Symbolic Formalization of PoW Integrity in Blockchain*

Kengo Matsui[1] and Shigeki Hagihara[1,2]†

[1] Chitose Institute of Scince and Technology, Chitose, Hokkaido, Japan
{m2240430,s-hagiha}@photon.chitose.ac.jp
[2] Chitose Silicon Research Center, Chitose, Hokkaido, Japan

Abstract

This study proposes a novel framework that formally and abstractly describes and verifies the integrity of Proof of Work (PoW) in blockchain using the strand space model. Conventional evaluations of PoW security have relied on quantitative methods based on probability theory. These methods are characterized by complex and detailed models and are constrained by specific assumptions and numerical conditions. However, to demonstrate that PoW satisfies integrity, it should be sufficient to provide a qualitative representation without relying on concrete values. Therefore, this study focuses on the essential operations in PoW, namely transaction generation, puzzle generation, computational effort, and verification, and qualitatively captures the mechanism by which the protocol satisfies integrity by representing these elements as symbolically structured sequences that are connected causally. In particular, by introducing computational effort as an extended message and expressing its sufficiency using the asterisk symbol, this model formalizes how the behavior of miners influences integrity. The model demonstrates that when honest miners constitute the majority, integrity is ensured, whereas if dishonest miners dominate, tampering may become possible. This study provides a structural understanding of PoW security and lays the foundation for future mechanical verification and automated analytical methods.

1 Introduction

Blockchain technology has attracted attention as a means to achieve highly reliable transaction records over decentralized networks, and its application has progressed in various fields such as finance and public administration [2]. Among its important properties is integrity, meaning that recorded data is difficult to tamper with.

A representative implementation of blockchain is Bitcoin, proposed by Nakamoto in 2008 [9]. In Bitcoin, consensus on the transaction ledger is achieved in a decentralized manner without relying on a trusted third party. This is made possible by a mechanism called Proof of Work (PoW), which ensures that only valid blocks are added to the blockchain through computational effort. Specifically, each node attempts to solve a computationally costly puzzle in order to add a new block, and the block is added once a correct answer is found. At this point, even if dishonest nodes attempt to add a different block, it will fail to do so as long as its computational power is inferior to that of the honest nodes, ensuring that only valid blocks are appended to the blockchain.

The integrity of the blockchain is thus mainly realized through PoW, and its security has conventionally been demonstrated using quantitative models based on probability theory [3]. However, such quantitative methods depend on specific numerical values and assumptions, and are insufficient as a framework to describe the principle of integrity in a more structural and

*Proceedings of the 9th International Conference on Mobile Internet Security (MobiSec'25), Article No. 57, December 16-18, 2025, Sapporo , Japan.

†Corresponding author

universal manner. By reconsidering the nature of PoW in an abstract and qualitative way, it may be possible to express the essence of its security more concisely.

This study aims to construct an abstract model suitable for formal description in order to clarify how PoW ensures integrity. Specifically, it introduces a framework that qualitatively describes the behavior of PoW by applying the strand space model [6], which is used in the analysis of security protocols. As an initial attempt, this study introduces a new term representing the effort of nodes, and by modeling it from the abstract perspective of whether it is large in quantity or not, shows that the structure of PoW can be represented as strands. Although this research has not yet reached a formal proof of integrity, it presents a direction toward the construction of a theoretical foundation for formally capturing the security of PoW.

2 PoW and Integrity in Blockchain

In this paper, we do not address the entire blockchain system but rather focus on the protocol of Proof of Work (PoW), one of the representative consensus algorithms.

2.1 Overview of the PoW Protocol

In a typical blockchain employing PoW, there are mainly two types of participants: transaction creators and miners. The protocol generally proceeds through the following four steps:

1. **Transaction generation and dissemination**
   A transaction creator generates a transaction and broadcasts it to the network. The transaction is then received by multiple miners.

2. **Puzzle generation and computational effort**
   Each miner who receives the transaction generates a unique puzzle based on the current state of its blockchain and the received transaction. The puzzle requires finding a nonce that satisfies certain conditions, which incurs a significant computational cost. Each miner attempts to solve the puzzle by expending computational resources.

