TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 19/31 pp262–278
DOI: 10.26599/TST.2023.9010160
Volume 30, Number 1, February 2025

Jamming-Resilient Consensus for Wireless Blockchain Networks

Yifei Zou, Meng Hou, Li Yang*, Minghui Xu, Libing Wu, Dongxiao Yu, and Xiuzhen Cheng

Abstract: As the device complexity keeps increasing, the blockchain networks have been celebrated as the cornerstone of numerous prominent platforms owing to their ability to provide distributed and immutable ledgers and data-driven autonomous organizations. The distributed consensus algorithm is the core component that directly dictates the performance and properties of blockchain networks. However, the inherent characteristics of the shared wireless medium, such as fading, interference, and openness, pose significant challenges to achieving consensus within these networks, especially in the presence of malicious jamming attacks. To cope with the severe consensus problem, in this paper, we present a distributed jamming-resilient consensus algorithm for blockchain networks in wireless environments, where the adversary can jam the communication channel by injecting jamming signals. Based on a non-binary slight jamming model, we propose a distributed four-stage algorithm to achieve consensus in the wireless blockchain network, including leader election, leader broadcast, leader aggregation, and leader announcement stages. With high probability, we prove that our jamming-resilient algorithm can ensure the validity, agreement, termination, and total order properties of consensus with the time complexity of $O(n)$. Both theoretical analyses and empirical simulations are conducted to verify the consistency and efficiency of our algorithm.

Key words: consensus in blockchain; jamming attacks; distributed algorithm

1 Introduction

Over the past decade, blockchain networks have emerged as popular platforms attributed to their ability to provide distributed and immutable ledgers and data-driven autonomous organizations. Since the proposed digital currency project Bitcoin[1], the blockchain network is initially used as the foundational infrastructure for public and distributed ledger systems to facilitate the processing of asset transactions, which involve digital tokens among Peer-to-Peer (P2P) users. When considering open-access policies, blockchain networks stand out due to their intrinsic traits of disintermediation, public accessibility of network functions (such as data transparency), and resilience against tampering[2]. As a result, blockchain networks have gained recognition as the fundamental building blocks for numerous prominent FinTech applications, which place significant demands on the security and reliability of data, such as cryptocurrencies[3].

Roughly, the implementation of a blockchain network encompasses various core elements, including but not confined to the following: cryptographic hashing, digital signatures, and distributed consensus algorithms[4, 5]. Specifically, cryptographic hashing plays a crucial role in constructing Merkle trees and devising Proof-of-Work (PoW) puzzles, among other

• Yifei Zou, Meng Hou, Li Yang, Minghui Xu, Dongxiao Yu, and Xiuzhen Cheng are with School of Computer Science and Technology, Shandong University, Qingdao 266237, China. E-mail: yfzou@sdu.edu.cn; shizilihm@foxmail.com; 202020632@mail.sdu.edu.cn; mhxu@sdu.edu.cn; dxyu@sdu.edu.cn; xzcheng@sdu.edu.cn.
• Libing Wu is with School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China. E-mail: wu@whu.edu.cn.
* To whom correspondence should be addressed.
Manuscript received: 2023-10-21; revised: 2023-12-03; accepted: 2023-12-20

functions. Digital signatures serve to safeguard the integrity of data blocks within the blockchain. Additionally, distributed consensus algorithms ensure the uniformity of the distributed ledgers, which all participating nodes in the blockchain network adhere to when exchanging messages and making decisions. A blockchain network is, fundamentally, a consensus-dependent distributed system that ensures agreement on the status of specific data across distributed agents. Consequently, a distributed consensus algorithm stands as the central element directly shaping the performance and characteristics of a blockchain network. Through decentralized consensus, blockchains have the capability to facilitate and validate transactions within a mutually untrusted distributed system, eliminating the need for a trusted third party's involvement. In contrast to traditional transaction management systems where a centralized entity must validate transactions, blockchains accomplish decentralized transaction validation, leading to substantial cost savings and alleviating performance bottlenecks that are often associated with centralized entities. Hence, the consensus mechanisms form the foundational bedrock of blockchain networks, enabling trust and unanimity to be achieved without the need for any third party intervention.

Currently, most of the consensus algorithms in blockchain networks[4, 6–8] are considered based on some reliable wireless environments. Whereas, the inherent characteristics of shared wireless channels, such as fading, interference, and openness, pose significant challenges to achieving consensus within these networks, especially in the open-access wireless environment. In real-life wireless networks, the attacker/adversary is able to destroy legitimate communications by injecting a sufficiently large malicious jamming signal, which is also known as the jamming attack. As a matter of fact, even when jamming persists for just a small fraction of time slots, the widely used IEEE 802.11 MAC protocol fails in delivering any information[9]. In the context of blockchain networks, jamming attacks can be particularly damaging to the consensus mechanism. Since consensus algorithms rely on nodes exchanging information and agreeing on the next block to be added to the blockchain, if an attacker successfully floods the communication channels with noise or malicious data, it can lead to confusion among nodes, delay the consensus process, or even cause nodes to reach different conclusions about the state of the network. For example, in a PoW-based blockchain[10], miners compete to solve cryptographic puzzles. A jamming attack targeting the communication between miners can disrupt the timely propagation of new blocks, leading to inconsistencies in the blockchain and possibly causing forks in the physical layer. Thus, the consensus issue for blockchain networks under the threat of jamming attacks is of utmost importance for their real-world network deployment[11].

To depict the jamming attacks/behaviors, various jamming models have been proposed in open-access wireless environments. In Ref. [12], Pirayesh et al. proposed a constant jamming model, where the jamming occurs constantly due to some natural faults. In Ref. [13], Awerbuch et al. introduced a jamming-resistant Media Access Control (MAC) algorithm for single-hop wireless networks, even when the jammer possesses knowledge of the algorithm and complete communication history. This is also known as adaptive jamming[14]. Richa et al.[15] presented self-stabilizing Leader Election (LE) for single-hop wireless networks despite reactive jamming. Different from the adaptive jamming model, the jammer in reactive jamming further knows the current network state/information. However, the proposed jamming models do not consider the crucial consensus problem in blockchain networks. In the context of blockchain networks, King and Nadal[16] introduced the Proof-of-Stake (PoS) concept via Peercoin. This concept aims to eliminate the need for resource-intensive hashing competitions and enhance the energy efficiency in block generation by freeing miners from resource requirements. These works investigate the consensus problem in blockchain networks, but are not resistant to jamming attacks. More recently, Xu et al.[11] simultaneously investigated the consensus mechanism and malicious jamming in wireless blockchain networks. Simulation results showed that their RAFT-based consensus algorithm achieves jamming resiliency. Whereas, the considered jamming phenomenon is binary-based, i.e., they considered the uplink/downlink communications for blockchain networks with or without the presence of malicious jamming. So, a more real-life and comprehensive jamming should be taken into account.

To the best of our knowledge, there exist only a few works that jointly consider the crucial consensus

problem and jamming attacks in blockchain networks, let alone the more realistic jamming scenario in the open-access wireless environment. Therefore, it is desirable to design a consensus algorithm for blockchain networks despite such a more realistic jamming model. Inspired by the work in Ref. [17], we adopt the slight jamming model to depict the jamming attacks in the wireless blockchain network. In this paper, we propose the first distributed consensus algorithm for blockchain networks in the presence of slight jamming that has an asymptotically optimal time complexity. The main contributions of our work are summarized as follows:

- Compared with most existing works on consensus for blockchain networks on reliable wireless communications, in this paper, we consider the crucial consensus problem in the open-access wireless environment, especially in the presence of malicious jamming attacks. We adopt the slight jamming to character the real-life wireless blockchain network, which is more realistic than the binary jamming models.
- Based on the realistic jamming model, we investigate the crucial consensus problem among n participating physical devices in a single-hop wireless blockchain network. To address this problem, an efficient distributed consensus algorithm is proposed to ensure the resiliency in a blockchain network within $O(n)$ running rounds with high probability (w.h.p.), i.e., with a probability of $1-n^{-c}$ for some constant $c>1$. Considering that the lower bound for blockchain consensus is $\Omega(n)$ with the assumption that the elected leader is able to aggregate one miner's report per round, the time complexity achieved by our algorithm is also asymptotically optimal.

Extensive simulations are conducted to validate the theoretical result of our algorithm.

Organization. We organize the remainder of this paper as follows: Section 2 introduces the related work and Section 3 gives some preliminaries and problem statement. The algorithm description and analyses are shown in Sections 4 and 5, respectively. The experimental results are conducted in Section 6. Lastly, we conclude this work in Section 7.

