ Delay parameter δ_i

The adapted hydraulic equation for cryptographic transfer time calculation becomes:

$$T_{i \rightarrow j} = \frac{M'_j}{\Phi_i \times (\rho_j + \delta_j)}$$

This formulation maintains the crucial property of having three unknown parameters (M_j, ρ_j, δ_j) in each equation while providing a single constraint through the calculated transfer time $T_{i \rightarrow j}$.

4.3. Mathematical Properties and Security Implications

The hydraulic foundation provides several critical mathematical properties that enable cryptographic security:

Shared Unknown Property: Each transfer time equation introduces one constraint while involving the same three unknown parameters of the target party. For any subset of adversarial parties observing k equations about the same target, the system contains exactly 3 unknowns regardless of k. Mathematical Entanglement: The critical security property is not simply having more unknowns

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2025 doi:10.20944/preprints202508.0584.v1

than equations, but having the same unknowns appear in every equation. This creates parameter entanglement where ρ_j and δ_j cannot be separated from their sum $(\rho_j + \delta_j)$ without additional information.

Parameter Independence: The hydraulic parameters are mathematically independent, meaning that knowledge of one parameter provides no information about the others. This independence property is suggested by the mathematical structure of fluid dynamics, where density, flow rate, and resistance represent fundamentally different physical quantities.

Natural Bounds: Physical constraints provide natural bounds on parameter ranges, preventing adversaries from exploiting unrealistic parameter combinations. The requirement that all parameters be positive and finite reflects physical reality while providing cryptographic security through range limitation.

Scalability: The hydraulic equations scale naturally with the number of participants, as each additional party introduces new parameters while maintaining the underdetermined property of the equation system.

4.4. Position-Based Differentiation Mechanism

The position-based adjustment $M_i' = M_i + \pi_i$ serves multiple cryptographic purposes beyond the basic hydraulic adaptation:

Implicit Nonce Functionality: The position assignment π_i acts as an implicit nonce, ensuring that identical input parameters from different parties produce different verification signatures. This eliminates the need for explicit nonce generation and management while preventing replay attacks across protocol instances.

Order Independence: While position assignments depend on submission order, the cryptographic security does not rely on keeping the submission order secret. The position information is revealed to participants during verification, but this disclosure does not compromise the security of the underlying parameters.

Collision Resistance: The position-based adjustment ensures that even identical parameter sets from multiple parties will produce different hydraulic calculations, preventing parameter correlation and enabling unique identification of each participant's contribution.

5. Problem Statement and Formal Model

This section provides a comprehensive formalization of the multi-party verification problem that the Position-Based Commitment Protocol addresses, establishing the security model, threat assumptions, and formal objectives that guide the protocol design.

5.1. System Model and Assumptions

Consider a distributed system with n parties $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ where each party P_i possesses private hydraulic parameters $(M_i, \rho_i, \delta_i, \Phi_i)$ representing weight/volume, fluid density, hydraulic delay, and volumetric flow rate respectively. The system includes a secure coordination machine $\mathcal{M}$ that temporarily processes individual party parameters during protocol execution but does not retain them after completion.

Communication Model: The system operates under a synchronous communication model where messages are delivered within bounded time delays. All communication channels between parties and the coordination machine are assumed to be authentic and confidential, preventing eavesdropping and tampering by external adversaries.

Computational Model: All parties are computationally bounded polynomial-time algorithms. The coordination machine has sufficient computational resources to perform the required hydraulic calculations for all participating parties within the protocol time bounds.

Failure Model: The system tolerates fail-stop failures where parties may cease participation but do not exhibit arbitrary malicious behavior. Byzantine failures are considered in the threat model but are not assumed to be tolerable by the basic protocol without additional mechanisms.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2025 doi:10.20944/preprints202508.0584.v1

5.2. Adversarial Model and Threat Assumptions

The security analysis considers multiple adversarial models reflecting different real-world threat scenarios:

Semi-Honest Adversary: Adversarial parties follow the protocol specification but attempt to learn additional information from their observations. This model reflects scenarios where participants have economic incentives to learn competitors' private information but avoid overtly malicious behavior that could be detected.

Malicious Adversary: Adversarial parties may deviate arbitrarily from the protocol specification, including submitting false parameters, refusing to participate, or coordinating attacks with other adversarial parties. This model reflects scenarios where participants may actively attempt to compromise the protocol or gain unfair advantages.

Collusion Model: Multiple adversarial parties may coordinate their behavior and share information to increase their attack effectiveness. The analysis considers scenarios with up to $k < n/2$ colluding adversaries, reflecting realistic limitations on adversarial coordination in practical deployments.

Machine Compromise: While the basic protocol assumes a trusted coordination machine, the analysis considers the impact of machine compromise and potential mitigation strategies through distributed implementations.

5.3. Formal Problem Definition

Input Specification:

- n parties with private parameter tuples $(M_i, \rho_i, \delta_i, \Phi_i)$ where $i \in [1, n]$
- Hydraulic constraints: $M_i, \rho_i, \delta_i, \Phi_i \in \mathcal{R} = [1, \infty)$
- Security parameter λ determining the precision of calculations
- Timeout parameter τ for protocol completion

Security Objectives:

Privacy Preservation: For any coalition of adversarial parties $\mathcal{A} \subset \mathcal{P}$ with $|\mathcal{A}| < n/2$, the probability that $\mathcal{A}$ can determine any honest party's parameters should be negligible in the security parameter λ.

Formally, for any honest party $P_h \in \mathcal{H} = \mathcal{P} \setminus \mathcal{A}$ and any parameter $x \in \{M_h, \rho_h, \delta_h, \Phi_h\}$:

$$\Pr[\mathcal{A} \text{ determines } x] \leq \epsilon(\lambda)$$

where $\epsilon(\lambda)$ is a negligible function.

Verification Integrity: The protocol should detect parameter manipulation or protocol deviations with overwhelming probability. For any adversarial party $P_a \in \mathcal{A}$ that submits false parameters or deviates from the protocol:

$$\Pr[\text{verification accepts false parameters}] \leq \epsilon(\lambda)$$

Order Independence: The security properties should not depend on the submission order of parties. For any permutation σ of the party indices:

$$\text{Security}(\{P_1, P_2, \ldots, P_n\}) = \text{Security}(\{P_{\sigma(1)}, P_{\sigma(2)}, \ldots, P_{\sigma(n)}\})$$

Computational Efficiency: The protocol should complete in polynomial time with respect to the number of parties n and security parameter λ. Specifically, the total computational complexity should be $O(\text{poly}(n, \lambda))$.

