Trust-Preserving Mechanism for Blockchain Assisted Mobile Crowdsensing

Long Zhang, Gang Feng, *Senior Member, IEEE,* Shuang Qin, *Member, IEEE,* Xiaoqian Li, Yao Sun, *Senior Member, IEEE,* and Bin Cao, *Senior Member, IEEE*

***Abstract*—Blockchain is envisioned as one of the promising technologies to address trust concern brought by mobile crowdsensing (MCS), due to its auditability, immutability and decentralization. Nevertheless, blockchain cannot fundamentally guarantee that the valuable sensed data outside the chain can enter the chain, although data integrity and consistency can be ensured once it is confirmed inside the chain. In addition, simply applying blockchain in MCS while ignoring possible abnormal saboteurs hidden in numerous devices may mislead the normal operation of blockchain, resulting in untrustworthy interactions. Consequently, it is highly desirable to build a trust-preserving mechanism (TPM) to fully enjoy the benefits of using blockchain in MCS. To this end, we first resort to a probabilistic trust assessment inferred from the interaction outcomes in blockchain, to incentivize participants to maintain the trustworthiness of interactions. By inferring trust to aid decision-making, trust decision is further made, including leader election and transaction data generation, to filter untrusted nodes from participating in blockchain process. Finally, extensive simulations are conducted to validate the effectiveness and efficiency of TPM, and improve the performance in terms of contribution rate, consensus accuracy and system stability.**

***Index Terms*—Blockchain, mobile crowdsensing, trust-preserving mechanism, trust decision.**

I. Introduction

LEVERAGING ubiquitous devices for data sensing, mobile crowdsensing (MCS) [1], [2] provides data analysis and computation to customers with common interests under centralized coordination, facilitating many novel application paradigms. In MCS, intelligent devices monitor the surrounding features at the edge of networks, upload the sensed data to the MCS server for intelligent processing [3], [4], and then get rewards based on the evaluation results. Due to the large amount of data generated from numerous devices in the current emerging applications, the existing centralized authority and client-server mode of MCS system hinder the bidirectional judgment of trust relationship between MCS servers and terminal devices, making them vulnerable to attack and deception. On the one hand, the data generated by different MCS participants differ in terms of trustworthiness which involves data quality and user reliability. On the other hand, most of data collected in MCS is related to the privacy of participants.

Recently, the emerging blockchain technology provides a decentralized paradigm for trustworthy data sharing in MCS [5]. Before interacting with other participants, the sensed data in blockchain-assisted MCS is hashed and encapsulated into a basic verifiable data structure, called the transaction. By deploying appropriate consensus protocols and cryptographic algorithms [6], [7], individual participants can independently and securely interact and verify transactions without the centralized authority of MCS. Meanwhile, participants in blockchain can create, transmit and access transactions anonymously, thus greatly eliminating the possibility of exposing or stealing real user identity due to the evil MCS. Moreover, only transactions verified by the majority of participants can be authorized to be recorded in a distributed ledger, which brings strong trust endorsement to the sensed data and interactions inside the blockchain.

However, an obvious vulnerability of blockchain-assisted MCS is that the data originates in untrusted participants and interacts through unreliable physical facilities and networks. While blockchain can ensure the integrity and traceability of sensed data once a transaction is confirmed inside the blockchain, it still lacks an effective trust assessment mechanism to determine whether the data outside the blockchain is allowed inside the blockchain [8]. Obviously, the truthfulness of the data outside the blockchain cannot be accurately verified, which is beyond the scope of current blockchain capabilities [9]. Meanwhile, the abnormal behavior of participants inevitably affects the normal interaction process inside the blockchain. For example, malicious users may launch byzantine faults to interfere with others [10], and lazy users may not respond to others in time, thus impeding the consensus process. This implies that in the absence of an effective trust assessment mechanism, the trustworthiness of any data and interactions cannot be guaranteed even if they are confirmed by the blockchain.