2 Related Work

The consensus algorithm plays a critical role in blockchain networks. It is the mechanism through which the distributed network of nodes reaches a consensus on the blockchain's state and verifies transactions. In the past decades, various consensus algorithms are employed in blockchain networks, each with its own characteristics and trade-offs. In Ref. [16], King and Nadal proposed the PoS concept via Peercoin. Within a PoS system, validators are selected to generate new blocks based on the quality of cryptocurrency they possess and are willing to use "stake" as collateral. Greater stake leads to an increased chance of selection. In addition, the Proof-of-Authority (PoA)[18] belongs to a group of consensus algorithms used in blockchain networks, where validators are selected based on their reputation or authority in the network, rather than on their stake or computational power. This series of algorithms is often used in private or consortium blockchains, in which participants are known entities. Xu et al.[4] presented a fast consensus algorithm in a single-hop wireless blockchain system. As yet another example, Ref. [8] considered the critical consensus problem for blockchains in multiple-hop Internet of Things networks. It has been proven that these consensus algorithms achieve consistency in a single/multiple-hop wireless network within $O(\log n)$ running rounds, which is the well-known lower bound for a successful transmission in wireless networks[19]. Their works achieve a fast consensus based on the reliable wireless environment. However, in real-life wireless blockchain networks, the shared wireless channel is unreliable and vulnerable to jamming attacks due to its openness-prone.

In the open-access wireless environment, it is easy for the un-permitted adversary to fault the message delivery in blockchain networks via injecting malicious jamming attacks. Several published works proposed various jamming models to characterize the jamming phenomenon in a wireless channel. Jamming can also be described as the deliberate disruption of wireless communication signals, aimed at disturbing the regular functioning of a network. Two widely used jamming models are the adaptive jamming model and the reactive jamming model. For the former, the works in Refs. [13, 20, 21] proposed jamming-resilient MAC algorithms despite adaptive jamming, in which the adversary has the knowledge of the designed algorithm and the past history information. In this way, the adversary is able to decide whether to jam the current

running round or not. For the latter, Zou et al. in Ref. [22] constructed a distributed backbone network in the presence of strong adversarial jamming. Different from the adaptive jamming model, the adversary in reactive jamming additionally possesses knowledge of the current network information. In addition to the above adaptive and reactive jamming patterns, several alternative jamming models have been put forth. These encompass the intelligent jamming discussed in Refs. [23, 24], the strategic jamming described in Ref. [25], the disguised jamming in Ref. [26], and the follower jamming elaborated upon in Ref. [27]. While these models present various perspectives on jamming behaviors, they remain bound by fundamental limitations, i.e., binary-based jamming.

In this paper, we consider the crucial consensus problem for wireless blockchain networks in the presence of malicious adversary jamming. Compared with previous jamming models that focus mainly on the binary-based jamming, we present a distributed consensus algorithm for blockchain networks under a non-binary jamming model, which is more general and realistic.

3 Preliminary and Problem Statement

We consider the consensus problem in a single-hop wireless blockchain network, where a collection V of n physical devices is placed arbitrarily within a two-dimensional Euclidean space. These devices, constituting the blockchain network, are referred to as miners (also known as nodes) within the wireless blockchain system. The Euclidean distance between any two miners, denoted as u and v, is represented as $d(u,v)$. Our algorithm divides time into synchronized slots, each being the smallest time unit for transmitting messages/information. During each of these slots, a miner v can choose to either transmit or listen. A round is defined as a period comprising a fixed number of slots, e.g., two slots. In order to accommodate networks that have a mix of half-duplex and full-duplex transceivers, we assume that each miner is equipped with a half-duplex transceiver. Consequently, in any given time slot, a miner can engage in either transmission or reception, but not both. Particularly, within each round, every miner strives to exchange information with other miners using a common wireless communication channel. However, it is worth noting that malicious adversaries have the capability to initiate jamming attacks on this channel, with the intention of obstructing the successful delivery of legitimate messages.

In this section, we first give the communication model based on Rayleigh fading. Then, we describe our adopted non-binary jamming pattern. Finally, we introduce the consensus and problem statement.

3.1 Rayleigh fading communication model

We utilize the Rayleigh fading model to represent the reception of signals, wherein interference and contention arise from simultaneous transmissions among miners. Let $\mathrm{Signal}(u,v)$ denote the signal strength emanating from miner u and received by v during a time slot. Given that the strength of $\mathrm{Signal}(u,v)$ diminishes with distance and is subjected to inherent uncertainty within the complex real-world setting, we employ a Rayleigh fading model to capture the unpredictability associated with signal reception, as discussed in Ref. [17]. Precisely, $\mathrm{Signal}(u,v)$ assumes the form of a random variable following an exponential distribution with a mean of $S(u,v)=P_u/d(u,v)^{\alpha}$, where P_u signifies the transmission power of miner u and $\alpha\in(2,6]$ represents the path-loss exponent. When a transmission originates from miner u to v, we define $\mathrm{SINR}(u,v)$ as the Signal-to-Interference-plus-Noise Ratio (SINR) rate, which can be expressed as

$$\begin{cases}\mathrm{Signal}(S,v)=\sum\limits_{w\in S}\mathrm{Signal}(w,v),\\ \mathrm{SINR}(u,v)=\dfrac{\mathrm{Signal}(u,v)}{\mathrm{Signal}(S\setminus\{u\},v)+N(v)}\end{cases}\tag{1}$$

where S denotes the collection of miners transmitting during the present time slot, w indicates a miner, $\mathrm{Signal}(S,v)$ corresponds to the sum of signal strengths accumulated at miner v from the transmitters in the set S, $\setminus$ denotes the difference set, and $N(v)$ signifies the non-zero ambient noise present at miner v. The value of this noise is determined by the prevailing environment or the influence of jamming adversaries. Notably, when $\mathrm{SINR}(u,v)$ is equal to or surpasses the threshold β, miner v is capable of deciphering the message transmitted by miner u. Here, $\beta\geqslant 1$ stands as a hardware-dependent minimum SINR threshold necessary for achieving a successful transmission.

The blockchain network lacks a prior topology structure. For any miner v operating with a transmission power P, the transmission range R is the maximum distance over which another miner, denoted

as u, can reliably receive the message sent by v. With this defined distance R, any pair of miners within the blockchain network is interconnected based on this range R to guarantee that the network takes on a single-hop configuration. We also standardize the minimum distance between any pair of miners to 1. If no other simultaneous transmissions over the wireless blockchain networks, then, deriving from the SINR Eq. (1), $R = (P/\beta N)^{1/\alpha}$.

3.2 Non-binary jamming pattern

Inspired by Ref. [17], we adopt the slight jamming model to illustrate the phenomenon of jamming, which is the typical non-binary jamming model. Specifically, in view of the ambient noise, the wireless network can be categorized into three cases: the non-jamming, slight jamming, and heavy jamming cases[17]. The non-jamming scenario describes that ambient noise remains very low to avoid impacting transmissions. Conversely, heavy jamming signifies an extreme level of ambient noise, leading to unsuccessful transmissions across the entire network, even for miner pairs situated at the smallest distance and utilizing the maximum available transmission power. The slight jamming category encapsulates intermediate scenarios not covered by either the non-jamming or heavy jamming cases. Of particular note is that a significant portion of jamming incidents realistically fall within the slight jamming category. Unlike the uniform jamming rooted in the graph model in Ref. [21], our non-binary jamming model relies on the Rayleigh fading model. For the sake of simplicity, we assume the presence of a malicious adversary responsible for determining the ambient noise $N(v)$ associated with miner v during each round. In this paper, the variation in jamming over the shared channel is round-based, implying that jamming characteristics remain consistent for each miner within a given round. One noteworthy constraint on the ambient noise imposed by the adversary for any miner v is that $N(v) \leqslant \ln t \times \dfrac{P}{R^{\alpha} \times \beta}$, where constant t serves as the jamming parameter in our model.

3.3 Consensus and problem statement

There has been comprehensive research on consensus mechanisms in distributed systems for nearly thirty years, aiming to facilitate unanimous agreement among all participating nodes on shared data or states. These mechanisms are generally expected to adhere to the following key properties: agreement (where all nodes converge on the same value), termination (ensuring that all nodes conclude within a finite timeframe), and validity (requiring the decision value to originate from a node's input)[28]. Our primary focus revolves around meeting these aforementioned requisites while designing a consensus algorithm resilient to jamming within a blockchain network. In a blockchain network, the history of network transactions is meticulously recorded by all participating miners, with these transactions organized into blocks and subsequently linked in a chain-like manner. Similarly, the objective of a blockchain consensus algorithm is to ensure unanimous agreement among participating miners regarding the chronological sequence of network transactions within the blockchain. Considering the wireless blockchain network under non-binary jamming, a jamming-resilient consensus algorithm should satisfy the following properties:

- **Validity.** When updating a new block into the individual miners' local blockchains, the agreed-upon transactions within this new block should align precisely with the historical transactions of the blockchain system.
- **Agreement.** All miners should eventually agree on the same value or decision, i.e., accept or discard a new block.
- **Termination.** Every miner ultimately reaches a decision within a finite time to either discard or incorporate the new block into its local blockchain.
- **Total order.** All miners agree on the order of all proposed blocks and their local blockchains should have the same sequence of blocks.

Problem statement. In this paper, we consider the crucial consensus problem for a single-hop wireless blockchain network under a non-binary jamming model. Specifically, we aim to design a jamming-resilient blockchain consensus issue. Formally, the problem can be cast as follows: given the Rayleigh fading model, each miner has the knowledge of R, P, as well as parameters c, α, β, and γ, where $\gamma \geqslant c + \ln(R \ln n \times [1 + c\beta(n-1)] - n)$ is a constant. And the adversary has the capacity to determine the ambient noise $N(v)$ associated with each miner v. We are asked to design an efficient distributed algorithm, such that all participating miners agree on the new block and then record the network transactions history into the blockchain while resisting the malicious jammer.