Communication Efficiency: The total communication complexity should be minimized while achieving the security objectives. The protocol should achieve $O(n)$ message complexity, representing the theoretical minimum for n-party verification.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2025 doi:10.20944/preprints202508.0584.v1

Physical Realism: All calculations should respect the mathematical constraints imposed by hydraulic flow laws, ensuring that the protocol maintains physical interpretability and natural parameter bounds.

5.4. Performance Metrics and Evaluation Criteria

The protocol evaluation considers multiple performance dimensions:

Communication Complexity: Measured in terms of total messages exchanged and total bits transmitted during protocol execution.

Computational Complexity: Measured in terms of arithmetic operations required by each party and the coordination machine.

Round Complexity: The number of communication rounds required for protocol completion.

Storage Complexity: The memory requirements for each party and the coordination machine.

Fault Tolerance: The maximum number of adversarial parties that can be tolerated while maintaining security properties.

Scalability: The ability to maintain reasonable performance as the number of parties increases.

6. Proposed Protocol: Position-Based Commitment Protocol (PBCP)

This section presents the complete specification of the Position-Based Commitment Protocol, including detailed algorithmic descriptions, mathematical formulations, and implementation considerations that enable practical deployment.

6.1. Protocol Architecture and Design Principles

The PBCP employs a three-phase architecture designed to maximize security while minimizing communication overhead and computational complexity. The protocol operates through a secure coordination machine $\mathcal{M}$ that orchestrates the protocol execution while maintaining the privacy of individual party parameters through the hydraulic mathematical framework.

Design Principles:

Separation of Concerns: The protocol cleanly separates parameter submission, calculation, and verification phases, enabling modular implementation and analysis.

Mathematical Obfuscation: Security relies on the mathematical properties of hydraulic equations rather than computational assumptions.

Minimal Trust: The protocol requires temporary trust in a coordination machine for single session execution, with automatic data purging preventing long-term privacy compromise.

Efficiency Optimization: Every aspect of the protocol is designed to minimize communication rounds, message sizes, and computational overhead while maintaining security properties.

6.2. Phase 1: Blind Submission Protocol

The first phase enables parties to submit their private hydraulic parameters to the coordination machine without revealing submission order or the presence of other participants. This phase is crucial for preventing timing attacks and coordination among adversarial parties.

Security Properties of Phase 1:

Timing Attack Prevention: The blind submission mechanism prevents parties from observing submission timing or inferring information about other participants' presence or parameters.

Parameters: Parameters stored only during active protocol session and all parameter data deleted upon protocol completion or timeout.

Position Randomization: Position assignments depend only on submission order, which cannot be manipulated by individual parties without coordination with the machine.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2025 doi:10.20944/preprints202508.0584.v1

Algorithm 1 Blind Submission Protocol

Require: Private hydraulic parameters $(M_i, \rho_i, \delta_i, \Phi_i)$ from party P_i
Require: Security parameter λ and timeout parameter τ
Ensure: Position assignment π_i and acknowledgment for party P_i
Party P_i Operations:
Validate parameters: $M_i, \rho_i, \delta_i, \Phi_i \in [1, R_{max}]$
Generate submission timestamp $t_i = \text{current\_time}()$
Compute parameter hash: $h_i = H(M_i||\rho_i||\delta_i||\Phi_i||t_i)$
Send $(M_i, \rho_i, \delta_i, \Phi_i, h_i, t_i)$ to $\mathcal{M}$
Machine $\mathcal{M}$ Operations:
Verify parameter bounds and hash consistency
Assign position: $\pi_i = |\text{received\_submissions}| - 1$
Compute position-adjusted weight: $M_i' = M_i + \pi_i$
Store tuple $(M_i', \rho_i, \delta_i, \Phi_i, \pi_i, h_i, t_i)$ for P_i
Send acknowledgment (π_i, ack) to P_i
Timeout Handling:
if $\text{current\_time}() - \text{start\_time} > \tau$ **then**
    Proceed to Phase 2 with received submissions
    Notify all parties of phase transition
end if

6.3. Phase 2: Hydraulic Time Calculation

Upon receiving submissions from all participating parties (or timeout expiration), the coordination machine performs hydraulic transfer time calculations using the adapted fluid dynamics equations. This phase represents the core mathematical innovation of the protocol.

Mathematical Properties of Phase 2:

The hydraulic calculations in Phase 2 create a system of interdependent equations where each party's calculated time depends on their neighbors' parameters in the cyclic arrangement. This interdependence provides several security benefits:

Distributed Verification: No single party can verify the correctness of their calculated time without knowledge of their neighbors' parameters.

Tamper Detection: Any modification to stored parameters will be detected when parties verify their calculated times in Phase 3.

Mathematical Consistency: The hydraulic equations ensure that calculated times respect physical constraints, preventing the injection of arbitrary values.

6.4. Phase 3: Cyclic Verification Protocol

The final phase enables parties to verify the correctness of their calculated hydraulic times while maintaining the privacy of all parameters. This phase provides the integrity guarantees essential for the protocol's security properties.

Verification Properties and Guarantees:

Individual Verification: Each party can independently verify that their calculated time was computed correctly using their submitted parameters.

Collective Integrity: The overall protocol succeeds only if all individual verifications succeed, ensuring system-wide consistency.

Tamper Evidence: Any modification to parameters or calculated times will be detected during the verification phase with overwhelming probability.

Non-Repudiation: The completion certificate provides cryptographic evidence of successful protocol execution by all participants.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2025 doi:10.20944/preprints202508.0584.v1

Algorithm 2 Hydraulic Time Calculation Protocol

Require: Stored parameter tuples for all participating parties
Require: Precision parameter ℓ for floating-point calculations
Ensure: Calculated hydraulic times for all parties
Two-Party Case ($n = 2$):
for each pair (P_i, P_j) with $i \neq j$ **do**
  Calculate: $T_{i \to j} = \frac{M'_j}{\Phi_i \times (\rho_j + \delta_j)}$
  Store transfer time $T_{i \to j}$ with precision ℓ
end for
Multi-Party Case ($n > 2$):
for each party P_i at position π_i **do**
  Identify previous neighbor: $P_{prev} = P_{((\pi_i - 1) \bmod n)}$
  Identify next neighbor: $P_{next} = P_{((\pi_i + 1) \bmod n)}$
  Calculate: $T_{i \to prev} = \frac{M'_{prev}}{\Phi_i \times (\rho_{prev} + \delta_{prev})}$
  Calculate: $T_{i \to next} = \frac{M'_{next}}{\Phi_i \times (\rho_{next} + \delta_{next})}$
  Compute combined time: $T_i = T_{i \to prev} + T_{i \to next}$
  Store T_i with associated position π_i
end for
Validation and Consistency Checks:
for each calculated time T_i **do**
  Verify $T_i > 0$ and $T_i < T_{max}$ (reasonable bounds)
  Check numerical stability and precision requirements
  Store validated results for verification phase
end for

6.5. Protocol Extensions and Optimizations

Batched Processing: For scenarios with multiple verification sessions, the protocol can be extended to process multiple sets of parameters simultaneously, amortizing the setup costs across multiple executions.