Consequently, there will be a need for a trust-preserving mechanism (TPM) to bridge the trust gap inside and outside the blockchain by forming an assessment of interaction information. On the one hand, the founding rationale of blockchain is that mutually untrusted participants interact based on crowd consensus and verifiable transactions, resulting in truthful and

L. Zhang, G. Feng, S. Qin, and X. Li are with the National Key Laboratory of Wireless Communications, University of Electronic Science and Technology of China, Chengdu 611731, China, and also with the Yangtze Delta Region Institute (Huzhou), University of Electronic Science and Technology of China, Huzhou 313001, P. R. China. Y. Sun is with the James Watt School of Engineering, the University of Glasgow, Glasgow, UK. B. Cao is with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China, and is also with the Zhejiang Lab, Hangzhou, 311121, China. Email: caobin@bupt.edu.cn. (Corresponding author: Bin Cao.)

verifiable interaction outcomes. On the other hand, trust occurs naturally in various interaction services in blockchain-assisted MCS, which is a measurable belief and can be assessed based on past interaction outcomes [11]. Therefore, incorporating TPM into the blockchain is a way not only to create an unforgeable chain of data but also to establish trust inside and outside the blockchain.

Whereas various trust assessment mechanisms have been proposed [12]–[15], there is still a lack of an effective assessment approach for blockchain-assisted MCS, which is crucial to the trustworthiness of the system and the validity of the data. In general, the existing trust assessment approaches infer a trust score according to the past interaction outcomes between participants, which can be designed based on evidence theory, subjective logic, fuzzy logic, machine learning, etc [16]. However, the rightness of interaction information is hard to verify in conventional trust assessment approaches due to the presence of lazy and malicious participants, which directly affects the accuracy of the inter-node trust scoring. The advent of blockchain, with its verifiable and immutable features, makes up for the lack of trust. These observations inspire us to exploit blockchain to design a trust assessment approach and facilitate the transfer of valuable data from outside the chain to inside the chain through transactions. Our main contributions in this paper are summarized as follows:

1) We propose to incorporate TPM into a blockchain-assisted MCS framework, which relies on a chained database to track and secure transactions from MSC, while TPM is designed to filter untrusted participants so that valuable data outside the blockchain can be admitted by the blockchain. In TPM, a trust assessment process is used to measure whether the participants in the system are trustworthy.
2) We develop a probabilistic multi-class trust assessment model, and employ the multinomial distribution to map interaction outcomes verified by blockchain into ternary trust scores, i.e., belief, disbelief, and uncertainty. Meanwhile, we consider the adverse impact of deficient interaction information on inferring trust scores and thus derive a knowledge defect to compensate for the bias introduced by insufficient interaction to trust assessment.
3) To use inferred trust to aid decision-making, we resort to a trusted-leader election to give highly-trusted nodes more opportunities to be elected as leaders, thus starting the consensus process correctly. Afterward, we analyze the maximum transaction throughput considering system stability and obtain the optimal transaction data size to reduce network and storage overhead.
4) We provide numerical results to validate the performance of the proposed TPM in blockchain-assisted MCS. Compared with benchmarks with and without trust assessment, the TPM can achieve significant improvement in contribution rate, consensus accuracy, and system stability.

The remainder of the paper is organized as follows. Section II reviews the existing solutions for trust, blockchain and MCS. In Section III we illustrate a TPM framework in blockchain-assisted MCS. Section IV details the trust assessment methodology, which is followed by the trust decision in Section V. Section VI further uses inferred trust to aid decision-making. Extensive experiments are conducted in Section VII. Finally, Section VIII concludes the paper.

II. Related Work

We survey the related work of TPM and blockchain in MCS, and highlight the motivation and novelty of our work.