3. **Puzzle solving and dissemination of the answer**
   Once a miner successfully solves the puzzle, the corresponding answer is broadcast to the network.

4. **Verification and block addition**
   Other miners in the network verify the correctness of the answer. If the verification succeeds, the block is appended to their respective blockchains.

Through this mechanism, adding transactions to the blockchain entails a high computational cost, and the validity of blocks can be publicly verified.

2.2 Definition of Integrity

In general, integrity is one of the fundamental properties of a system, referring to the characteristic that data cannot be tampered with or corrupted by adversarial entities. In this study, we define the integrity of blockchain as follows:

Integrity: Only valid blocks are added to the blockchain, and the data and transactions contained within them are not subject to tampering.

This property is ensured by the computational effort of honest miners. Specifically, as long as honest miners who follow the protocol and perform legitimate computational effort remain in the majority, adversarial miners cannot surpass them in computational power, nor can they solve puzzles ahead of them. Consequently, adversarial miners cannot add invalid blocks or alter past blocks.

In this study, we construct a simple symbolic model to explain how this integrity is maintained, and we demonstrate within this model that integrity is achieved.

3 Formalization of PoW and Verification of Integrity Using the Strand Space Model

3.1 Messages

Let $\mathbb{T}$ denote the set of transactions and $\mathbb{B}$ the set of blockchains. Then, the set of messages $\mathbb{M}$ is defined as follows:

1. If $m \in \mathbb{T} \cup \mathbb{B}$, then $m \in \mathbb{M}$.
2. If $m_i \in \mathbb{M}$ $(1 \leq i \leq n)$, then the concatenated message $m_1 \dots m_n \in \mathbb{M}$.
3. If $b \in \mathbb{B}$ and $t \in \mathbb{T}$, then $\mathit{puzzle}(b, t) \in \mathbb{M}$.
4. If $b \in \mathbb{B}$ and $t \in \mathbb{T}$, then $\mathit{ans}(b, t) \in \mathbb{M}$.

1 indicates that both transactions and blockchains themselves are considered messages. 2 shows that if $m_1, \dots, m_n$ are each messages, then their concatenation $m_1 \dots m_n$ is also a message. 3 states that if b and t are a blockchain and a transaction, respectively, then the derived puzzle $\mathit{puzzle}(b, t)$ is a message. 4 similarly shows that the answer $\mathit{ans}(b, t)$ to the puzzle $\mathit{puzzle}(b, t)$ is also a message.

In this formalization, it is assumed that one transaction generates one block, which is then added to the blockchain. Accordingly, the puzzle $\mathit{puzzle}(b, t)$ and its answer $\mathit{ans}(b, t)$ in 3 and 4 are defined with respect to a single transaction t and a single blockchain b.

3.2 An Example of the PoW Protocol

Let T denote the transaction creator and M_i $(1 \leq i \leq n)$ the miners. The PoW protocol then proceeds as follows:

1. **Transaction generation and dissemination**
   T generates a transaction t and broadcasts it to all M_i.
2. **Puzzle generation and computational effort**
   Each M_i generates $\mathit{puzzle}(b, t)$ using the received transaction t and attempts to solve it.
3. **Puzzle solving and dissemination of the answer**
   The miner M_j who solves the puzzle disseminates both the puzzle $\mathit{puzzle}(b, t)$ and its answer $\mathit{ans}(b, t)$ to the other miners M_i.
4. **Verification and block addition**
   Each M_i who receives $\mathit{puzzle}(b, t)$ and $\mathit{ans}(b, t)$ verifies that the received answer corresponds to the received puzzle. If the verification succeeds, the transaction t is appended to the blockchain b.

3.3 Formalization of the Protocol

In this section, we formalize the PoW protocol described in 3.2 using the strand space framework.