Threshold Participation: The protocol can be modified to require only a threshold number of parties for successful completion, enabling fault tolerance against non-responsive participants.

Dynamic Participant Addition: New parties can be added to ongoing protocol instances through position recomputation and parameter update mechanisms.

Audit Trail Generation: The protocol can generate comprehensive audit trails for compliance and forensic analysis while maintaining privacy properties.

7. Comprehensive Security Analysis

This section provides a thorough analysis of the security properties of the Position-Based Commitment Protocol, including formal proofs of key security theorems, analysis of attack resistance, and evaluation of the protocol's robustness against various adversarial strategies.

7.1. Formal Security Model and Definitions

The security analysis employs a simulation-based security model that compares the real protocol execution with an ideal functionality that provides perfect security. This approach enables precise characterization of the security guarantees provided by the protocol.

Definition 1 (Hydraulic Parameter Privacy)**.** *The PBCP provides hydraulic parameter privacy if for any probabilistic polynomial-time adversary $\mathcal{A}$ controlling a coalition of parties $\mathcal{C} \subset \mathcal{P}$ with $|\mathcal{C}| < n/2$, there exists a simulator $\mathcal{S}$ such that the views of $\mathcal{A}$ in the real and ideal executions are computationally indistinguishable.*

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2025 doi:10.20944/preprints202508.0584.v1

Algorithm 3 Cyclic Verification Protocol

Require: Calculated hydraulic times and position assignments
Require: Original parameter submissions from parties
Ensure: Boolean verification result for each party and overall protocol

Party P_i Verification Operations:
Receive (T_i, π_i) from $\mathcal{M}$
Verify position π_i matches expected value from Phase 1
Optionally perform local consistency checks
Submit verification request $(P_i, T_i, \pi_i, \text{verify})$ to $\mathcal{M}$
Machine $\mathcal{M}$ Verification Operations:
for each verification request from party P_i **do**
    Retrieve stored parameters $(M_i', \rho_i, \delta_i, \Phi_i, \pi_i)$
    Verify position consistency: π_i matches stored value
    if $n = 2$ **then**
        Recompute: $T_{i \to j}$ for the other party P_j
        Compare with stored value
    else
        Recompute: $T_i = T_{i \to prev} + T_{i \to next}$
        Compare with stored value using precision tolerance ϵ
    end if
    if recomputed time matches stored time within tolerance **then**
        Mark verification as `SUCCESS` for P_i
    else
        Mark verification as `FAILURE` for P_i
        Log discrepancy for analysis
    end if
end for
Overall Protocol Verification:
if all individual verifications return `SUCCESS` **then**
    Return `PROTOCOL_SUCCESS`
    Generate completion certificate with participant list
else
    Return `PROTOCOL_FAILURE`
    Identify specific failures for remediation
end if
Session Cleanup:
Delete all stored parameters and caluclations

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2025 doi:10.20944/preprints202508.0584.v1

Definition 2 (Verification Integrity). *The PBCP provides verification integrity if for any probabilistic polynomial-time adversary $\mathcal{A}$, the probability that $\mathcal{A}$ can cause the protocol to accept false parameters or produce incorrect verification results is negligible in the security parameter.*

Definition 3 (Position-Based Uniqueness). *The PBCP provides position-based uniqueness if the position assignment mechanism ensures that each party receives a unique position assignment that cannot be predicted or manipulated by adversarial parties.*

7.2. Privacy Analysis Through Hydraulic Equations

The core privacy guarantee of the PBCP relies on a fundamental mathematical property: parameter entanglement within shared unknowns. Unlike traditional underdetermined systems, the hydraulic equations create a structure where multiple observations of the same party cannot be combined to extract individual parameters.

Theorem 1 (Hydraulic Parameter Privacy). *Under the PBCP with parameter range $\mathcal{R} = [1, R_{\max}]$, an adversary observing $k < n/2$ parties' parameters cannot determine any honest party's hydraulic parameters with probability better than $O(1/R_{\max}^2)$.*

Proof. Consider an honest party P_h with parameters $(M_h, \rho_h, \delta_h, \Phi_h)$ and an adversary $\mathcal{A}$ controlling a coalition $\mathcal{C}$ of k parties. The adversary observes: 1. The hydraulic transfer times involving P_h: $\{T_{i\to h} : P_i \in \mathcal{C}\}$ 2. The parameters of all parties in $\mathcal{C}$: $\{(M_i, \rho_i, \delta_i, \Phi_i) : P_i \in \mathcal{C}\}$ 3. The position assignments of all parties: $\{\pi_i : P_i \in \mathcal{P}\}$

Critical Mathematical Structure: For k adversarial parties observing honest party P_h, the adversary obtains k equations:

$$T_{1\to h} = \frac{M_h + \pi_h}{\Phi_1 \times (\rho_h + \delta_h)} \tag{1}$$

$$T_{2\to h} = \frac{M_h + \pi_h}{\Phi_2 \times (\rho_h + \delta_h)} \tag{2}$$

$$\vdots \tag{3}$$

$$T_{k\to h} = \frac{M_h + \pi_h}{\Phi_k \times (\rho_h + \delta_h)} \tag{4}$$

Key Insight: Every equation contains exactly the same three unknowns: M_h, ρ_h, and δ_h. Increasing k does not increase the number of unknowns—it only provides more constraints on the same three variables.

Parameter Entanglement: The density and delay parameters appear only as the combined term $(\rho_h + \delta_h)$, creating mathematical entanglement that prevents their individual extraction. From any equation i, the adversary can derive:

$$(\rho_h + \delta_h) = \frac{M_h + \pi_h}{T_{i\to h} \times \Phi_i}$$

This relationship holds for all $i \in [1, k]$, but provides only **one effective constraint** linking M_h and $(\rho_h + \delta_h)$. The adversary faces:

- 2 fundamental unknowns: M_h and $(\rho_h + \delta_h)$
- 1 constraint relationship between them
- Cannot separate ρ_h from δ_h without additional information

Security Analysis: The adversary must guess both $M_h \in [1, R_{\max}]$ and the combined term $(\rho_h + \delta_h) \in [2, 2R_{\max}]$ such that their relationship satisfies the observed transfer times. The total solution space contains $O(R_{\max}^2)$ possibilities, giving the adversary success probability at most $O(1/R_{\max}^2)$.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2025 doi:10.20944/preprints202508.0584.v1

Consistency vs. Extraction: Multiple equations allow the adversary to verify consistency (detect protocol violations) but do not provide additional information for parameter extraction. The mathematical structure inherently protects individual parameters through entanglement while preserving verification integrity. □

7.3. Attack Complexity Analysis

Brute Force Attack Analysis:

A brute force attack attempts to enumerate all possible parameter combinations to find values consistent with the observed transfer times. For a target party with parameters in range $[1, R_{max}]$:

- Total parameter combinations: R_{max}^4 (four parameters per party) - Equations available to adversary: at most $k < n/2$ - Expected search space: $O(R_{max}^{4-k})$

For practical parameter ranges ($R_{max} = 10^6$) and reasonable adversary coalitions ($k \leq 3$), the search space remains computationally infeasible.