A. Blockchain-enabled MCS

In order to tackle the challenges of trust, security and privacy caused by the centralized MCS architecture, blockchain, as a backbone decentralized framework in data management systems, has been introduced into MCS, with the aim to realize trusted transactions among different participants. Recently, some researchers have studied how to implement task publish, worker selection and other issues in MCS using blockchain [17]–[20]. In [17], Zou *et al.* apply Ethereum-based public blockchain to replace centralized MCS to protect location privacy and avoid repudiation and tempering of information. In addition to the publishers and workers in traditional MCS, the verifiers act as miners in blockchain-based MCS to execute proof of work (PoW) consensus. For the purpose of safely enforcing transactions, the smart contracts in a blockchain are used to automatically execute digital contracts, which are triggered when predetermined contract terms are met. With Ethereum-based public blockchain, Li *et al.* in [18] design a series of smart contracts to perform crowdsourcing tasks, including user register, task release, task reception, etc.

To promote data trading between producers and consumers without brokers, An *et al.* in [19] focus on the crowdsensed data trading using blockchain and smart contracts. To guarantee the trustworthiness of data sensing and trading, the authors regard the blockchain as a reverse auctioneer to determine sellers and bidding strategy securely and reliably. In the field of intelligent transportation, Ning *et al.* in [20] address the safety, latency and utility challenges brought by blockchain-enabled MCS and formulate a multi-objective optimization problem by jointly considering these challenges.

Overall, the aforementioned research provides useful references for understanding blockchain-assisted MCS to improve trust, security and privacy from a system view. Nevertheless, there is little research available on how to fill the trust gap inside and outside the blockchain. As noted, if the data generated by untruthful nodes is added to the blockchain, the rightness of the ledger will be degraded. Conversely, the untrusted ledger can further mislead the interaction behavior of nodes and destroy the normal operation of the blockchain. Consequently, establishing trust in MCS is a critical problem to be solved.

B. Integration of Trust and Blockchain

To address the trust challenge caused by untruthful data and abnormal interactions, trust management mechanisms can be exploited to supervise and motivate distributed nodes. In

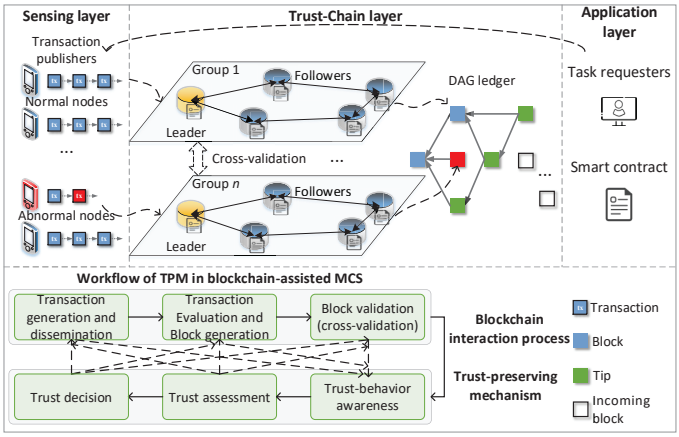


Fig. 1. The TPM framework in blockchain-assisted MCS, where the red dashed box represents the malicious transactions published by abnormal nodes.

leader-based blockchain, a leader (miner) should be elected to initiate the consensus process. In [12], Mohammed *et al.* use the payment in the form of reputation to incentivize normal nodes and punish misbehaving nodes, where the reputation is calculated by Vickrey, Clarke, and Groves (VCG) model. To resist selfish edge attacks and faked service attacks, Xiao *et al.* in [13] propose a blockchain-based trust mechanism to assess service reputation using computational results from edge devices. In PoW, the edge device with the highest service reputation has the highest opportunity to win the miner election. An *et al.* in [14] present a quality control mechanism for crowdsensing applications through a credit-based consensus node selection, where the credit value is incremented or decremented with the occurrence of normal or abnormal behavior.

In addition, trusted computing based on cryptographic technology has been developed to improve the security and trustworthiness of computer platforms [21]. By embedding trusted platform modules on the chip, transactions can be securely executed in an off-chain computing environment, and then the results can be returned to the blockchain [22].