Knowledge and capability of nodes. Initially, all miners initiate their operations (transmission or listen) and compete for leadership to propose new blocks. They possess knowledge of the total number of miners, denoted as n within the network, and the list of parameters (R, P, c, α, β, and γ). Importantly, miners are not required to be aware of the number of their neighbors. Every miner is equipped with a half-duplex transceiver featuring physical carrier sensing capability. The concept of physical carrier sensing adheres to the IEEE 802.11 MAC standard and has been widely employed in prior studies within the wireless domain[29−32]. This mechanism facilitates the monitoring of signals within the channel when miners are in a listening state.

4 Distributed Jamming-Resilient (DJR) Consensus

The underlying idea of achieving consensus in a blockchain network relies on solving two crucial issues: How to collect all reports of participating miners, and who is responsible for collecting these reports and making a decision on consensus. To address the above two problems, a classic solution is to let a miner break the symmetry of the blockchain network, i.e., let an elected leader aggregate all miners' reports, and make/broadcast the decision to all other miners. Whereas, in the open-access wireless environment, the shared wireless channel is unreliable and jamming-prone. The jammer can inject jamming signals and jam the shared channel, which destroys the block delivery among miners. In this way, the leader election progress would be failed, let alone the decision-making or block transmission. In other words, the malicious jamming behavior incurs great difficulty in reaching consensus in blockchain networks.

To cope with the mentioned difficulty, we show our distributed jamming-resilient consensus algorithm for blockchain networks. Basically, the algorithm includes four stages: the leader election stage, the Leader Broadcast (LB) stage, the Leader Aggregation (LA) stage, and the leader announcement stage. Specifically, we initially present an oblivious jamming-resilient leader election scheme to elect a leader within $O(\log n)$ rounds. This oblivious leader election strategy is simple yet powerful for jamming-tolerance. For instance, the statistical exponential back-off algorithm in Ref. [13] is an adaptive scheme to resolve the contention resolution issue with an asymptotically optimal $\Omega(\log n)$ bound in the unjamming setting. Considered in a jammed scenario, the adaptive algorithm may not converge since each node cannot distinguish the contention and jamming signal when it receives a message. Secondly, we design a leader broadcast subroutine to resist non-binary jamming by leveraging the Rayleigh fading model. Thereafter, using specific power control, we present a leader aggregation algorithm to collect all nodes' reports despite the malicious jamming. Lastly, a leader announcement subroutine is proposed for updating/discarding local blockchains.

In this section, we present our distributed jamming-resilient consensus algorithm for wireless blockchain networks, including the leader election stage, the leader broadcast stage, the leader aggregation stage, and the leader announcement stage. The pseudo-code of our DJR-consensus is presented in Algorithm 1. Before introducing Algorithm 1 in detail, we first give a definition of the Bernoulli random variable as below:

Definition A Bernoulli random variable X takes on the options transmission or listen such that $\Pr[X = \text{transmission}] = p$ and $\Pr[X = \text{listen}] = 1 - p$, where probability $p \in (0, 1)$ is a constant.

Leader election stage. The objective of this stage is to elect a leader among all participating miners, who is tasked with creating a new block. The underlying idea is to consistently engage each miner in competition with others, and the still transmission miner in this competition procedure will be the leader. In our algorithm execution, there are three states for miners, namely A, I, and L. Concretely, State A denotes a

Algorithm 1 DJR-consensus for node v

initialization: state_v = A; $\text{count}_1 = 0$; $\text{count}_2 = 0$;
Stage 1: leader election stage
for $a_1 \times \log n$ rounds **do**
  if state_v = A **then**
    in Slot 1:
    listen on the channel; N_1 = Signal(v);
    in Slot 2:
    if (X = transmission) **then**
      transmit its message in the channel;
    else
      listen on the channel; N_2 = Signal(v);
      if $N_2 > N_1$ **then**
        $\text{state}_v \leftarrow$ I;
  else
    do nothing;
if state_v = A **then**
  $\text{state}_v \leftarrow$ L;

(To be continued)

(Continued)

Algorithm 1 DJR-consensus for node v

Stage 2: leader broadcast stage
for $a_2 \times \log n$ rounds **do**
  in each slot:
  if state_v = L **then**
    transmit its message associated with block B_v;
  else
    listen on the channel;
if state_v = L **then**
  $\text{state}_v \leftarrow$ I;
if state_v = I **then**
  $\text{state}_v \leftarrow$ A;

Stage 3: leader aggregation stage
Compute a geometric random variable g_v;
$f(g_v) = P \times (2^{g_v} \times g_v^4)^{\gamma \times 2^{g_v} \times g_v^4}$;
$P_v \leftarrow$ randomly and uniformly selected from $[f(g_v), 2f(g_v)]$;
for $a_3 \times n$ rounds **do**
  if state_v = A **then**
    in Slot 1:
    verify the block BLOCK $\leftarrow$ {approval, reject};
    transmit its message M_v (BLOCK, P_v) with power P_v;
    in Slot 2:
    listen on the channel;
  if state_v = I **then**
    in Slot 1;
    listen on the channel;
    if receive the message report M_v' (BLOCK, P_v'), where P_v' is the transmission power **then**
      if the report of BLOCK is approval **then**
        $\text{count}_1 \leftarrow \text{count}_1 + 1$;
      if the report of BLOCK is reject **then**
        $\text{count}_2 \leftarrow \text{count}_2 + 1$;
    in Slot 2;
    transmit ack. message with power P_v';

Stage 4: leader announcement stage
for $a_4 \times \log n$ rounds **do**
  in each slot:
  if state_v = I **then**
    record count_1 and count_2;
    transmit announcement message A_v (count_1, count_2);
  if state_v = A **then**
    listen on the channel;
    if receive the announcement message A_v' (count_1, count_2) **then**
      if $\text{count}_1 > \text{count}_2$ **then**
        update the local block BLOCK;
      else
        discard the local block BLOCK;

miner actively participating in leader competition. State I signifies a miner forfeiting the leader election and remaining inactive during the current phase. And State L designates a miner as the leader, responsible for proposing a new block to all participating miners. Initially, all miners are in State A, indicating their readiness for leader election. Miner v's status is referred to as state_v, while Signal(v) represents the signal strength detected by v in the current slot. The delay bound of this stage is $a_1 \times \log n$ rounds (analyze later), in which n is the number of miners and a_1 is a positive constant. Within each round, active miners first listen to the channel in Slot 1, then in Slot 2, transmit with a constant probability of p, or alternatively, listen with a constant probability of $1 - p$. By comparing signal strengths from Slot 1 and Slot 2, an active miner can ascertain the presence of other miners' competition for leadership. If $N_2 > N_1$, where N_2 is the signal strength in Slot 2 per round in the leader election stage and N_1 represents the signal strength in Slot 1 per round in the leader election stage, the miner will turn to State I and quit the leader election process. At the end of the leader election stage, any miner still in State A will be selected as the leader and turn to State L, which is also called breaking symmetry.

Leader broadcast stage. After the leader election stage, the elected leader/miner v will record the network transactions history into a block (BLOCK) and broadcast such a block with transmission power P. It aims to let every participating miner know the generated block. However, implementing this procedure presents severe jamming attacks caused by a malicious adversary, and it is hard for other miners to quickly receive the new block proposed by the leader. Different from the leader election stage, which only needs to elect a leader through the competition signals and does not involve block delivery, in the leader broadcast stage, the elected leader should broadcast the message associated with its proposed block and all other miners keep listening while decoding the message. With the help of the Rayleigh fading model, all the non-leader miners can receive the block from the leader within $a_2 \times \log n$ rounds w.h.p., where constant a_2 is sufficiently large. When the leader completes the broadcast, it will change to State I, which means that it has already broadcast the new block, and keeps inactive in the next leader aggregation stage. Besides, all other miners in State I will turn to State A.

Leader aggregation stage. In the leader aggregation stage, the elected leader endeavors to collect all other miners' reports as soon as possible despite the non-binary jamming. To begin with, each miner takes action on preprocessing. Concretely, each miner v

computes a geometric random variable g_v, which represents the consecutive occurrences of tails in coin flips until the first head is encountered. This is a Bernoulli $(1/2)$ trial. After that, miner v selects transmission power uniformly at random from the interval $[f(g_v), 2f(g_v)]$, where $f(g_v) = P \times (2^{g_v} \times g_v^4)^{\gamma \times 2^{g_v} \times g_v^4}$ is the power-transfer function. Similar to the work in Ref. [33], there is a unique miner with the highest transmission power w.h.p. Using the specific power control, the leader is capable of aggregating all miners' reports within $a_3 \times n$ rounds since it receives one report per round w.h.p., in which a_3 is a positive constant. Each round consists of two time slots. In Slot 1, the non-leader miner v (A) firstly verifies the block BLOCK and gives its answer: approval or reject. By incorporating the answer bits into the message, it then transmits the message report M_v(BLOCK) to the leader with power P_v. Meanwhile, the leader (miner at State I) listens on the channel. If the received report of BLOCK is approval, then the number of approved miners is increased by one. If the received report of BLOCK is reject, then the number of dissenting miners is increased by one as well. We use parameters count_1 and count_2 to record the number of approved miners and dissenting miners, respectively. In Slot 2, the leader transmits an acknowledgment (ack. for short) message with power P' while other mines keep listening on the channel. Note that the transmission power P' is identical to the power of a non-leader miner that is transmitted in Slot 1.