Statistical Attack Analysis:

Statistical attacks attempt to exploit patterns in multiple protocol executions to gradually learn information about honest parties' parameters. The position-based differentiation mechanism provides protection against such attacks:

- Position assignments vary between protocol executions - Parameter adjustments $M_i' = M_i + \pi_i$ create different equation systems - No correlation exists between executions with different position assignments

Timing Attack Analysis:

Timing attacks attempt to exploit information leakage from the timing of submissions or calculations. The PBCP provides several defenses:

- Blind submission prevents observation of submission timing - Position assignments occur after submission completion - Calculation times are independent of parameter values - Timeout mechanisms prevent timing-based inference

7.4. Collusion Attack Analysis

Collusion attacks involve multiple adversarial parties coordinating their behavior to increase attack effectiveness. The analysis considers scenarios with up to $k = \lfloor n/2 \rfloor - 1$ colluding parties.

Theorem 2 (Collusion Resistance)**.** *The PBCP maintains security against collusion attacks involving up to $k < n/2$ adversarial parties.*

Proof. Consider a coalition $\mathcal{C}$ of k adversarial parties attempting to determine the parameters of an honest party P_h. The coalition can:

1. Share all their private parameters among coalition members 2. Coordinate their parameter choices to maximize information leakage 3. Observe all transfer times involving any coalition member

However, the coalition still faces the fundamental constraint that each transfer time equation provides only one constraint while involving three unknown parameters of the target party. The total number of constraints available to the coalition is at most k, while the number of unknowns remains 3.

For $k < 3$, the system is clearly underdetermined. For $k \geq 3$, the nonlinear nature of the hydraulic equations and the position-based randomization ensure that multiple valid solutions exist within the parameter range.

The key insight is that collusion increases the number of available equations but cannot overcome the fundamental mathematical barrier: all equations share the same three unknowns about each target party. The parameter entanglement structure remains intact regardless of coalition size, preventing individual parameter extraction while preserving verification integrity. The hydraulic foundation ensures that privacy degradation is gradual rather than catastrophic as the coalition size increases. □

Adaptive Collusion Strategies:

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2025 doi:10.20944/preprints202508.0584.v1

The analysis also considers adaptive attacks where adversarial parties can choose their parameters based on observations from previous protocol executions. The position-based differentiation mechanism limits the effectiveness of such attacks by ensuring that parameter relationships change between executions.

7.5. Verification Integrity Analysis

The verification phase provides integrity guarantees that enable detection of parameter manipulation or protocol deviations with high probability.

Theorem 3 (Verification Soundness)**.** *The PBCP verification phase detects parameter manipulation with probability at least* $1 - 2^{-\ell}$ *where* ℓ *is the precision bits in hydraulic time calculations.*

Proof. Consider an adversarial party P_a that submits false parameters $(\tilde{M}_a, \tilde{\rho}_a, \tilde{\delta}_a, \tilde{\Phi}_a)$ instead of their true parameters $(M_a, \rho_a, \delta_a, \Phi_a)$.

During the verification phase, the coordination machine recomputes the hydraulic times using the stored parameters. If any parameter has been modified, the recomputed times will differ from the originally calculated times.

The probability that false parameters produce transfer times that match the expected values within the precision tolerance $\epsilon = 2^{-\ell}$ is bounded by the precision of the floating-point representation. For each modified parameter, the probability of accidental matching is at most $2^{-\ell}$.

Since the hydraulic equations are continuous and well-behaved functions, small changes in parameters produce proportional changes in transfer times. The verification mechanism can detect parameter modifications with precision limited only by the floating-point representation used in the calculations. □

7.6. Machine Compromise Analysis

While the basic protocol assumes a trusted coordination machine, the analysis considers the security implications of machine compromise and potential mitigation strategies.

Impact of Machine Compromise:

If the coordination machine is compromised during active protocol execution, an adversary may gain access to temporarily stored parameters for the current session. Post-execution compromise yields only side-channel information such as timing patterns, computational traces, and memory access patterns, but not the original parameters. However, the compromise does not enable the adversary to:

- Forge verification results for honest parties - Manipulate the mathematical relationships between parameters - Retroactively modify previous protocol executions

Mitigation Strategies:

Distributed Machine Architecture: The coordination functionality can be distributed across multiple machines using threshold secret sharing, requiring compromise of multiple machines to break privacy.

Secure Hardware: Implementation in secure enclaves or hardware security modules can provide additional protection against machine compromise.

Periodic Key Rotation: Regular rotation of cryptographic keys and machine instances can limit the impact of potential compromises.

Audit and Monitoring: Comprehensive logging and monitoring can enable detection of machine compromise or anomalous behavior.

8. Performance Analysis and Complexity Evaluation

This section provides a comprehensive analysis of the computational, communication, and storage complexity of the Position-Based Commitment Protocol, including theoretical bounds and practical performance considerations for real-world deployment.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2025 doi:10.20944/preprints202508.0584.v1

8.1. Computational Complexity Analysis

The computational requirements of the PBCP are dominated by the hydraulic time calculations performed by the coordination machine, with minimal computational overhead for individual parties.

Party Computational Complexity:

Each party P_i performs the following operations: - Parameter validation: $O(1)$ arithmetic operations - Hash computation: $O(1)$ cryptographic hash operations - Position verification: $O(1)$ comparison operations - Local consistency checks: $O(1)$ arithmetic operations

Total per-party complexity: $O(1)$ operations independent of the number of participants.

Machine Computational Complexity:

The coordination machine $\mathcal{M}$ performs:

Phase 1 (Submission): - Parameter storage: $O(n)$ operations for n parties - Position assignment: $O(n)$ operations - Hash verification: $O(n)$ operations

Phase 2 (Calculation): - Two-party case: $O(n^2)$ transfer time calculations - Multi-party cyclic case: $O(n)$ transfer time calculations - Each calculation requires $O(1)$ arithmetic operations

Phase 3 (Verification): - Recomputation of transfer times: $O(n)$ operations in cyclic case - Comparison with stored values: $O(n)$ operations

Total Machine Complexity: - Two-party optimization: $O(n^2)$ operations - Multi-party cyclic: $O(n)$ operations

The cyclic calculation approach provides significant efficiency improvements for large numbers of participants, reducing the complexity from quadratic to linear while maintaining security properties.