Although some progress has been made in trust-preserving blockchain, most of the work is based on oversimplified and subjective trust assessment, without inferring trust from the actual interaction behavior of blockchain. Furthermore, most studies integrate MCS into a blockchain with single chain ledger structure and PoW consensus protocol. Unfortunately, such blockchain yields limited throughput, long confirmation latency and high resource consumption, which is not appropriate for resource-constrained mobile devices. To this end, Chatzopoulos *et al.* in [23] introduce a directed acyclic graph (DAG)-based blockchain, which allows concurrent transactions to improve cost-efficiency and scalability. By measuring the reputation of users with the earned fees, the probability of a user being selected to maintain DAG is proportional to the reputation.

C. The Novelty of the Paper

Owing to the aforementioned considerations, we aim to design an effective TPM under the framework of DAG-based blockchain, so that valuable sensed data outside the blockchain can be authorized to enter the blockchain. Different from the aforementioned research, we provide a more holistic view of modeling trust in blockchain by constructing a ternary trust assessment. It features the use of multinomial distribution to map the interaction outcomes verified by blockchain into multi-class trust scores including belief, disbelief and uncertainty. Meanwhile, we also address the problem of knowledge defect caused by deficient interaction quantity, which is crucial for forming accurate and fair trust assessment.

III. Framework of TPM in Blockchain-Assisted MCS

Depending on the specific service requirements, interaction nodes can be assigned to one or more roles, including transaction requesters, transaction publishers and transaction verifiers [17]. Specifically, transaction requesters are the initiators to publish MCS tasks with requirements. Motivated by rewards, transaction publishers are recruited to generate transactions with sensed data, ID, Hash, etc., which are then submitted to the blockchain. Through consensus protocols, transaction verifiers verify the validity and integrity of received transactions. For leader-based consensus protocols, transaction verifiers regularly elect a leader to aggregate the valid and eligible transactions into a candidate block and send it to all other verifiers called followers.

To reduce consensus delay and transaction throughput, splitting the large-scale networks into multiple groups is an effective method while improving system scalability [24]. Hence, we consider that the interaction nodes in the system constitute $\mathcal{G}$ non-overlapping and non-empty groups. For the sake of presentation, the g-th group is indexed by $\mathcal{G}_g$, which can be formed based on distances, social ties and interests [24], [25]. In this paper, we can statically define groups according to the regions where nodes are located.

In Fig. 1, the framework of TPM in blockchain-assisted MCS is built on a hierarchy of IoT systems, composed of sensing layer, trust-Chain layer and application layer from left to right. At the bottom of Fig. 1, we show the workflow of TPM in blockchain-assisted MCS. Starting from the transaction generation and dissemination at sensing layer, the transactions are evaluated and verified at trust-chain layer, and eligible ones are allowed to be included in the ledger.

A. Sensing Layer

Interaction nodes at sensing layer are responsible for realizing *transaction generation and dissemination*. At this layer, the normal nodes, such as heterogeneous smart devices, act as transaction publishers to honestly participate in the tasks. Meanwhile, abnormal nodes can deliberately publish transactions that cause failures to occur in blockchain or affect the service quality in MCS. In transaction generation stage, the sensed data with some additional information will be filled into a transaction $Tx = (Hash, ID, Trust, Data, Timestamp)$,

where *Hash* is the hash digest of *Tx*, *ID* is the assigned identity, *Trust* is the trust score of the transaction publisher, *Data* is the transaction data, and *Timestamp* is the update time of transaction generation. In the subsequent transaction dissemination stage, the signed transaction is forwarded to the associated leader node for further verification.

B. Trust-Chain Layer

The trust-chain layer is served by verifiers composed of leader and follower nodes, in which leader nodes are in charge of verifying the rightness and integrity of transactions received from sensing layer, and aggregating valid and eligible transactions into candidate blocks, and the follower nodes are in charge of checking the candidate blocks. The functions involved include *trust assessment and decision*, *transaction evaluation and block generation* and *block validation*.