3.3.1 Extended Messages

The computational effort required to solve a puzzle is formally expressed by means of *extended messages*. The set of extended messages $\mathbb{E}$ is defined as follows:

5. If $b \in \mathbb{B}$ and $t \in \mathbb{T}$, then $\mathit{effort}(b,t) \in \mathbb{E}$.

Here, $\mathit{effort}(b,t)$ represents the effort devoted to solving the puzzle $\mathit{puzzle}(b,t)$.

3.3.2 Strands and Their Extensions

A strand is defined as a sequence consisting of signed messages $+m, -m$, signed extended messages $+e$, or extended messages accompanied by the asterisk symbol $*$ and a minus sign $-e$. Here, the sign $+$ denotes transmission and $-$ denotes reception. Thus, $+m$ indicates the transmission of message m, while $-m$ indicates the reception of message m. Likewise, $+e$ indicates that the effort represented by an extended message has been made, and $*-e$ denotes that such effort has been sufficiently expended.

An extended strand is a strand marked with the symbol $*$, signifying that the operations represented by that strand are executed with sufficient frequency. Moreover, the $*$ symbol is assigned only to the dominant strand within a particular strand space, and there cannot exist more than one such symbol simultaneously.

3.3.3 Formalization of the Behavior of Protocol Participants

The actions of participants are formalized using strands.
Behaviors of the Transaction Creator and an Honest Miner

$$\begin{aligned} \mathit{Trans}(t) &= \langle +t \rangle \\ \mathit{Miner}(b,t,t') &= \langle -t,\ +\mathit{effort}(b,t), -\mathit{puzzle}(b,t'),\ -\mathit{ans}(b,t') \rangle \end{aligned}$$

$\mathit{Trans}(t)$ is the strand representing the transmission operation in which the transaction creator disseminates transaction t to the network. $\mathit{Miner}(b,t,t')$ is the strand representing the behavior of a miner maintaining blockchain b. The miner first receives transaction t, then performs the computational effort $\mathit{effort}(b,t)$ to solve $\mathit{puzzle}(b,t)$, and subsequently receives from another node the puzzle $\mathit{puzzle}(b,t')$ together with its answer $\mathit{ans}(b,t')$, verifying their validity.

At this point, it is possible that the transaction t on which the miner was working differs from the transaction t' that underlies the puzzle received from another node. Once the parameters are instantiated, the strands become concrete, and by combining the strands of the transaction creator and miners, the flow of the PoW protocol can be represented.

In this formalization, the Miner strand represents the behavior in which a miner learns the answer to a puzzle from another participant. In an actual Proof of Work (PoW) system, there exists at least one miner who solves the puzzle and disseminates the correct answer to other miners. However, such behavior is not explicitly modeled in this formalization. Since the number of miners who actually solve the puzzle is relatively small, this behavior has been abstracted away in the current, high-level formalization.

Formalization of the Behavior of Dishonest Miners The set of strands representing the behavior of dishonest miners is defined as follows:

$$
\begin{aligned}
&\mathit{Dishonest}(b) = \\
&\qquad \{\langle +t \rangle \mid t \in \mathbb{T}\} && (1)\\
&\qquad \cup \{\langle -m \rangle \mid m \in \mathbb{M}\} && (2)\\
&\qquad \cup \{\langle -m, +m, +m \rangle \mid m \in \mathbb{M}\} && (3)\\
&\qquad \cup \{\langle -m_1, -m_2, +m_1 m_2 \rangle \mid m_1, m_2 \in \mathbb{M}\} && (4)\\
&\qquad \cup \{\langle -m_1 m_2, +m_1, +m_2 \rangle \mid m_1, m_2 \in \mathbb{M}\} && (5)\\
&\qquad \cup \{\langle -t, +\mathit{puzzle}(b,t) \rangle \mid t \in \mathbb{T}\} && (6)\\
&\qquad \cup \{\langle -\mathit{puzzle}(b,t), +t \rangle \mid t \in \mathbb{T}\} && (7)\\
&\qquad \cup \{\langle -\mathit{puzzle}(b,t), +\mathit{effort}(b,t) \rangle \mid t \in \mathbb{T}\} && (8)
\end{aligned}
$$