Leader announcement stage. When ending the leader aggregation stage, the leader (i.e., the miner v at State **I**) will count the number of approved miners and dissenting miners, denoted by count_1 and count_2, respectively. By incorporating count_1 and count_2 into the announcement message A_v, the miner v transmits the message $A_v(\text{count}_1, \text{count}_2)$ to all other miners. At the same time, the miners at State **A** listen to the channel. If they receive the announcement message and find that $\text{count}_1 > \text{count}_2$, then these miners decide to update the local block BLOCK. Otherwise, they choose to discard this local block BLOCK.

5 Analysis on Correctness and Validity

In this section, we show the correctness and time complexity of our proposed distributed jamming-resilient consensus algorithm. To this end, we first argue that the leader will be elected at the end of the leader election stage within $O(\log n)$ rounds against slight jamming w.h.p. Then, we prove that the leader broadcast stage takes $O(\log n)$ rounds, the leader aggregation stage needs at most $O(n)$ rounds, and the leader announcement stage takes $O(\log n)$ rounds w.h.p., respectively. In addition, we will analyze that our DJR-consensus algorithm well satisfies the mentioned properties: validity, agreement, termination, and total order. Formally, we prove the correctness and validity of our DJR-consensus algorithm via Lemmas 1–4 and Theorem 1.

Lemma 1 In the leader election stage, a leader will be elected within $O(\log n)$ rounds w.h.p.

Lemma 2 After $O(\log n)$ rounds, the elected leader broadcasts the proposed block to all other miners despite the slight jamming w.h.p.

Lemma 3 Despite the slight jamming, the leader can collect all other miners' reports within $O(n)$ rounds in the leader aggregation stage w.h.p.

Lemma 4 It takes at most $O(\log n)$ rounds to complete the leader announcement stage w.h.p.

Theorem 1 With high probability, our distributed jamming-resilient consensus algorithm is able to reach a consensus against slight jamming for wireless blockchain networks within $O(n)$ rounds.

5.1 Proof of Lemma 1

Within each round, there always exist some miners giving up the leader competition.

Lemma 5 In the leader election stage, within each round r, when there are some miners transmitting and another miner listening, there will always be a miner that gives up the leader competition and moves to State I.

Proof In the first slot of round r, all participating miners/nodes listen on the channel and sense the ambient noise, whose signal strength is recorded by parameter N_1. Then, in the second slot, each miner transmits with a constant probability p, or listens with a complement probability $1-p$. Taking a listener v as an example, the received signal strength in Slot 2 is $\text{Signal}(v) = \sum_{w \in S} \text{Signal}(w, v) + N(v)$, where S denotes the set of all simultaneously transmitting miners. It can be seen that $N_2 > N_1$ if S is non-empty. From our leader election stage, once a miner v has sensed that $N_2 > N_1$ in Slot 2, it immediately gives up the leader competition and turns to State I. So, we proved Lemma 5. ■

Lemma 6 In each round of the leader election

stage, with probability $1-\mathrm{e}^{-\delta^2\varphi/3}-\mathrm{e}^{-\delta^2\varphi/2}$, where δ and φ are two positive constants, there are at least $1-(1+\delta)p$ fraction of miners giving up the leader election, until only one active miner left.

Proof $V(r)$ represents the set of active miners at the beginning of round r. Let x_v be the random variable as follows:

$$x_v=\begin{cases}0, & \text{if } v \text{ listens in Slot 2 of round } r;\\ 1, & \text{if } v \text{ transmits in Slot 2 of round } r\end{cases} \tag{2}$$

Let φ_1 be the expectation of $\sum_{v\in V(r)} x_v$, we have that $\varphi_1=E\left[\sum_{v\in V(r)} x_v\right]=\sum_{v\in V(r)} p$. Applying Chernoff bound (please refer to the Appendix) with a constant $\delta\in(0,\min\{1/p-1,1\})$, we get $\Pr\left[\sum_{v\in V(r)} x_v\geqslant(1+\delta)\varphi_1\right]\leqslant \mathrm{e}^{-\delta^2\varphi_1/3}$ and $\Pr\left[\sum_{v\in V(r)} x_v\leqslant(1-\delta)\varphi_1\right]\leqslant \mathrm{e}^{-\delta^2\varphi_1/2}$. Then, define ε_1 to be the event that there are at least $(1-\delta)p$ fraction of miners transmitting and at least $1-(1+\delta)p$ fraction of the miners listening, and $\Pr[\varepsilon_1]$ to be the corresponding probability. $\Pr[\varepsilon_1]\geqslant 1-\mathrm{e}^{-\delta^2\varphi_1/3}-\mathrm{e}^{-\delta^2\varphi_1/2}$ is the probability that ε_1 occurs. Thus, at least with probability of $1-\mathrm{e}^{-\delta^2\varphi_1/3}-\mathrm{e}^{-\delta^2\varphi_1/2}$, there exist $1-(1+\delta)p$ fraction of miners listening in Slot 2 of round r, combined with Lemma 5, Lemma 6 is proved directly. ■

Lemma 7 After $O(\log n)$ rounds in the leader election stage, one and only one active miner will be left w.h.p.

Proof According to Lemma 6, we can obtain that every time ε_1 occurs, the number of active miners decreases with a factor of $(1+\delta)p$ $(p<1/2)$, and when ε_1 happens $\log_{(1+\delta)p}\frac{1}{n}$ times, there is only one active miner left in the blockchain network. Considering an interval I_1 with length of k_1 rounds, let ε_2 be the event that ε_1 occurs for $\log_{(1+\delta)p}\frac{1}{n}$ times within this interval, and $\Pr[\varepsilon_2]$ to be the corresponding probability.

Then, we focus on bounding the probability of ε_2 occurring in interval I_1. Denote $x(r)$ to be the random variable as follows:

$$x(r)=\begin{cases}0, & \text{when } \varepsilon_1 \text{ does not occur in round } r;\\ 1, & \text{when } \varepsilon_1 \text{ occurs in round } r\end{cases} \tag{3}$$

So, we have $\Pr[\varepsilon_2]=\Pr\left[\sum_{r\in I_1} x(r)\geqslant\log_{(1+\delta)p}\frac{1}{n}\right]$. Also, let φ_2 be the expectation of the time ε_1 occurring within interval I_1. As aforementioned, ε_1 happens in each round at least with a probability of $1-\mathrm{e}^{-\delta^2\varphi_1/3}-\mathrm{e}^{-\delta^2\varphi_1/2}$, we can get $\varphi_2=k_1\times(1-\mathrm{e}^{-\delta^2\varphi_1/3}-\mathrm{e}^{-\delta^2\varphi_1/2})$. Applying the Chernoff bound associated with a constant $\delta_1\in(0,1)$, we have

$$\Pr\left[\sum_{r\in I_1} x(r)\leqslant(1-\delta_1)\varphi_2\right]\leqslant \mathrm{e}^{-\delta_1^2\varphi_2/2} \tag{4}$$

Substituting φ_2 into Eq. (2) and setting $k_1=\frac{c_1\times\log n}{1-\mathrm{e}^{-\delta^2\varphi_1/3}-\mathrm{e}^{-\delta^2\varphi_1/2}}$, where constant c_1 is sufficient large, it leads to

$$\begin{aligned}&\Pr\left[\sum_{r\in I_1} x(r)\leqslant(1-\delta_1)k_1\times(1-\mathrm{e}^{-\delta^2\varphi_1/3}-\mathrm{e}^{-\delta^2\varphi_1/2})\right]\leqslant\\ &\mathrm{e}^{-\delta_1^2\times k_1\times(1-\mathrm{e}^{-\delta^2\varphi_1/3}-\mathrm{e}^{-\delta^2\varphi_1/2})/2}=\\ &\mathrm{e}^{-\delta_1^2\times c_1\times\log n/2}=n^{-\delta_1^2\times c_1/2}\end{aligned} \tag{5}$$

Letting $\delta_1=1-\frac{\log_{(1+\delta)p}\frac{1}{n}}{c_1\times\log n}\in(0,1)$, we obtain

$$\begin{aligned}\Pr[\varepsilon_2]=\Pr\left[\sum_{r\in I_1} x(r)\geqslant\log_{(1+\delta)p}\frac{1}{n}\right]=&\\ \Pr\left[\sum_{r\in I_1} x(r)\geqslant(1-\delta_1)\varphi_2\right]=&\\ 1-\Pr\left[\sum_{r\in I_1} x(r)<(1-\delta_1)\times c_1\times\log n\right]\geqslant&\\ 1-n^{-\delta_1^2\times c_1/2}=1-n^{-O(1)}&\end{aligned} \tag{6}$$

Hence, we have proved that event ε_2 occurs with high probability. ■

Considering the conclusions in Lemmas 5–7, we can derive that the constant factor, termed as a_1, behind the $O(\log n)$ is $a_1\geqslant\frac{c_1}{1-\mathrm{e}^{-\delta^2\varphi_1/3}-\mathrm{e}^{-\delta^2\varphi_1/2}}$. Each miner is featured with physical carrier sensing, which is helpful for detecting whether there are other active miners in the leader election stage and facilitates the leader election procedure against jamming. However, in the leader broadcast stage, the physical carrier sensing is no longer useful because we need to disseminate exact blocks under jamming. Surprisingly, leveraging the uncertainty and probability of the Rayleigh fading model, we are going to prove that the uncertainty is beneficial for communications under jamming.