For *trust assessment and decision*, each node infers the trust score based on the interaction information from blockchain and makes trusted decisions according to TPM. Please refer to the next section for details of TPM. For *Transaction evaluation and block generation*, according to the votes of follower nodes in each group, a trusted leader is elected to fairly evaluate the transaction. Through transaction evaluation, the leader is empowered to incorporate the eligible transactions into a block, so as to maintain the integrity and trustworthiness of transaction data inside the blockchain. For *block validation*, the verifiers in a group validate received candidate blocks, including hash digest, trust score, signature, timestamp, etc. The verified candidate block will be added to the ledger and wait to be confirmed as a block until the preset policy is met.

In this paper, we resort to a directed acyclic graph (DAG)-based distributed ledger to record transaction data. For conventional single-chain data structure (e.g., bitcoin), it is impermissible for concurrent transactions to form forks at the same position in the chain. Instead, the DAG-based distributed ledger allows concurrent transactions to be processed simultaneously, resulting in a forking structure. As a result, DAG-based design significantly improves transaction throughput and scalability, but is inevitably vulnerable to attacks with intensive-computing power.

In DAG, the new incoming blocks need to select some published and yet unapproved candidate blocks (called tips) for approvals and store their hashes in own blocks, yielding a forking-chain structure. After that, the incoming block also becomes a tip waiting for subsequent approvals. To measure the workload of issuing a block, each block is attached with a weight value. With the continuous arrival of incoming blocks, a tip is eventually regarded as a block when its cumulative weight reaches a predefined threshold. Note that the cumulative weight of a block is calculated as the weight of itself plus the weights of other blocks that directly and indirectly approve it. In TPM in blockchain-assisted MCS, the incoming block prefers to approve top-k tips with the highest trust score in DAG. In Fig. 1, we depict the DAG ledger in the presence of abnormal nodes, where an abnormal tip (depicted by red box) cannot be approved by incoming blocks with the assistance of TPM, and thus it will be eventually isolated.

C. Application Layer

The application layer uses application programming interfaces (APIs) to allow interaction nodes to access. With the requirements of APIs, interaction nodes can be assigned to the appropriate group and smart contracts.

IV. Trust Assessment of TPM in Blockchain-assisted MCS

According to the definition of ITU-T recommendation [11], *trust can be viewed as the capacity and belief of one entity to predict another entity's future behavior based on the interactions accumulated from history*. In this section, we aim to infer the trust score between nodes using the interaction information from transaction generation to block validation.

After an interaction, the corresponding interaction outcome (typically positive/negative) can be used as the first-hand observation to infer a trust score. For example, to mitigate counterparty risk, Bitcoin-otc marketplace infers the trust score of a user by aggregating the number of positive and negative ratings[1]. Considering the uncertain interaction outcomes, we express the general trust score as $T = (T_b, T_d, T_u)$ in $[0, 1]$, where T_b is the belief degree inferred from normal interaction outcome b, T_d is the disbelief degree inferred from abnormal interaction outcome d and T_u is the uncertainty degree inferred from uncertain interaction outcome u [26].

A. Trust Assessment Process

The designed TPM operates in two phases: trust behavior-awareness and trust assessment.

1) Trust behavior-awareness: The trust behavior-awareness identifies a series of interaction behavior used to form trust assessment. Owing to the auditability and verifiability of blockchain, *trust behavior-awareness* monitors and records the interaction outcomes b, d and u throughout *transaction generation and dissemination*, *transaction evaluation and block generation* and *block validation* processes.

- The interaction behavior results in outcome b if the transaction generated by a node is successfully added to DAG ledger through a leader, i.e, the transaction is confirmed from *trust generation* to *block validation*. The interactions that successfully pass transaction evaluation, block validation and trust assessment can be detected as $Hash(\textit{PreHash}, \textit{Merkle}, \textit{Nonce}) \leq \textit{Target}$, where $Hash()$ is the hash operation, *PreHash* is the hash value of the previous block, *Merkle* is the root of the Merkle tree containing eligible transactions in the block, and *Target* is a hash value that a valid block must be smaller than or equal to.
- The interaction behavior results in outcome d if any of the functions from *trust generation* to *block validation* fails. The abnormal interactions are typically caused by Byzantine failures, i.e., some of the nodes fail in responding or interact maliciously. Once abnormal interactions are detected ($Hash(\textit{PreHash}, \textit{Merkle}, \textit{Nonce}) > \textit{Target}$), the resultant interaction outcome d will be recorded.