$\mathit{Dishonest}(b)$ is the set of basic strands representing the possible behaviors of a dishonest miner maintaining blockchain b. The basic behaviors are as follows: (1) Generate a transaction and send it to the network. (2) Receive a message and discard it without disseminating it. (3) Duplicate a received message and transmit it multiple times (message duplication). (4) Concatenate multiple messages to generate a new concatenated message (message concatenation). (5) Decompose a concatenated message and extract its individual components (message decomposition). (6) Generate a puzzle for a transaction and transmit it to the network (puzzle generation). (7) Receive a puzzle and transmit the transaction from which the puzzle was derived. (8) Receive a puzzle and expend effort to solve it.

By combining these basic behaviors, a dishonest miner can carry out various attacks, such as tampering with transactions or promoting malicious growth of the blockchain.

Formalization of Puzzle Solvability If sufficient effort is expended to solve a puzzle, the puzzle and its corresponding answer can be obtained. This is formalized as follows:

$$\mathit{Solvable}(b) = \{\langle * - \mathit{effort}(b,t), +\mathit{puzzle}(b,t), +\mathit{ans}(b,t) \rangle \mid t \in \mathbb{T}\}$$

In this expression, $* - \mathit{effort}(b,t)$ denotes that sufficient effort has been devoted to solving the puzzle. $+\mathit{puzzle}(b,t)$ and $+\mathit{ans}(b,t)$ represent the puzzle and its corresponding answer, respectively.

3.3.4 Formalization of the PoW Protocol

The strand space in the case where honest miners outnumber dishonest miners is defined as follows. Here, a multiset $\Sigma(b,t)$ represents the situation in which both honest and dishonest miners maintain blockchain b, a transaction creator generates a transaction t, many honest miners attempt to add it to their blockchain, and dishonest miners attempt to disrupt this process.

$$
\begin{aligned}
&\Sigma(b,t) \subseteq \\
&\qquad \{\mathit{Trans}(t)\} && (9)\\
&\qquad \cup \{\mathit{Miner}(b,t,t') \mid t' \in \mathbb{T}\} && (10)\\
&\qquad \cup \{*\mathit{Miner}(b,t,t') \mid t' \in \mathbb{T}\} && (11)\\
&\qquad \cup \mathit{Dishonest}(b) && (12)\\
&\qquad \cup \mathit{Solvable}(b) && (13)
\end{aligned}
$$

(9) represents that a transaction creator generates transaction t. (10) and (11) indicate that many miners are attempting to add transaction t to blockchain b. (12) denotes the existence of a minority of dishonest miners who attempt to obstruct this process. (13) states that, in general, puzzle solvability holds.

Next, we assume that $\Sigma(b, t)$ satisfies the following conditions:

(i). Σ is a finite set.

(ii). If a node $-m$ exists, then the corresponding transmission node $+m$ exists.

(iii). If a node $* - e$ exists, then the corresponding $+e$ exists, and the strand containing it is marked with $*$.

(iv). There is no cycle in the causal or temporal ordering of send/receive operations within each strand, nor in the dependency relations defined by (ii) and (iii).

Condition (i) means that the overall behavior consists of a finite number of actions. Condition (ii) ensures that for every receiving node $-m$, there exists a corresponding sending node $+m$. Condition (iii) guarantees that the operation representing effort has been executed sufficiently many times, by means of the $*$ marker, to ensure that "sufficient effort has been made." Condition (iv) ensures that there is no cycle in the causal or temporal order of operations.

3.4 Formalization and Verification of Integrity

3.4.1 Formalization of Integrity

Integrity, in the case where honest miners outnumber dishonest miners, is formalized as follows:

Integrity In $\Sigma(b, t)$, for every $*\mathit{Miner}(b, t, t')$, it holds that $t = t'$.

Here, $\mathit{Miner}(b, t, t')$ represents the behavior of a miner who attempted to add a valid transaction t to the blockchain b but ultimately added transaction t'. By requiring $t = t'$, this formalization captures the property that "a valid transaction t is added to the blockchain b without being tampered with."