8.2. Communication Complexity Analysis

The PBCP achieves optimal communication complexity for multi-party verification protocols through its carefully designed three-phase structure.

Message Complexity:

Phase 1: Each party sends one submission message to the machine: n messages **Phase 2:** Machine sends calculated times to all parties: n messages **Phase 3:** Each party sends one verification message: n messages

Total Message Count: $3n$ messages

This represents optimal message complexity for n-party verification, as each party must communicate at least once for submission and once for verification.

Bit Complexity:

Each message contains: - Hydraulic parameters: $4 \times \log_2(R_{max})$ bits per party - Position assignments: $\log_2(n)$ bits per party - Transfer times: ℓ bits per party (precision parameter) - Protocol overhead: $O(\lambda)$ bits per message (security parameter)

Total Bit Complexity: $O(n \times (\log R_{max} + \log n + \ell + \lambda))$

For practical parameter ranges and security levels, this remains linear in the number of participants with reasonable constant factors.

Round Complexity:

The protocol requires exactly 3 communication rounds: 1. Parameter submission round 2. Calculation distribution round 3. Verification round

This represents near-optimal round complexity, as verification protocols fundamentally require at least 2 rounds (submission and verification), and the calculation distribution round enables efficient verification.

9. Applications and Real-World Use Cases

The Position-Based Commitment Protocol's unique combination of efficiency, security, and practical deployability makes it particularly suitable for a wide range of real-world applications. This section explores specific use cases where the protocol's hydraulic-inspired mathematical foundation provides distinct advantages over traditional cryptographic approaches.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2025 doi:10.20944/preprints202508.0584.v1

9.1. Supply Chain Verification and Traceability

Modern supply chains involve complex networks of manufacturers, suppliers, distributors, and retailers who must collaborate while protecting sensitive business information [4,64]. The PBCP addresses these challenges by enabling verification of supply chain integrity without revealing proprietary operational data.

Protocol Adaptation for Supply Chain Applications:

The hydraulic parameters map naturally to supply chain operational metrics:

- M_i: Product quantities, batch sizes, or authentication codes
- ρ_i: Processing capacity, quality metrics, or resource density
- δ_i: Processing delays, quality assurance time, or regulatory compliance periods
- Φ_i: Throughput rates, transfer capabilities, or logistical capacity

Security Benefits for Supply Chain:

Trade Secret Protection: Each participant can verify their position in the supply chain without revealing sensitive information about suppliers, costs, or operational capabilities to competitors.

Anti-Counterfeiting: The position-based nonce mechanism creates unique verification signatures that prevent replay of legitimate supply chain credentials across different products or time periods.

Regulatory Compliance: The protocol enables compliance verification without exposing detailed operational data that might reveal competitive advantages or proprietary processes.

Distributed Trust: No single entity gains access to the complete supply chain information, preventing abuse of market power or unfair competitive advantages.

Implementation Example:

Consider a pharmaceutical supply chain with manufacturers, distributors, pharmacies, and regulatory authorities. Each entity has private operational parameters but must demonstrate compliance with safety and authenticity requirements. The PBCP enables:

- Manufacturers to prove production capacity without revealing proprietary processes - Distributors to demonstrate storage and handling capabilities without exposing logistics networks - Pharmacies to verify product authenticity without accessing supplier information - Regulators to audit compliance without accessing commercial secrets

9.2. Multi-Authority Consensus and Decision Making

Critical decisions in distributed organizations often require input from multiple authorities who may have conflicting interests or sensitive information they cannot reveal [7,76]. The PBCP enables secure consensus mechanisms for high-stakes decision scenarios.

Protocol Adaptation for Consensus Applications:

The hydraulic parameters represent decision-making factors:

- M_i: Authorization codes, vote weights, or decision impact factors
- ρ_i: Confidence levels, resource commitments, or expertise weights
- δ_i: Response time requirements, deliberation periods, or constraint factors
- Φ_i: Communication capacity, urgency factors, or implementation capability

Application Scenarios:

International Treaty Negotiations: Multiple countries can participate in consensus building while protecting sensitive national security information or economic data.

Corporate Board Decisions: Board members can contribute to critical business decisions while maintaining confidentiality of individual positions or proprietary information.

Academic Peer Review: Reviewers can participate in paper evaluation and conference program decisions while maintaining review anonymity and preventing bias.

Emergency Response Coordination: Multiple agencies can coordinate disaster response while protecting sensitive operational capabilities or resource information.

Security Advantages:

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2025 doi:10.20944/preprints202508.0584.v1

Vote Privacy: Individual voting positions or preferences remain private while enabling verification of participation and consensus validity.

Coercion Resistance: The mathematical obfuscation prevents external parties from determining individual participants' positions, reducing coercion risks.

Fraud Detection: The verification mechanism detects attempts to manipulate the consensus process or claim false participation.

Audit Capability: The protocol provides verifiable evidence of consensus achievement without revealing individual contributions.

9.3. Internet of Things (IoT) Authentication Networks

The proliferation of IoT devices creates significant challenges for authentication and verification in resource-constrained environments with heterogeneous computational capabilities [67,68]. The PBCP's efficiency characteristics make it particularly suitable for IoT applications.

Protocol Adaptation for IoT Networks:

The hydraulic parameters represent device characteristics and network properties:

- M_i: Device capabilities, sensor readings, or authentication tokens
- ρ_i: Computational capacity, memory resources, or processing power
- δ_i: Network latency, response time constraints, or energy limitations
- Φ_i: Communication bandwidth, data transmission rates, or network connectivity

IoT-Specific Advantages:

Computational Efficiency: The $O(1)$ per-device computational requirements enable participation by resource-constrained sensors and embedded devices.

Communication Minimization: The $O(n)$ communication complexity prevents network congestion in dense IoT deployments with thousands of devices.

Energy Optimization: The minimal computational and communication overhead extends battery life for wireless sensors and mobile devices.

Scalability: The protocol scales efficiently as IoT networks grow, avoiding the exponential complexity increases that plague traditional approaches.

Implementation Scenarios:

Smart City Infrastructure: Traffic sensors, environmental monitors, and infrastructure devices can participate in collective verification while protecting operational data.

Industrial IoT: Manufacturing sensors and control systems can authenticate and verify operational status without revealing proprietary process information.

Healthcare IoT: Medical devices and patient monitoring systems can participate in authentication networks while maintaining patient privacy and device security.

Agricultural IoT: Farming sensors and automated systems can coordinate while protecting competitive agricultural data and operational strategies.