[1] https://bitcoin-otc.com/viewratings.php

- The interaction behavior results in outcome u if the generated transactions and blocks cannot be included in DAG ledge due to ineligible and invalid reasons. In addition, we consider that some leaders cannot complete an interaction during their term due to lazy behavior, resource constraints, etc. As a result, the resultant interaction outcome u between nodes will be recorded.

In the following, we use $\mathrm{a}_{i,j} = (a_{b,i,j}, a_{d,i,j}, a_{u,i,j})$ to denote the number of direct interaction outcome from node i to node j, where $a_{b,i,j}$, $a_{d,i,j}$ and $a_{u,i,j}$ represent the number of interaction outcomes b, d and u from node i to node j, respectively. Accordingly, the total interaction outcome from node i to node j can be expressed as $a_{i,j} = \sum_{o\in\{b,d,u\}} a_{o,i,j}$. In addition, the interaction outcome of node i can be expressed as $\mathrm{a}_i = (a_{b,i}, a_{d,i}, a_{u,i})$, where $a_{b,i}$, $a_{d,i}$ and $a_{u,i}$ represent the number of interaction outcomes b, d and u of node i, respectively. The total interaction outcome of node i can be calculated as $a_i = \sum_{o\in\{b,d,u\}} a_{o,i}$.

2) Trust assessment: Trust assessment aims to infer the trust score between a pair of interaction nodes from historical interaction information. For ease of representation, we use $\mathrm{a} = (a_b, a_d, a_u)$ as a specific example to represent $\mathrm{a}_{i,j}$ or a_i, where a_b, a_d and a_u represent the number of interaction outcomes b, d and u, respectively. In addition, in order to measure the impact of different interaction outcomes on trust assessment, we use the weight $\tau = (\tau_b, \tau_d, \tau_u)$ to indicate the importance of interaction outcomes b, d and u, respectively. To punish abnormal behavior and prevent the proportion of normal outcomes b from increasing rapidly, τ_d and τ_u are usually greater than τ_b.

To form a trust assessment on different interaction outcomes, the Dirichlet distribution can be used to map the multi-class interaction information into a probability distribution [27]. Given the multi-class interaction outcome $\mathrm{a} = (a_b, a_d, a_u)$, we can construct a ternary trust score $T = (T_b, T_d, T_u)$, in which each element in T is regarded as an expectation of the parameter in a under the Dirichlet distribution. Accordingly, the probability distribution of each possible outcome b, d and u can be regarded as a multinomial distribution $\Theta = (\Theta_b, \Theta_d, \Theta_u)$, where Θ_b, Θ_d and Θ_u are unknown prior probability of each possible outcome b, d and u, respectively, and $\sum_{o\in\{b,d,u\}} \Theta_o = 1$. According to the Bayesian theory, the Dirichlet distribution is the conjugate prior of multinomial distribution $\Theta = (\Theta_b, \Theta_d, \Theta_u)$. Thus, we can express the probability density function (PDF) of the Dirichlet distribution [28] as

$$Dir\left(\Theta|\mathrm{a}\right) = \frac{\Gamma\left(\sum_{o\in\{b,d,u\}} \tau_o a_o\right)}{\prod_{o\in\{b,d,u\}} \Gamma\left(\tau_o a_o\right)} \prod_{o\in\{b,d,u\}} \Theta_o^{\tau_o a_o - 1}, \quad (1)$$

where $\Gamma(\cdot)$ is Gamma function. In addition, the expectation of Θ is $E_{Dir(\Theta|\mathrm{a})}\left(\Theta_o\right) = \frac{\tau_o a_o}{\sum_{o\in\{b,d,u\}} \tau_o a_o}$.