3.4.2 Verification of Integrity

Proposition 1. *Integrity holds. That is, in* $\Sigma(b, t)$*, for every* $*\mathit{Miner}(b, t, t')$*, it holds that* $t = t'$*.*

(Proof of Proposition 1)
Assume that $*\mathit{Miner}(b, t, t')$ exists in $\Sigma(b, t)$. In this case, the final node of $*\mathit{Miner}(b, t, t')$ is $-\mathit{ans}(b, t')$.
Hence, the first node that transmits $\mathit{ans}(b, t')$ is the third node of the strand $\langle * - \mathit{effort}(b, t'), +\mathit{puzzle}(b, t'), +\mathit{ans}(b, t')\rangle$ included in $\mathit{Solvable}(b)$. Therefore, this strand is contained in $\Sigma(b, t)$.

Since $* - \mathit{effort}(b, t')$ exists, there must also exist a $*$-marked strand that contains $+\mathit{effort}(b, t')$.
Among such $*$-marked strands, the only one containing this node is $\mathit{Miner}(b, t, \cdot)$. In this strand, the corresponding node is $+\mathit{effort}(b, t)$.
Therefore, $t = t'$ holds. □

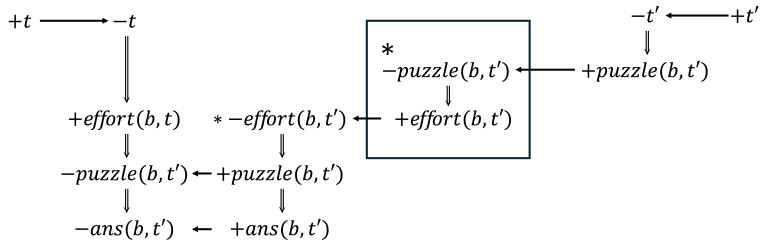

Figure 1: Counterexample to Integrity

3.5 Case Where Honest Miners Are in the Minority

3.5.1 Formalization of the PoW Protocol

The strand space in the case where honest miners are fewer than dishonest miners is redefined as follows:

$$\Sigma'(b,t) \subseteq \{\mathit{Trans}(t)\} \tag{14}$$
$$\cup \{\mathit{Miner}(b,t,t') \mid t' \in \mathbb{T}\} \tag{15}$$
$$\cup \mathit{Dishonest}{*}(b) \tag{16}$$
$$\cup \mathit{Solvable}(b) \tag{17}$$

Here, $\mathit{Dishonest}{*}(b)$ in (16) is obtained by adding $*$ to the strand defined in (8) within $\mathit{Dishonest}(b)$. This represents the behavior that occurs when a majority of miners are dishonest. In particular, (8) expresses the behavior of attempting to solve a puzzle, and the addition of $*$ indicates that this behavior is executed by many dishonest miners.

(14), (15), and (17) correspond to (9), (10), and (13) in the definition of $\Sigma(b,t)$. On the other hand, (11) is not adopted here, since it is inappropriate to include a $*$-marked miner strand when honest miners are not in the majority.

Furthermore, $\Sigma'(b,t)$ is also assumed to satisfy the conditions (i)–(iv) introduced in the previous subsection.

3.5.2 Formalization of Integrity

Integrity in the case where honest miners are in the minority is defined as follows:

Integrity In $\Sigma'(b,t)$, for every $\mathit{Miner}(b,t,t')$, it holds that $t = t'$.

The difference from the case where honest miners are in the majority lies in the use of $\mathit{Miner}(b,t,t')$ instead of $*\mathit{Miner}(b,t,t')$, since the number of honest miners is smaller.