5.2 Proof of Lemma 2

Lemma 8 During every time slot of the leader broadcast stage, each miner has the ability to receive the message/block from the leader with a consistent probability of at least $1/t$.

Proof As mentioned before, we assume that the ambient noise at each miner v caused by the adversary is limited by $N(v) \leqslant \ln t \times \dfrac{P}{R^{\alpha} \times \beta}$. Define ε_3 to be the event that the strength of signal $\text{Signal}(u,v) \geqslant \ln t \times \dfrac{P}{R^{\alpha}}$ for the message delivery from a leader u to a non-leader miner v, and $\Pr[\varepsilon_3]$ to be the probability associated with event ε_3. Then, from the exponential distribution of the Rayleigh fading model associated with $\text{Signal}(u,v)$, we know that

$$\Pr[\varepsilon_3] = \Pr\left[\text{Signal}(u,v) > \ln t \times \frac{P}{R^{\alpha}}\right] = e^{-\ln t \times \frac{P}{R^{\alpha}}/S(u,v)} \geqslant \frac{1}{t} \tag{7}$$

We define $\lambda = \frac{1}{t}$ as our slight variable in the slight jamming model. If the value of λ is close to 1, then the probability of event ε_3 is in close proximity to 1, and vice versa.

Also, in each slot of the leader broadcast stage for the transmission from the leader u to a listening miner v, once the event ε_3 occurs, v can receive the message/block from u. Since there is only the leader u transmitting in each slot of this stage, the interference is avoided. Then, according to the SINR rate, we get

$$\text{SINR}(u,v) \geqslant \frac{\ln t \times \dfrac{P}{R^{\alpha}}}{N(v)} \geqslant \frac{\ln t \times \dfrac{P}{R^{\alpha}}}{\ln t \times \dfrac{P}{R^{\alpha} \times \beta}} \geqslant \beta \tag{8}$$

Therefore, each non-leader miner v has the ability to receive the message/block from the leader with a constant probability of at least $1/t$ in each slot of the leader broadcast stage, and Lemma 8 is proved. ■

Now we focus on the time delay for any of a miner v decoding the message/block from the leader u.

Lemma 9 In the leader broadcast stage, each non-leader miner v can decode the message/block from the leader u in $O(\log n)$ rounds, w.h.p.

Proof Define ε_4 to be the event that the non-leader miner v decodes the block/message from the leader u in a given round, and based on Lemma 8, it can be seen that the corresponding probability $\Pr[\varepsilon_4]$ is $1/t$. Then, let $x_1(r)$ be the random variable as below:

$$x_1(r) = \begin{cases} 0, & \text{when } \varepsilon_4 \text{ does not occur in round } r; \\ 1, & \text{when } \varepsilon_4 \text{ occurs in round } r \end{cases} \tag{9}$$

Considering a time interval I_2 with k_2 rounds, let φ_3 be the expectation of the time ε_4 occurring in the interval I_2. With $\Pr[\varepsilon_4] = 1/t$, we have $\varphi_3 = k_2 \times 1/t$. Using the Chernoff bound with a constant $\delta_2 = 1/2$, we have

$$\Pr\left[\sum_{r \in I_2} x_1(r) \leqslant (1-\delta_2)\varphi_3\right] \leqslant e^{-\delta_2^2 \varphi_3/2} \tag{10}$$

Then, by setting $k_2 = \mu \log n$ and $\mu = 16t$, we get

$$\Pr\left[\sum_{r \in I_2} x_1(r) \leqslant (1-\delta_2)k_2 \times 1/t\right] \leqslant e^{-\delta_2^2 \times k_2 \times \frac{1}{t}/2} = e^{-k_2/8t} = n^{-2} \tag{11}$$

Hence, it holds that within $O(\log n)$ rounds, any non-leader miner v can decode the block/message from the leader with high probability of $1 - n^{-2}$, and it also holds for all other miners with probability of at least $P_{\text{all}} \geqslant (1-n^{-2})^n \geqslant 1 - n^{-1}$. ■

5.3 Proof of Lemma 3

Lemma 10 After $O(n)$ running rounds, the leader can collect all the verification results of other miners in spite of the slight jamming w.h.p.

Proof The specific analyses are similar to the work in Ref. [34], given that c is a constant and $t < n$. Lemma 10 can be proved via Claims 1–3. They are merely differ in the setting of γ, where $\gamma \geqslant c + \ln(R \ln n \times [1 + c\beta(n-1)] - n)$ in Lemma 10. For more details, please refer to the work in Ref. [34]. ■

Claim 1 With high probability, in any given round, there exists a unique miner assigned with the highest transmission power.

Claim 2 In any given round, the miner with the highest transmission power can successfully transmit the verified report to the leader despite the slight jamming w.h.p.

Claim 3 Within $O(n)$ rounds, the leader knows the number of approved miners and dissenting miners in the blockchain networks w.h.p.

Proof From Claims 1 and 2, we can see that the leader is able to receive a report per round despite the slight jamming and record the feedback by approval or reject. In our leader aggregation stage, once the leader has received a report from non-leader miners, it instantly answers an ack. message. The acknowledged miner will halt at the same time. As analogous to the Theorem 1 in Ref. [34], the leader aggregation stage will be completed by at most $O(n)$ rounds. As mentioned before, we use count_1 and count_2 to represent the number of approved miners and dissenting miners, respectively. Every round the leader aggregation scheme runs, the leader will update the result of count_1 or count_2. So, when the leader

aggregation stage ends, the leader knows the number of approved miners and dissenting miners in the blockchain networks. Thus, Claim 3 is proved. ■

5.4 Proof of Lemma 4

Lemma 11 After $O(\log n)$ running rounds, the leader can transmit its announcement message to all other miners despite the slight jamming w.h.p.

Proof Similar to the leader broadcast stage, the elected leader aims to transmit its announcement message to all other miners. According to Lemma 8, in each time slot of the leader announcement stage, each participating miner has a consistent probability of at least $1/t$ to receive the message/block from the leader. Then, based on Lemma 9, it takes at most $O(\log n)$ rounds to make all non-leader miners receive the message/block from the elected leader w.h.p. With the above analyses, each non-leader miner is capable of receiving the leader's announcement message and making a decision based on the results of statistical parameters count_1 and count_2. If $\text{count}_1 > \text{count}_2$, all miners take action to update the newly generated block to their local chains. Otherwise, i.e., $\text{count}_1 \leqslant \text{count}_2$, they collectively opt to reject the newly proposed block. ■

5.5 Proof of Theorem 1

We prove Theorem 1 by demonstrating that the validity, agreement, termination, and total order properties of our DJR-consensus algorithm are fulfilled within $O(n)$ rounds w.h.p. Actually, our jamming-resilient distributed consensus algorithm can be summarized as: leader election stage, leader broadcast stage, leader aggregation stage, and leader announcement stage. And the corresponding delay bounds are $O(\log n)$, $O(\log n)$, $O(n)$, and $O(\log n)$, respectively. As a result, our DJR-consensus algorithm can achieve a consensus within $O(n)$ rounds w.h.p. In other words, our jamming-resilient distributed consensus algorithm for blockchain networks satisfies the termination property. Thereafter, the validity, agreement, and total order properties will be proved in Lemmas 12–14.

Lemma 12 Our jamming-resilient distributed consensus algorithm for blockchain networks satisfies the validity property.

Proof Within a blockchain network, the objective of each miner is to verify if the transactions within the newly proposed block match historical records. Based on this verification, miners decide whether to incorporate the block into their local chains or not. In the leader election stage, there is only one leader will be elected. This leader subsequently introduces a new block and disseminates it to all non-leader miners. Due to the malicious jamming, the uncertainty of the Rayleigh fading model helps to make any of the miners receive the block. Upon receiving the block, a miner evaluates its validity based on transaction history and information within the block. After that, the number of approved miners and dissenting miners would be aggregated to the leader in the leader aggregation stage. If $\text{count}_1 > \text{count}_2$, the proposed block is integrated into their respective local blockchains as the subsequent component of the blockchain system. Conversely, the new proposed block will be discarded. ■

Lemma 13 Our jamming-resilient distributed consensus algorithm for blockchain networks satisfies the agreement property.