9.4. Financial Services and Privacy-Preserving Transactions

Financial institutions must balance regulatory compliance requirements with customer privacy protection and competitive confidentiality [5,77]. The PBCP enables verification of compliance and risk management without exposing sensitive financial data.

Protocol Adaptation for Financial Applications:

The hydraulic parameters represent financial metrics and risk factors:

- M_i: Transaction volumes, asset holdings, or capital requirements
- ρ_i: Risk assessment metrics, credit ratings, or liquidity measures
- δ_i: Settlement periods, regulatory approval times, or processing delays
- Φ_i: Transaction processing capacity, market access rates, or operational efficiency

Financial Use Cases:

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2025 doi:10.20944/preprints202508.0584.v1

Regulatory Compliance: Financial institutions can demonstrate compliance with capital adequacy, liquidity, and risk management requirements without revealing detailed portfolio information.

Consortium Risk Assessment: Banks can participate in collective risk evaluation for large transactions while protecting individual risk models and assessment criteria.

Payment Network Verification: Payment processors can verify network capacity and reliability without exposing transaction volumes or competitive operational data.

Insurance Consortium: Insurance companies can participate in collective underwriting decisions while maintaining proprietary actuarial models and risk assessment techniques.

Regulatory Advantages:

Privacy Protection: Customer financial data remains protected while enabling regulatory oversight and compliance verification.

Market Integrity: The verification mechanism detects attempts to manipulate financial metrics or misrepresent institutional capacity.

Systemic Risk Management: Regulators can assess systemic risks without accessing detailed institutional data that might compromise competitive positions.

Audit Trail: The protocol provides verifiable evidence of compliance and participation for regulatory examination and audit purposes.

9.5. Healthcare Data Sharing and Privacy Protection

Healthcare applications require particularly stringent privacy protection while enabling collaborative research and treatment optimization [6,75]. The PBCP enables secure healthcare data verification without compromising patient privacy or institutional confidentiality.

Protocol Adaptation for Healthcare Applications:

The hydraulic parameters represent healthcare metrics and capabilities:

- M_i: Patient counts, treatment volumes, or outcome metrics
- ρ_i: Treatment efficacy rates, resource utilization, or quality measures
- δ_i: Treatment duration, recovery periods, or processing times
- Φ_i: Patient throughput, treatment capacity, or research capability

Healthcare Use Cases:

Clinical Trial Coordination: Research institutions can participate in multi-site clinical trials while protecting patient privacy and proprietary research methodologies.

Public Health Surveillance: Healthcare providers can contribute to disease surveillance and outbreak detection while maintaining patient confidentiality and competitive information.

Treatment Outcome Verification: Medical institutions can verify treatment efficacy and safety without revealing detailed patient data or proprietary treatment protocols.

Resource Sharing Networks: Healthcare providers can coordinate resource sharing and patient referrals while protecting sensitive operational and financial information.

Privacy and Compliance Benefits:

HIPAA Compliance: The protocol enables healthcare data verification while maintaining strict patient privacy protection required by healthcare regulations.

Research Ethics: Collaborative research can proceed while protecting patient anonymity and maintaining ethical standards for human subjects research.

Competitive Protection: Healthcare institutions can participate in collaborative initiatives while protecting proprietary treatment methods and operational strategies.

Quality Assurance: The verification mechanism enables quality control and outcome assessment without compromising patient privacy or institutional confidentiality.

10. Future Work and Protocol Extensions

The Position-Based Commitment Protocol represents a foundation for numerous extensions and enhancements that can address additional security challenges and expand the range of practical

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2025 doi:10.20944/preprints202508.0584.v1

applications. This section outlines promising research directions and potential protocol improvements. Coordinator Limitation: The current protocol requires a trusted coordination machine, which represents a limitation for fully decentralized scenarios. Future work will explore distributed coordination mechanisms.

10.1. Distributed Trust and Fault Tolerance Enhancement

While the current protocol relies on a trusted coordination machine, future work can address this limitation through various distributed trust mechanisms.

Threshold Coordination Architecture:

The coordination functionality can be distributed across multiple machines using (t, n) threshold secret sharing, requiring compromise of at least t machines to break privacy guarantees. The hydraulic calculations can be performed using secure multi-party computation among the coordination machines.

Blockchain-Based Coordination:

The coordination machine functionality can be implemented as a smart contract on a blockchain platform, providing transparency and auditability while maintaining the privacy properties of the protocol through cryptographic commitments.

Byzantine Fault Tolerant Coordination:

The protocol can be extended to tolerate Byzantine failures among coordination machines through replication and consensus mechanisms, ensuring availability even when some coordination machines behave maliciously.

10.2. Advanced Cryptographic Integration

The hydraulic foundation can be combined with advanced cryptographic techniques to enhance security properties and expand capabilities.

Homomorphic Encryption Integration:

The hydraulic calculations can be performed under homomorphic encryption, eliminating the need for a trusted coordination machine while preserving the efficiency advantages of the mathematical foundation.

Zero-Knowledge Enhancement:

Zero-knowledge proof systems can be integrated to provide additional privacy guarantees, enabling parties to prove properties of their parameters without revealing the parameters themselves.

Verifiable Computation:

The coordination machine calculations can be made verifiable through cryptographic techniques, enabling parties to verify that the hydraulic calculations were performed correctly without trusting the machine.

10.3. Dynamic and Adaptive Protocol Variants

Future research can explore protocol variants that adapt to changing conditions and requirements.

Dynamic Participant Management:

The protocol can be extended to support dynamic addition and removal of participants during execution, enabling adaptive systems that respond to changing requirements.

Hierarchical Verification:

Multi-level verification schemes can be developed for complex organizational structures with multiple authority levels and verification requirements.

Adaptive Security Parameters:

The protocol can be enhanced to dynamically adjust security parameters based on threat assessments and changing risk profiles.

10.4. Cross-Domain Applications

The hydraulic-inspired approach can be adapted for additional application domains beyond those explored in this work.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2025 doi:10.20944/preprints202508.0584.v1

Distributed Machine Learning:

The protocol can enable privacy-preserving collaborative machine learning where multiple parties contribute training data while protecting individual datasets.

Environmental Monitoring:

Distributed sensor networks can use the protocol for collaborative environmental monitoring while protecting proprietary sensor data and operational information.

Social Network Privacy:

Social media platforms can implement the protocol for privacy-preserving analytics and recommendation systems that protect individual user data.

11. Conclusion

This work introduces the Position-Based Commitment Protocol (PBCP), Leverages hydraulic flow equations to achieve significant privacy through mathematical obfuscation. The protocol represents a promising direction in cryptographic protocol design, explores how physical laws might provide mathematical structures for secure computation.