3.5.3 Counterexample to Integrity

Proposition 2. *Integrity does not hold. That is, in $\Sigma'(b,t)$, there may exist $\mathit{Miner}(b,t,t')$ such that $t \neq t'$.*

(Proof of Proposition 2)
An example of $\Sigma'(b,t)$ containing $\mathit{Miner}(b,t,t')$ with $t \neq t'$ is shown in Fig. 1. Here, solid arrows indicate message send/receive relationships, while double arrows indicate causal order

within each strand. Moreover, strands enclosed in boxes labeled with $*$ denote those marked with $*$. □

This example of $\Sigma'(b,t)$ represents the situation in which a majority of dishonest miners attempt to add a transaction t' different from the valid transaction t to the blockchain, and succeed in doing so by solving the puzzle first.

As illustrated in Fig. 1, in PoW, the miner who first solves the puzzle earns the right to add the block. Therefore, if dishonest miners with abundant computational resources reach the correct solution earlier, even an invalid block may be added to the blockchain. This phenomenon can be regarded as a natural consequence of the PoW design, and the security guarantee critically depends on the assumption that honest miners dominate in computational resources.

4 Discussion

4.1 Level of Abstraction in Formalization

The strand space model introduced in this study focuses on the essential operations of the PoW protocol and enables verification of the integrity of PoW by representing participant behavior based on fundamental procedures, i.e., transaction generation, puzzle generation, effort and verification, without relying on detailed numerical values or probabilistic assumptions.

The advantage of such a formalization lies in its ability to qualitatively and intuitively represent the basic mechanism of PoW, namely, that "a block is generated and verified only through computational effort." In particular, by introducing the abstract extended message $\mathit{effort}(b,t)$ to represent computational effort and expressing its sufficiency with the $*$ symbol, the model makes it possible to explicitly show how the allocation of computational resources affects the security of the protocol.

On the other hand, limitations associated with abstraction also exist. For example, the determination of whether "effort is sufficient" is represented in this model by the presence or absence of the $*$ symbol, but the meaning of this $*$ implicitly includes assumptions such as the actual amount of computation and its distribution. Therefore, this model does not perform a quantitative evaluation of security, but rather provides a framework to show whether tampering succeeds or not under conditions of sufficient or insufficient effort.

4.2 Design Principle and Limitation of PoW

In this study, we formally demonstrated, using the strand space model, that integrity is satisfied when honest miners outnumber dishonest ones, and conversely, that it is not satisfied when they are fewer. In particular, in the counterexample shown in Figure 1, a situation arises in which the honest transaction is not appended to the blockchain and a dishonest one is added instead, due to the puzzle being solved by a dishonest miner.

This situation stems from a structural characteristic inherent in the design of the PoW protocol. That is, PoW emphasizes not "who is honest" but "who solves the puzzle first," and the validity of a block is judged solely by the correctness of its solution. For this reason, if dishonest miners who possess a large amount of computational resources solve the puzzle first, a block that differs from the intention of the honest miners may be accepted as valid.

This implies that the security of PoW depends on the assumption that honest miners are the majority in terms of computational effort. The formal description presented in this study explicitly reveals this structural limitation of PoW and confirms, in a formal manner, that the assurance of security is strongly influenced by the distribution of computational effort.

5 Related Work

This study lies at the intersection of two research areas: formal verification of security protocols and the evaluation of the security of Proof of Work (PoW) in blockchain technology.

First, symbolic approaches to protocol verification have been extensively studied. Representative examples include inductive approaches for proving properties via induction [10], the Strand Space model [6], and dedicated logical systems for verifying authentication protocols [4]. In this study, we extend [6] by incorporating not only communication messages but also extended messages representing computational effort, thereby modeling the causal relationships involved in block addition under PoW.

Second, regarding the security of PoW, many works have employed computational approaches. Representative studies include those of Gervais et al. [3] and Garay et al. [7]. Moreover, the seminal paper on Bitcoin by Nakamoto [9] also discusses the security of PoW. By symbolically modeling the PoW protocol, this study provides a new perspective for examining structural security aspects that have not been fully captured by conventional quantitative approaches.