Furthermore, we use $\mathrm{a}' = (a'_b, a'_d, a'_u)$ to denote the possible interaction outcome for the subsequent interaction, where a'_b, a'_d and a'_u represent the possible outcomes of belief b, disbelief d and uncertainty d, respectively. As the conjugate prior of multinomial distribution, the fact is if the prior distribution of multinomial follows the Dirichlet distribution, so does the posterior distribution. Therefore, for the o-th possible outcome, its weighted expectation under the posterior distribution can be expressed as

$$\begin{aligned} E_{D(\Theta_o|\mathrm{a}')}\left(\Theta_o\right) &= \int_{\Theta_o} \Theta_o D\left(\Theta_o|\mathrm{a}'\right) d\Theta_o \\ &= \frac{\tau_o a_o + \tau_o a'_o}{\sum_{o\in\{b,d,u\}} \left(\tau_o a_o + \tau_o a'_o\right)}. \end{aligned} \quad (2)$$

According to the expectation of the Dirichlet distribution, the trust assessment T_o can be represented as $T_o = \frac{\tau_o a_o + \tau_o a'_o}{\sum_{o\in\{b,d,u\}} (\tau_o a_o + \tau_o a'_o)}$ ($o \in \{b, d, u\}$). However, using deficient observations to assess trust can easily lead to knowledge defect, resulting in inaccurate and unfair trust score. As a result, a small number of interactions may yield a high trust score. As such, malicious nodes can easily improve the belief degree T_b by increasing the number of normal interactions over a period of time. To remedy this defect, we consider the impact of imperfect interaction knowledge on trust assessment by exploiting the definition of certainty in [15]. For the multinomial distribution considered in this paper, we further derive a knowledge defect $c\left(\mathrm{a}'\right)$ as below.

To obtain $c\left(\mathrm{a}'\right)$, we first express the conditional PDF of Θ given a' as $f\left(\Theta|\mathrm{a}'\right)$. Due to the mean value $\left(\int_0^1 f\left(\Theta|\mathrm{a}'\right) d\Theta\right) / (1-0) = 1$, the idea of computing $c\left(\mathrm{a}'\right)$ is to use mean absolute deviation (MAD) to count the number of increases and decreases from mean value 1 [15]. Given the observed interaction space $\mathrm{a}' = (a'_b, a'_d, a'_u)$ and corresponding probability $\Theta = (\Theta_b, \Theta_d, \Theta_u)$, $c\left(\mathrm{a}'\right)$ can be calculated based on MAD, expressed by $c\left(\mathrm{a}'\right) = \frac{1}{2}\iiint_0^1 \left|f\left(\Theta|\mathrm{a}'\right) - 1\right| d\Theta$, where $\frac{1}{2}$ is a scaling factor to eliminate double counting. To obtain $f\left(\Theta|\mathrm{a}'\right)$, we should calculate PDF $f\left(\Theta\right)$ and probability distribution $Prob\left(\mathrm{a}'|\Theta\right)$. In fact, $f\left(\Theta\right)$ follows Dirichlet distribution $Dir\left(\Theta|\mathrm{a}'\right)$ and $Prob\left(\mathrm{a}'|\Theta\right)$ is multinomial distribution, i.e., $Prob\left(\mathrm{a}'|\Theta\right) = \binom{a'}{a'_b, a'_d, a'_u} \prod_{o\in\{b,d,u\}} \Theta_o^{a'_o}$. Substituting $D\left(\Theta|\tilde{\mathrm{a}}_{i,j}\right)$ and $Prob\left(\mathrm{a}'|\Theta\right)$ into $c\left(\mathrm{a}'\right)$, we have