Proof Irrespective of the validity of the newly proposed block B_u by leader u, all miners within the blockchain network reach a unanimous consensus regarding whether to accept or discard the block B_u. From Lemma 12, we have: (1) In the event that the newly generated block B_u proves to be valid, all miners will incorporate B_u into their respective local chains. (2) Conversely, if B_u is deemed invalid, it will be collectively disregarded by all participating miners. Thus, the agreement property is well satisfied.

Lemma 14 Our jamming-resilient distributed consensus algorithm for blockchain networks satisfies the total order property.

Proof We prove Lemma 14 by contradiction. Since our blockchain consensus algorithm satisfies the agreement property, therefore the most recent block in each miner's local blockchain will be identical. Considering any distinct pair of miners, denoted as u and v, within the blockchain network, let $\{B_u^1, B_u^2, \ldots, B_u^i, \ldots, B_u^x\}$ and $\{B_v^1, B_v^2, \ldots, B_v^i, \ldots, B_v^y\}$ be their local blockchains, where B_u^i/B_v^i is the i-th block in local blockchain of u/v. Suppose that B_u^i and B_v^i are different, it is obvious that B_u^i and B_v^i would be generated by variously valid leaders and the agreement property of our blockchain consensus algorithm is violated, which resulted in contradiction. Similarly, B_u^x and B_v^y would be the same as well $(x = y)$. Therefore, all miners agree on the order of all proposed blocks, and the local blockchains of miners have the same sequence of blocks. ■

6 Experimental Result

We evaluate the empirical performance of our jamming-resilient distributed consensus algorithm for blockchain networks in this section. Concretely, our main focus is on the time spent for achieving consensus, leader election and broadcast, and data aggregation in the single-hop wireless blockchain network with the network sizes n, the slight variable λ, and the SINR parameters varying.

Parameter setting. We simulate our single-hop wireless blockchain network as a square area with the size of $L\times L$, where the side length L is set to 150 m, and we normalize the minimum distance between any pair of miners to 1. All n participating miners are initially at State A and uniformly and randomly deployed in the blockchain area. Considering the slight jamming, the ambient noise imposed by the adversary for any miner v is limited by $N(v)\leqslant \ln t\times \dfrac{P}{R^{\alpha}\times\beta}$, where constant t is the adversary-specified jamming parameter. Then, based on the jamming parameter, the value of our slight variable λ varies within {0.05, 0.06, 0.07, 0.08, 0.09, 0.10}, i.e., $\lambda = 1/t$. Notice that the transmission power for leader election and broadcast is set to $P = R^{\alpha}\cdot N\beta$, because $\dfrac{P}{d(u,v)^{\alpha}}\geqslant N\beta$, for $\forall u,v\in V$ when there are no simultaneous transmissions and jamming signals and R is defined as the maximum distance among physical devices. The number of nodes n belongs to $[1,10\,000]$, and the parameter γ satisfies $\gamma\geqslant c+\ln(R\ln n\times[1+c\beta(n-1)]-n)$. Some other SINR parameters are listed in Table 1 for reference. Besides, we conduct all our simulations on the identical platform associated with 128 GB main memory and an Intel Xeon CPU E5-2670@2.60 GHz, implemented in Python programming language and compiled by a Python compiler. Without loss of generality, we conducted the simulation over 20 runs for each presented experimental result.

Table 1 Parameter in simulation.

Parameter	Value
L	150 m
$N(v)$	$\leqslant \ln t\times \dfrac{P}{R^{\alpha}\times\beta}$, $\forall v\in V$
λ	$\{0.05,0.06,0.07,0.08,0.09,0.10\}$
P	$R^{\alpha}\times N\beta$
n	$\in[1,10\,000]$
γ	$\geqslant c+\ln(R\ln n\times[1+c\beta(n-1)]-n)$
α	$\in\{3.0,4.0\}$
β	$\in\{1.0,1.5\}$

Algorithm performance. The simulation results of our jamming-resilient distributed consensus algorithm for blockchain networks with different network sizes, SINR parameters, and slight variables are shown in Figs. 1–3, respectively.

Figure 1 depicts the average running time of our proposed DJR-consensus algorithm in single-hop wireless blockchain networks in the presence of slight jamming, in which the x-axes and y-axes represent the number of nodes and the average time for consensus, respectively. All curves in Fig. 1 show the time bounds of consensus when the network sizes n, slight variables λ, and SINR parameters α and β vary. As shown in Fig. 1, it can be seen that, (1) the average time for consensus is linearly related to the number of nodes n regardless of the slight variables λ and the SINR parameters α and β. These experimental results imply that our theoretical time complexity $O(n+\log n)$ is quasi-equivalent to $O(n)$, which achieves asymptotically optimal performance on time-bound. (2) When the value of slight variables $\lambda\in[0.05,0.10]$ and the SINR parameters $\alpha\in\{3.0,4.0\}$ and $\beta\in\{1.0,1.5\}$, all the consensus time on illustrated curves changes slightly, which indicates that our proposed algorithm is insensitive to the slight variables and SINR parameters. Take a closer look at our algorithm, we further separately show the average time for leader election and broadcast and leader aggregation.

Figure 2 illustrates the average time for LE and LB in terms of the network sizes n, slight variables λ, and SINR parameters α and β varying. From Fig. 2, we can draw conclusions that, (1) the average time for LE and LB is logarithmic in the network sizes n under different slight variables λ and various SINR parameters α and β. The simulation results corroborate our analysis that the time complexity of our LE and LB stags is both $O(\log n)$. (2) From Figs. 2a–2d, by comparing all the curves with the same n and SINR parameters α and β, we can see that when λ gets larger, it takes less time for LE and LB. Since in the leader broadcast stage, each miner can receive the message/block from the leader with a consistent probability of at least λ. This means, the greater the probability λ, the faster the completion of the leader broadcast. (3) Similar to the analysis in Fig. 1, by further comparing all the curves with the same n and λ, it can be seen that when SINR parameters α and β vary, the average time for LE and

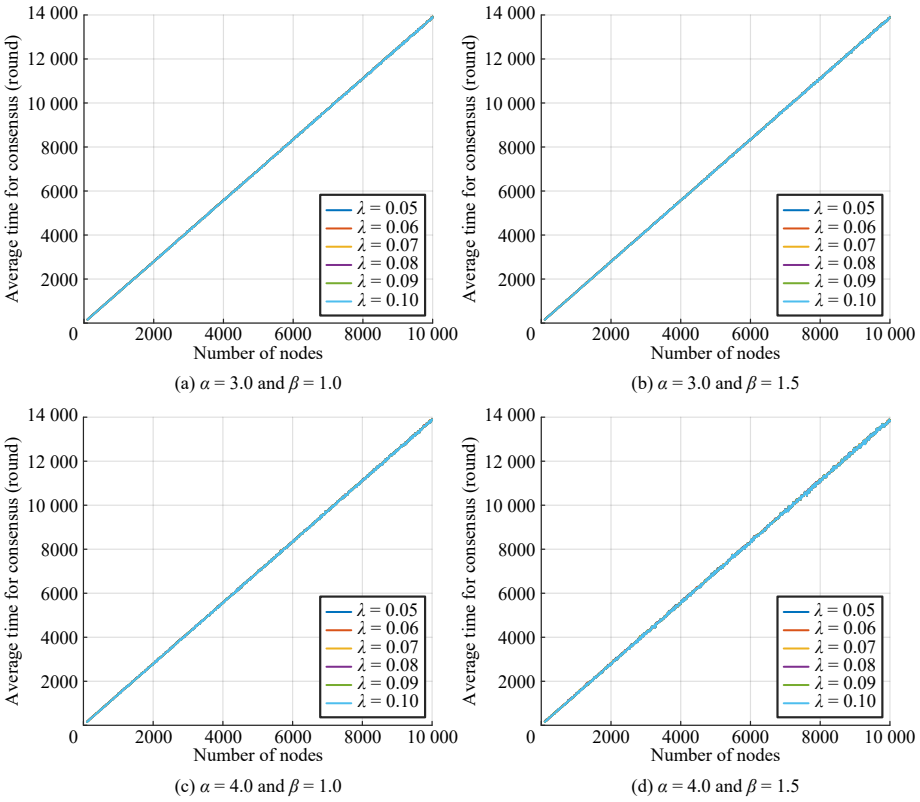


Fig. 1 Average time of our algorithm for achieving consensus when SINR parameters α and β vary.