6 Conclusion

In this study, we have attempted to describe and verify the integrity property of the Proof of Work (PoW) protocol in blockchain through a formal and abstract framework based on the strand space model. Whereas PoW has traditionally been evaluated using probabilistic techniques or numerical simulations, this study adopts a qualitative approach that focuses on the essential structure of PoW, namely, the fundamental principle of the protocol that valid blocks are added based on computational effort.

Specifically, by introducing the extended message $\mathit{effort}(b, t)$ and its sufficiency expressed as $*\mathit{effort}(b, t)$, we explicitly modeled the existence of computational effort by miners as part of the causal structure. We formally demonstrated that integrity is maintained when honest miners are in the majority. In addition, we constructively showed that in a strand space where dishonest miners are dominant, a counterexample exists in which an incorrect transaction is added to the blockchain, thereby revealing the structural limitations of PoW.

Through such formal description, this study provides a foundation for intuitively and logically understanding the structural conditions and assumptions under which PoW ensures security. In particular, it shows that the PoW design principle, where the validity of a block is determined by "who solves the puzzle first", can be expressed symbolically and that it inherently depends on the distribution of computational effort among participants.

We identify two directions for future research. The first is to construct a more mechanical and systematic method of security verification based on the abstract model proposed in this study. This would involve developing a formal system with axioms and inference rules that captures the properties and reasoning on the strand space model, thereby enabling systematic verification of integrity. This would allow current manual evaluations of PoW security to evolve into a more general and automatable framework.

The second direction is to establish the validity of the strand space model introduced in this study. Since the work by Abadi and Rogaway on bridging the gap in the indistinguishability of ciphertexts [1], there has been extensive researches such as [5][8] aimed at connecting symbolic and probabilistic verification frameworks. Applying a similar methodology to the analysis of PoW would be highly intriguing. By providing a correspondence between this abstract model and more detailed models using probabilistic assumptions, we can show that the properties

valid in the detailed models align with those in our model. For example, $*effort(b, t)$ in our model may correspond to the case where the computational hash rate to append transaction t to blockchain b exceeds 50% in a more detailed model. By establishing such correspondence between messages and properties, it will be possible to relate verification results across both models.

Although this study is merely an initial attempt to formally understand the structural properties of PoW, we hope that the direction it suggests will serve as a foundation for future theoretical and practical research.

References

[1] Martí;n Abadi and Phillip Rogaway. Reconciling two views of cryptography. In Jan van Leeuwen, Osamu Watanabe, Masami Hagiya, Peter D. Mosses, and Takayasu Ito, editors, *Theoretical Computer Science: Exploring New Frontiers of Theoretical Informatics*, pages 3–22, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[2] Iram Abrar and Javaid A. Sheikh. Current trends of blockchain technology: architecture, applications, challenges, and opportunities. *Discover Internet of Things*, 4(1):7, 2024.

[3] Karl Wüst Vasileios Glykantzis Hubert Ritzdorf Arthur Gervais, Ghassan O. Karame and Srdjan Capkun. On the security and performance of proof of work blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 3–16, NY, USA, 2016. ACM.

[4] Michael Burrows, Martin Abadi, and Roger Needham. A logic of authentication. *ACM Trans. Comput. Syst.*, 8(1):18–36, February 1990.

[5] Véronique Cortier and Bogdan Warinschi. Computationally sound, automated proofs for security protocols. In Mooly Sagiv, editor, *Programming Languages and Systems*, pages 157–171. Springer Berlin Heidelberg, 2005.

[6] Joshua D. Guttman F. Javier Thayer Fábrega, Jonathan C. Herzog. Strand spaces: proving security protocols correct. *J. Comput. Secur.*, 7(2–3):191–230, March 1999.

[7] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: Analysis and applications. *Journal of the ACM*, 71(4), August 2024.

[8] Daniele Micciancio and Bogdan Warinschi. Soundness of formal encryption in the presence of active adversaries. In Moni Naor, editor, *Theory of Cryptography*, pages 133–151. Springer Berlin Heidelberg, 2004.

[9] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf, 2008. Accessed: 2025-06-03.

[10] Lawrence C. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, (1-2):85–128.