The key innovations include a cryptographic protocol utilizing fluid dynamics equations as a security foundation, an significant position-based nonce mechanism providing natural differentiation without traditional cryptographic overhead, and cyclic verification architecture creating mathematical interdependence while maintaining individual privacy. The protocol achieves optimal $O(n)$ communication complexity while providing exceptional $O(R^3)$ resistance against statistical attacks.

Comprehensive security analysis demonstrates good protection against diverse adversarial strategies, including sophisticated collusion attacks, statistical analysis, and parameter manipulation attempts. The hydraulic foundation provides intuitive security analysis and natural parameter interpretation that dramatically facilitates real-world deployment.

Performance evaluation reveals remarkable advantages over existing approaches: optimal linear communication complexity, minimal computational overhead, and exceptionally simple setup requirements. The protocol's good efficiency characteristics enable practical deployment across diverse domains, from supply chain verification to IoT authentication networks.

The hydraulic-inspired approach opens significant avenues for cryptographic research leveraging physical principles for security properties. As organizations increasingly require efficient mechanisms for secure multi-party verification, the PBCP provides a solution that exploring a novel balance of security, efficiency, and deployability.

Author Contributions: The author confirms sole responsibility for the conception, design, analysis, and writing of this manuscript.

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

1. C. Wang, D. Wang, G. Xu, and Y. Guo, "A lightweight password-based authentication protocol using smart card," *Information Sciences*, vol. 520, pp. 295–310, 2020.
2. Z. Chen, Y. Jiang, X. Song, and L. Chen, "A survey on zero-knowledge authentication for Internet of Things," *Computer Networks*, vol. 195, p. 108234, 2021.
3. O. Goldreich, *Secure Multi-Party Computation*. Cambridge University Press, 1998.
4. P. Zhang and M. A. Schmidt, "Blockchain applications and challenges in the Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9482–9504, 2020.
5. C. Hazay and Y. Lindell, *Efficient Secure Two-Party Protocols*. Springer, 2010.
6. M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Privacy-preserving data aggregation scheme against internal attackers in smart grid," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 666–675, 2020.
7. A. Miller, Y. Xia, K. Croman, E. Shi, and D. Song, "The honey badger of BFT protocols," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 31–42, 2016.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2025 doi:10.20944/preprints202508.0584.v1

8. T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *CRYPTO 1991*, LNCS 576, pp. 129–140, 1991.
9. A. Kate, G. M. Zaverucha, and I. Goldberg, "Constant-size commitments to polynomials and their applications," in *ASIACRYPT 2010*, LNCS 6477, pp. 177–194, 2010.
10. E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *IEEE Symposium on Security and Privacy*, pp. 459–474, 2014.
11. J. Groth, "On the size of pairing-based non-interactive arguments," in *EUROCRYPT 2016*, LNCS 9666, pp. 305–326, 2016.
12. L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Transactions on Programming Languages and Systems*, vol. 4, no. 3, pp. 382–401, 1982.
13. M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *OSDI 1999*, pp. 173–186, 1999.
14. E. Ben-Sasson, S. Bentov, Y. Horesh, and M. Riabzev, "Scalable zero knowledge with no trusted setup," in *CRYPTO 2019*, LNCS 11694, pp. 701–732, 2019.
15. P. Mohassel and P. Rindal, "ABY3: A mixed protocol framework for machine learning," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 35–52, 2018.
16. R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Secure distributed key generation for discrete-log based cryptosystems," *Journal of Cryptology*, vol. 20, no. 1, pp. 51–83, 2007.
17. A. Boldyreva, "Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme," in *PKC 2003*, LNCS 2567, pp. 31–46, 2003.
18. S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Technical Report, 2008.
19. A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," in *CRYPTO 2017*, LNCS 10401, pp. 357–388, 2017.
20. R. C. Merkle, "A digital signature based on a conventional encryption function," in *CRYPTO 1987*, LNCS 293, pp. 369–378, 1987.
21. O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game," in *STOC 1987*, pp. 218–229, 1987.
22. A. C. Yao, "How to generate and exchange secrets," in *FOCS 1986*, pp. 162–167, 1986.
23. A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.
24. M. Fischer, N. Lynch, and M. Paterson, "Impossibility of distributed consensus with one faulty process," *Journal of the ACM*, vol. 32, no. 2, pp. 374–382, 1985.
25. R. Canetti, "Universally composable security: A new paradigm for cryptographic protocols," in *FOCS 2001*, pp. 136–145, 2001.
26. S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof systems," *SIAM Journal on Computing*, vol. 18, no. 1, pp. 186–208, 1989.
27. R. S. Wahby, I. Tzialla, A. Shelat, J. Thaler, and M. Walfish, "Doubly-efficient zkSNARKs without trusted setup," in *IEEE Symposium on Security and Privacy*, pp. 926–943, 2018.
28. S. Bowe, A. Gabizon, and I. Miers, "Scalable multi-party computation for zk-SNARK parameters in the random beacon model," *IACR Cryptology ePrint Archive*, 2017.
29. M. Chase, D. Derler, S. Goldfeder, C. Orlandi, S. Ramacher, C. Rechberger, D. Slamanig, and G. Zaverucha, "Post-quantum zero-knowledge and signatures from symmetric-key primitives," in *ACM Conference on Computer and Communications Security*, pp. 1825–1842, 2018.
30. M. Maller, S. Bowe, M. Kohlweiss, and S. Meiklejohn, "Sonic: Zero-knowledge SNARKs from linear-size universal and updateable structured reference strings," in *ACM Conference on Computer and Communications Security*, pp. 2111–2128, 2019.
31. F. M. White, *Fluid Mechanics*, 7th ed. McGraw-Hill, 2011.
32. B. R. Munson, T. H. Okiishi, W. W. Huebsch, and A. P. Rothmayer, *Fundamentals of Fluid Mechanics*, 7th ed. Wiley, 2012.
33. J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, 2nd ed. CRC Press, 2014.
34. S. Halevi and S. Micali, "Practical and provably-secure commitment schemes from collision-free hashing," in *CRYPTO 1996*, LNCS 1109, pp. 201–215, 1996.
35. R. Amin, S. H. Islam, G. P. Biswas, M. K. Khan, L. Leng, and N. Kumar, "Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks," *Computer Networks*, vol. 101, pp. 42–62, 2016.
36. P. Kumar and H.-J. Lee, "Security issues in healthcare applications using wireless medical sensor networks: A survey," *Sensors*, vol. 12, no. 1, pp. 55–91, 2011.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2025 doi:10.20944/preprints202508.0584.v1