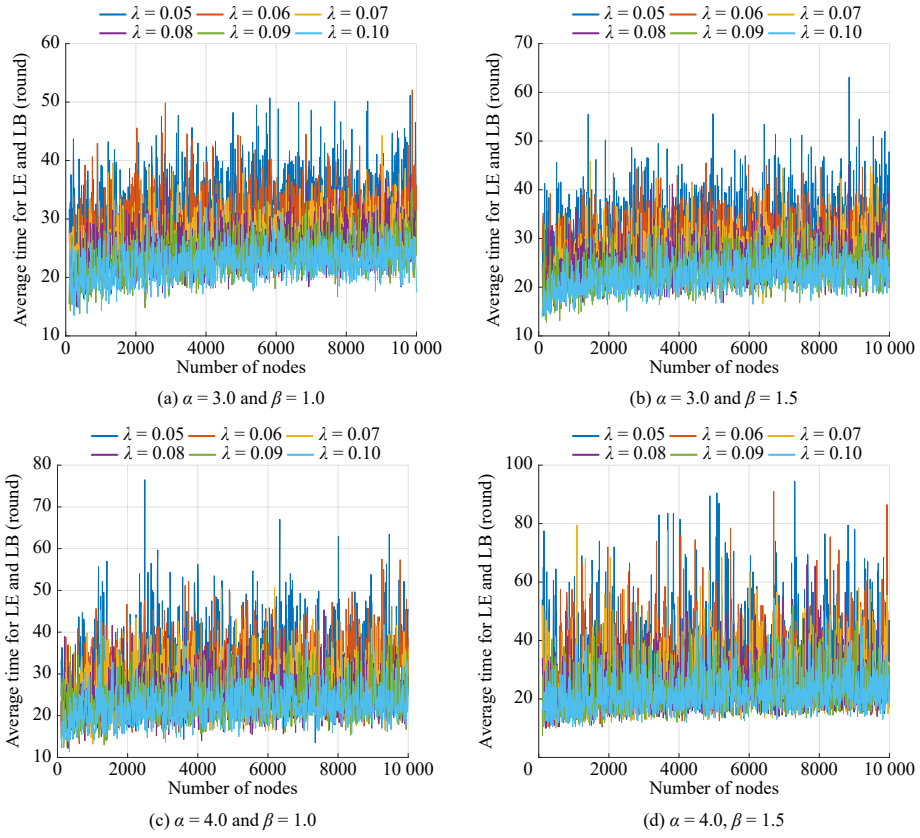


Fig. 2 Performance of our algorithm on the leader election and broadcast under various SINR parameters α and β.

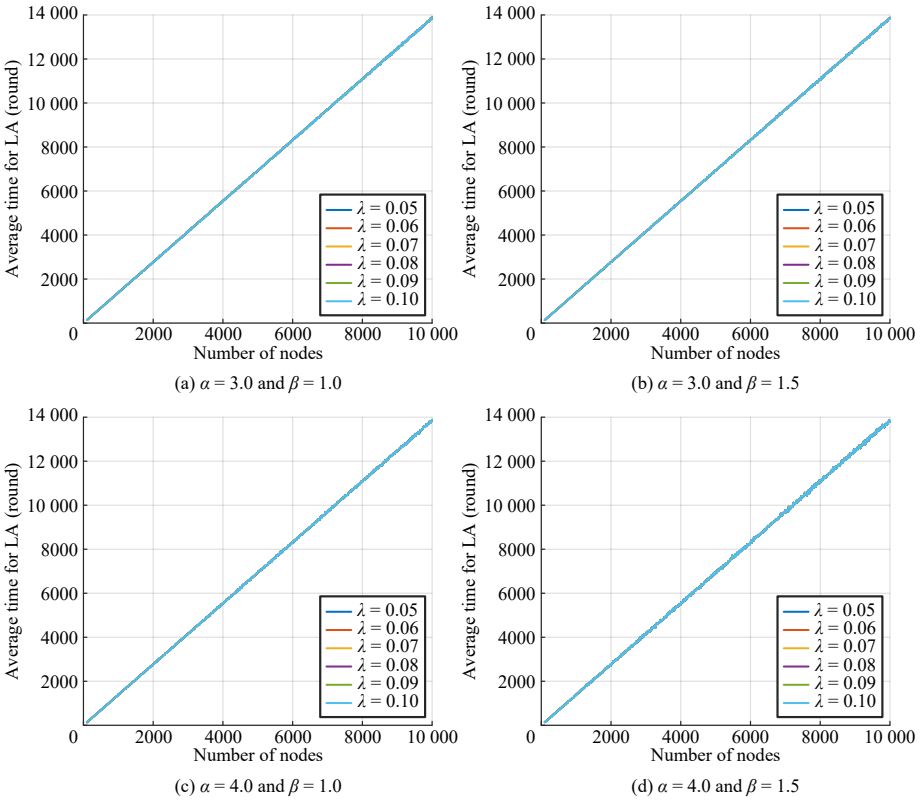


Fig. 3 Performance of our algorithm on the leader aggregation under various SINR parameters α and β.

LB changes slightly as well, which indicates that the LE and LB stages are insensitive to the SINR parameters.

Figure 3 shows the average time for LA with the network sizes n, slight variables λ, and SINR parameters α and β varying. In view of Fig. 3, one can conclude that, (1) the average running time in the leader aggregation stage is linearly increasing with the number of nodes n in spite of the slight variables λ and the SINR parameters α and β, which confirms our theoretical analysis that the delay bound of LA stage is $O(n)$. (2) Analogous to the analysis in Fig. 1, in the cases of slight variables $\lambda \in [0.05, 0.10]$ and the SINR parameters $\alpha \in \{3.0, 4.0\}$ and $\beta \in \{1.0, 1.5\}$, the LA time of the whole curves has the merely lightweight-level difference. These empirical results reveal that the LA stage is insensitive to both slight variables and SINR parameters.

Summary. In this experiment, we simulate the running time for consensus, leader election and broadcast, and leader aggregation of our DJR-consensus algorithm when the number of physical devices n, the slight variables λ, and the SINR parameters α and β vary. From the numerical results, we can see that (1) our DJR-consensus algorithm is jamming-resilient for blockchain networks; (2) all the presented experimental results well corroborate our theoretical analyses, which achieve asymptotically optimal time-bound despite the realistic slight jamming.

7 Conclusion

In this paper, we investigated the consensus algorithm for blockchain networks in the presence of slight jamming, which characterizes most of the real-life non-binary jamming phenomenon. We proposed a distributed jamming-resilient consensus algorithm to guarantee the consistency in a single-hop wireless blockchain network, which achieves asymptotically optimal performance on time complexity. The correctness and effectiveness of our algorithm have been theoretically analyzed and empirically validated. Extending our algorithm to tolerate the Byzantine failures will be our work in the future.

Appendix

A Chernoff Bound

Chernoff bound in Ref. [35] describes the tail behavior of the distribution of the sum of independent Bernoulli

experiments.

Lemma A1 (Chernoff bound) For a parameter $\gamma > 0$, let $X_1, X_2, \ldots, X_i, \ldots, X_n$ be independent or negatively associated non-negative random variables with $X_i \leqslant \gamma$. Moreover, let $X = X_1 + X_2 + \cdots + X_n$ and $\varphi = E[X]$. For $\delta > 0$, it holds that

$$\Pr[X \geqslant (1+\delta)\varphi] \leqslant \left(\frac{\mathrm{e}^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\varphi/\gamma} \tag{A1}$$

For $\delta \leqslant 1$, the bound can be upper bound by $\Pr[X \geqslant (1+\delta)\varphi] \leqslant \mathrm{e}^{-\delta^2\varphi/3\gamma}$. Furthermore, for every $\delta \in (0,1)$, we have

$$\Pr[X \leqslant (1-\delta)\varphi] \leqslant \left(\frac{\mathrm{e}^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{\varphi/\gamma} \leqslant \mathrm{e}^{-\delta^2\varphi/2\gamma} \tag{A2}$$

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (Nos. 62102232 and 62122042), Shandong Science Fund for Excellent Young Scholars (Nos. 2023HWYQ-007 and 2023HWYQ-008), and Key R&D Program of Shandong Province (No. 2022CXGC020107).

References

[1] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, https://bitcoin.org/bitcoin.pdf, 2008.

[2] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang, Untangling blockchain: A data processing view of blockchain systems, *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1366–1385, 2018.

[3] F. Tschorsch and B. Scheuermann, Bitcoin and beyond: A technical survey on decentralized digital currencies, *IEEE Commun. Surv. Tutorials*, vol. 18, no. 3, pp. 2084–2123, 2016.

[4] Q. Xu, Y. Zou, D. Yu, M. Xu, S. Shen, and F. Li, Consensus in wireless blockchain system, in *Proc. Wireless Algorithms, Systems, and Applications*, Qingdao, China, 2020, pp. 568–579.

[5] B. Huang, L. Peng, W. Zhao, and N. Chen, Workload-based randomization Byzantine fault tolerance consensus protocol, *High Confid. Comput.*, vol. 2, no. 3, p. 100070, 2022.

[6] M. Xu, S. Liu, D. Yu, X. Cheng, S. Guo, and J. Yu, CloudChain: A cloud blockchain using shared memory consensus and RDMA, *IEEE Trans. Comput.*, vol. 71, no. 12, pp. 3242–3253, 2022.

[7] Y. Zou, M. Xu, J. Yu, F. Zhao, and X. Cheng, A fast consensus for permissioned wireless blockchains, *IEEE Internet Things J.*, vol. 10, no. 14, pp. 12102–12111, 2023.

[8] L. Yang, Y. Zou, M. Xu, Y. Xu, D. Yu, and X. Cheng, Distributed consensus for blockchains in Internet-of-Things networks, *Tsinghua Science and Technology*, vol. 27, no. 5, pp. 817–831, 2022.