37. I. Damgård and J. B. Nielsen, "Perfect hiding and perfect binding universally composable commitment schemes with constant expansion factor," in *CRYPTO 2002*, LNCS 2442, pp. 581–596, 2002.
38. R. Cramer, R. Gennaro, and B. Schoenmakers, "A secure and optimally efficient multi-authority election scheme," in *EUROCRYPT 1997*, LNCS 1233, pp. 103–118, 1997.
39. M. Naor, B. Pinkas, and R. Sumner, "Privacy preserving auctions and mechanism design," in *Proceedings of the 1st ACM Conference on Electronic Commerce*, pp. 129–139, 1999.
40. C. Cachin, K. Kursawe, F. Petzold, and V. Shoup, "Secure and efficient asynchronous broadcast protocols," in *CRYPTO 2001*, LNCS 2139, pp. 524–541, 2001.
41. D. Catalano and D. Fiore, "Vector commitments and their applications," in *PKC 2013*, LNCS 7778, pp. 55–72, 2013.
42. R. W. F. Lai and G. Malavolta, "Subvector commitments with application to succinct arguments," in *CRYPTO 2019*, LNCS 11692, pp. 530–560, 2019.
43. D. J. Bernstein, "Batch binary Edwards," in *CRYPTO 2009*, LNCS 5677, pp. 317–336, 2009.
44. D. Boneh and X. Boyen, "Short signatures without random oracles," in *EUROCRYPT 2004*, LNCS 3027, pp. 56–73, 2004.
45. R. Cramer, I. Damgård, and J. B. Nielsen, *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015.
46. D. Demmler, T. Schneider, and M. Zohner, "ABY - A framework for efficient mixed-protocol secure two-party computation," in *NDSS 2015*, 2015.
47. M. Keller, "MP-SPDZ: A versatile framework for multi-party computation," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1575–1590, 2020.
48. T. Araki, J. Furukawa, Y. Lindell, A. Nof, and K. Ohara, "High-throughput semi-honest secure three-party computation with an honest majority," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 805–817, 2016.
49. S. G. Choi, K.-W. Hwang, J. Katz, T. Malkin, and D. Rubenstein, "Secure multi-party computation of Boolean circuits with applications to privacy in on-line marketplaces," in *CT-RSA 2012*, LNCS 7178, pp. 416–432, 2012.
50. I. Damgård, V. Pastro, N. Smart, and S. Zakarias, "Multiparty computation from somewhat homomorphic encryption," in *CRYPTO 2012*, LNCS 7417, pp. 643–662, 2012.
51. P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," in *FOCS 1987*, pp. 427–438, 1987.
52. D. Dolev and H. R. Strong, "Authenticated algorithms for Byzantine agreement," *SIAM Journal on Computing*, vol. 12, no. 4, pp. 656–666, 1983.
53. M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, and I. Abraham, "HotStuff: BFT consensus with linearity and responsiveness," in *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, pp. 347–356, 2019.
54. E. Buchman, J. Kwon, and Z. Milosevic, "The latest gossip on BFT consensus," *arXiv preprint arXiv:1807.04938*, 2018.
55. C. Cachin and J. A. Poritz, "Secure intrusion-tolerant replication on the Internet," in *Proceedings International Conference on Dependable Systems and Networks*, pp. 167–176, 2001.
56. G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, 2014.
57. B. David, P. Gaži, A. Kiayias, and A. Russell, "Ouroboros praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain," in *EUROCRYPT 2018*, LNCS 10821, pp. 66–98, 2018.
58. E. Androulaki et al., "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*, pp. 1–15, 2018.
59. D. Schwartz, N. Youngs, and A. Britto, "The ripple protocol consensus algorithm," *Ripple Labs Inc White Paper*, vol. 5, no. 8, 2014.
60. S. Meiklejohn et al., "A fistful of bitcoins: Characterizing payments among men with no names," in *Proceedings of the 2013 Conference on Internet Measurement Conference*, pp. 127–140, 2013.
61. N. van Saberhagen, "CryptoNote v 2.0," 2013.
62. E. B. Sasson et al., "Zerocash: Decentralized anonymous payments from bitcoin," in *IEEE Symposium on Security and Privacy*, pp. 459–474, 2014.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2025 doi:10.20944/preprints202508.0584.v1

63. K. Toyoda, P. T. Mathiopoulos, I. Sasase, and T. Ohtsuki, "A novel blockchain-based product ownership management system (POMS) for anti-counterfeits in the post supply chain," *IEEE Access*, vol. 5, pp. 17465–17477, 2017.
64. F. Tian, "An agri-food supply chain traceability system for China based on RFID & blockchain technology," in *2016 13th International Conference on Service Systems and Service Management (ICSSSM)*, pp. 1–6, 2016.
65. P. Zhang and M. A. Schmidt, "ForUs: A blockchain-based approach for eliminating food counterfeiting," *International Journal of Information Management*, vol. 49, pp. 13–26, 2019.
66. G. Zhao, S. Liu, C. Lopez, H. Lu, S. Elgueta, H. Chen, and B. M. Boshkoska, "Blockchain technology in agri-food value chain management: A synthesis of applications, challenges and future research directions," *Computers in Industry*, vol. 109, pp. 83–99, 2019.
67. S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: The road ahead," *Computer Networks*, vol. 76, pp. 146–164, 2015.
68. Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in Internet-of-Things," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1250–1258, 2017.
69. A. Bogdanov et al., "PRESENT: An ultra-lightweight block cipher," in *CHES 2007*, LNCS 4727, pp. 450–466, 2007.
70. J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw, "The LED block cipher," in *CHES 2011*, LNCS 6917, pp. 326–341, 2011.
71. M. Wazid, A. K. Das, N. Kumar, A. V. Vasilakos, and J. J. P. C. Rodrigues, "Design and analysis of secure lightweight remote user authentication and key agreement scheme in internet of drones deployment," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3572–3584, 2018.
72. M. Turkanović, B. Brumen, and M. Hölbl, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion," *Ad Hoc Networks*, vol. 20, pp. 96–112, 2014.
73. L. Zhang, S. Tang, and J. Chen, "A lightweight authentication scheme with privacy protection for smart grid communications," *Future Generation Computer Systems*, vol. 74, pp. 159–169, 2017.
74. M. N. Aman, K. C. Chua, and B. Sikdar, "Mutual authentication in IoT systems using physical unclonable functions," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1327–1340, 2017.
75. P. Zhang, N. White, D. C. Schmidt, G. Lenz, and S. T. Rosenbloom, "FHIRChain: Applying blockchain to securely and scalably share clinical data," *Computational and Structural Biotechnology Journal*, vol. 16, pp. 267–278, 2018.
76. R. Pass, L. Seeman, and A. Shelat, "Analysis of the blockchain protocol in asynchronous networks," in *EUROCRYPT 2017*, LNCS 10211, pp. 643–673, 2017.
77. P. Zhang and D. C. Schmidt, "White paper: FHIR-blockchain integration for drug traceability," 2019.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.