[9] E. Bayraktaroglu, C. King, X. Liu, G. Noubir, R. Rajaraman, and B. Thapa, Performance of IEEE 802.11 under jamming, *Mobile Netw. Appl.*, vol. 18, pp. 678–696, 2013.

[10] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, Modeling the impact of network connectivity on consensus security of proof-of-work blockchain, in *Proc. IEEE INFOCOM 2020 - IEEE Conf. Computer Communications*, Toronto, Canada, 2020, pp. 1–10.

[11] H. Xu, L. Zhang, Y. Liu, and B. Cao, RAFT based wireless blockchain networks in the presence of malicious jamming, *IEEE Wirel. Commun. Lett.*, vol. 9, no. 6, pp. 817–821, 2020.

[12] H. Pirayesh, P. Kheirkhah Sangdeh, and H. Zeng, Securing ZigBee communications against constant jamming attack using neural network, *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4957–4968, 2021.

[13] B. Awerbuch, A. Richa, and C. Scheideler, A jamming-resistant MAC protocol for single-hop wireless networks, in *Proc. Twenty-Seventh ACM Symp. on Principles of Distributed Computing*, Toronto, Canada, 2008, pp. 45–54.

[14] Z. Han, J. Ma, C. Xu, and G. Zhang, UltraJam: Ultrasonic adaptive jammer based on nonlinearity effect of microphone circuits, *High Confid. Comput.*, vol. 3, no. 3, p. 100129, 2023.

[15] A. Richa, C. Scheideler, S. Schmid, and J. Zhang, Self-stabilizing leader election for single-hop wireless networks despite jamming, in *Proc. Twelfth ACM Int. Symp. on Mobile Ad Hoc Networking and Computing*, Paris, France, 2011, pp. 1–10.

[16] S. King and S. Nadal, PPCoin: Peer-to-peer crypto-currency with proof-of-stake, https://peercoin.net/assets/paper/peercoin-paper.pdf, 2012.

[17] Y. Zou, M. Xu, D. Yu, L. Chen, S. Guo, and X. Xing, Implementation of abstract MAC layer under jamming, *Tsinghua Science and Technology*, vol. 27, no. 2, pp. 257–269, 2022.

[18] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, PBFT vs proof-of-authority: Applying the CAP theorem to permissioned blockchain, in *Proc. 2nd Italian Conference on Cyber Security, ITASEC 2018*, Milan, Italy, 2018, pp. 1–11.

[19] D. Yu, Q. S. Hua, Y. Wang, and F. C. M. Lau, An $O(\log n)$ distributed approximation algorithm for local broadcasting in unstructured wireless networks, in *Proc. IEEE 8th Int. Conf. Distributed Computing in Sensor Systems*, Hangzhou, China, 2012, pp. 132–139.

[20] A. Richa, C. Scheideler, S. Schmid, and J. Zhang, Competitive throughput in multi-hop wireless networks despite adaptive jamming, *Distrib. Comput.*, vol. 26, no. 3, pp. 159–171, 2013.

[21] A. Ogierman, A. Richa, C. Scheideler, S. Schmid, and J. Zhang, Competitive MAC under adversarial SINR, in *Proc. IEEE INFOCOM 2014 - IEEE Conf. Computer Communications*, Toronto, Canada, 2014, pp. 2751–2759.

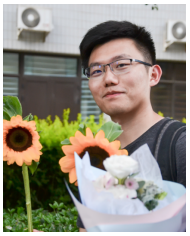
[22] Y. Zou, D. Yu, L. Wu, J. Yu, Y. Wu, Q. S. Hua, and F. C. M. Lau, Fast distributed backbone construction despite

strong adversarial jamming, in *Proc. IEEE INFOCOM 2019 - IEEE Conf. Computer Communications*, Paris, France, 2019, pp. 1027–1035.
[23] X. Lu, J. Jie, Z. Lin, L. Xiao, J. Li, and Y. Zhang, Reinforcement learning based energy efficient robot relay for unmanned aerial vehicles against smart jamming, *Sci. China Inf. Sci.*, vol. 65, no. 1, p. 112304, 2021.
[24] C. Zhao, Q. Wang, X. Liu, C. Li, and L. Shi, Reinforcement learning based a non-zero-sum game for secure transmission against smart jamming, *Digit. Signal Process.*, vol. 112, p. 103002, 2021.
[25] Y. Bai, S. Amin, X. Wang, and L. Jin, Securing signal-free intersections against strategic jamming attacks: A macroscopic approach, in *Proc. IEEE 61st Conf. Decision and Control*, Cancun, Mexico, 2022, pp. 1–17.
[26] Y. Liang, J. Ren, and T. Li, Secure and efficient OFDM system design under disguised jamming, in *Proc. Int. Conf. Computing, Networking and Communications* (*ICNC*), Big Island, HI, USA, 2020, pp. 394–399.
[27] L. Zhou, C. Zhang, Q. Zeng, X. Liu, and H. Wu, Optimal low-hit-zone frequency-hopping sequence sets with wide-gap for FHMA systems under follower jamming, *IEEE Commun. Lett.*, vol. 26, no. 5, pp. 969–973, 2022.
[28] C. Dwork, N. Lynch, and L. Stockmeyer, Consensus in the presence of partial synchrony, *J. ACM*, vol. 35, no. 2, pp. 288–323, 1988.
[29] D. Yu, L. Ning, Y. Zou, J. Yu, X. Cheng, and F. C. M. Lau, Distributed spanner construction with physical interference: Constant stretch and linear sparseness, *IEEE/ACM Trans. Networking*, vol. 25, no. 4, pp. 2138–2151, 2017.
[30] D. Yu, Y. Zhang, Y. Huang, H. Jin, J. Yu, and Q. S. Hua, Exact implementation of abstract MAC layer via carrier sensing, in *Proc. IEEE INFOCOM 2018 - IEEE Conf. Computer Communications*, Honolulu, HI, USA, 2018, pp. 1196–1204.
[31] D. Yu, Y. Zou, J. Yu, X. Cheng, Q. S. Hua, H. Jin, and F. C. M. Lau, Stable local broadcast in multihop wireless networks under SINR, *IEEE/ACM Trans. Networking*, vol. 26, no. 3, pp. 1278–1291, 2018.
[32] Y. Zou, M. Xu, H. Sheng, X. Xing, Y. Xu, and Y. Zhang, Crowd density computation and diffusion via Internet of Things, *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8111–8121, 2020.
[33] M. M. Halldórsson, S. Holzer, E. A. Markatou, and N. Lynch, Leader election in SINR model with arbitrary power control, *Theor. Comput. Sci.*, vol. 811, pp. 21–28, 2020.
[34] L. Yang, Y. Zou, D. Yu, and J. Yu, Distributed Age-of-Information optimization in edge computing for Internet of Vehicles, *J. Syst. Archit.*, vol. 144, p. 103000, 2023.
[35] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge, UK: Cambridge University Press, 1995.

Yifei Zou received the BEng degree from Wuhan University, China, in 2016, and the PhD degree from The University of Hong Kong, China, in 2020. He is currently an assistant professor at School of Computer Science and Technology, Shandong University, Qingdao, China. His research interests include wireless networks, ad hoc networks, and distributed computing.

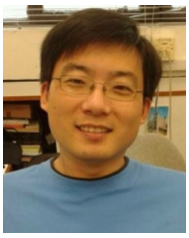
Meng Hou is currently pursuing the BS degree at School of Computer Science and Technology, Shandong University, Qingdao, China. His main research interests include distributed computing and machine learning.

Li Yang received the BS degree in information and computational science from Linyi University, China, in 2017, and the MS degree from Fuzhou University, China, in 2020. He is currently pursuing the PhD degree at School of Computer Science and Technology, Shandong University, China. His research interests include wireless networks and distributed computing.



Minghui Xu received the BS degree in physics from Beijing Normal University, Beijing, China, in 2018, and the PhD degree in computer science from The George Washington University, Washington, DC, USA, in 2021. He is currently an assistant professor at School of Computer Science and Technology, Shandong University, China. His research focuses on blockchain, distributed computing, and quantum computing.

Libing Wu received the BS and MS degrees in computer science from Central China Normal University, Wuhan, China, in 1994 and 2001, respectively, and the PhD degree from Wuhan University, Wuhan, China, in 2006. He is currently a professor at School of Cyber Science and Engineering, Wuhan University, China. He is also a researcher at Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen, China. His research interests include network security, Internet of Things, machine learning, and data security.



Dongxiao Yu received the BSc degree from Shandong University, China, in 2006, and the PhD degree from The University of Hong Kong, China, in 2014. He became an associate professor at School of Computer Science and Technology, Huazhong University of Science and Technology, China, in 2016. He is currently a professor at School of Computer Science and Technology, Shandong University, China. His research interests include wireless networks, distributed computing, and graph algorithms.

Xiuzhen Cheng received the MS and PhD degrees in computer science from University of Minnesota, Twin Cities, USA, in 2000 and 2002, respectively. She was a faculty member at Department of Computer Science, The George Washington University, USA, from 2002–2020. Currently she is a professor of computer science at Shandong University, Qingdao, China. Her research focuses on blockchain computing, security and privacy, and Internet of Things. She is a fellow of IEEE.