$$\begin{aligned} c\left(\mathrm{a}'\right) &= \frac{1}{2}\iiint_0^1 \left|f\left(\Theta|\mathrm{a}'\right) - 1\right| d\Theta \\ &= \frac{1}{2}\iiint_0^1 \left|\frac{Prob\left(\mathrm{a}'|\Theta\right) f\left(\Theta\right)}{\iiint_0^1 Prob\left(\mathrm{a}'|\Theta\right) f\left(\Theta\right) d\Theta} - 1\right| d\Theta, \\ &= \frac{1}{2}\iiint_0^1 \left|\frac{\prod_{o\in\{b,d,u\}} \Theta_o^{a_o + a'_o - 1}}{\iiint_0^1 \prod_{o\in\{b,d,u\}} \Theta_o^{a_o + a'_o - 1} d\Theta} - 1\right| d\Theta. \end{aligned} \quad (3)$$

Note that the calculation of knowledge defect usually consumes a relatively large amount of time due to the triple integral involved in (3). In practice, an effective solution is to pre-calculate the value of knowledge defect and save it as a lookup table to avoid repeated calculation. Based on $E_{D(\Theta_o|\mathrm{a}')}\left(\Theta_o\right) = \frac{\tau_o a_o + \tau_o a'_o}{\sum_{o\in\{b,d,u\}} (\tau_o a_o + \tau_o a'_o)}$ in (2) and $c\left(\mathrm{a}'\right)$ in (3), the general trust score can be calculated as $T = (T_b, T_d, T_u)$, where $T_b = c\left(\mathrm{a}'\right) \frac{\tau_b a_b + \tau_b a'_b}{\sum_{o\in\{b,d,u\}} (\tau_o a_o + \tau_o a'_o)}$, $T_d = c\left(\mathrm{a}'\right) \frac{\tau_d a_d + \tau_d a'_d}{\sum_{o\in\{b,d,u\}} (\tau_o a_o + \tau_o a'_o)}$ and $T_u = 1 - T_b - T_d$.

$$DT_{i,j} = \Big(\underbrace{c(\mathrm{a}'_{i,j}) \frac{\tau_b a_{b,i,j} + \tau_b a'_{b,i,j}}{\sum_{o\in\{b,d,u\}} (\tau_o a_{o,i,j} + \tau_o a'_{o,i,j})}}_{DT_{b,i,j}}, \underbrace{c(\mathrm{a}'_{i,j}) \frac{\tau_d a_{d,i,j} + \tau_d a'_{d,i,j}}{\sum_{o\in\{b,d,u\}} (\tau_o a_{o,i,j} + \tau_o a'_{o,i,j})}}_{DT_{d,i,j}}, \underbrace{1 - DT_{b,i,j} - DT_{d,i,j}}_{DT_{u,i,j}} \Big). \quad (4)$$

B. Direct and Indirect Trust Assessment

Typically, trust assessment between a pair of nodes yields direct and indirect trust scores. If node i has ever interacted with node j, the direct trust score can be obtained. In contrast, it is reasonable to infer that node i may trust node j to some degree according to the recommendation of another node, say x, yielding the indirect trust score.

First, substituting the direct interaction outcome $\mathrm{a}_{i,j} = (a_{b,i,j}, a_{d,i,j}, a_{u,i,j})$ into the general trust score T and the knowledge defect $c(\mathrm{a}')$, the direct trust score is given by $DT_{i,j} = (DT_{b,i,j}, DT_{d,i,j}, DT_{u,i,j})$ with the corresponding $c(\mathrm{a}'_{i,j})$. As a result, the direct trust score $DT_{i,j}$ can be expressed in (4).

Next, the indirect trust score $IT_{i \xrightarrow{x} j}$ from node i to node j can be calculated based on the recommendation of common neighbors. Here we use $i \xrightarrow{x} j$ $(x \in \Psi_{i,j})$ to denote an interaction path from node i to node j through neighbor node x, $\Psi_{i,j} = N(i) \bigcap N(j)$ is the set of common neighbors of nodes i and j, $N(i)$ and $N(j)$ are the neighbors of nodes i and j respectively [29]. For fairness and motivation, the common relationship between nodes can be considered in indirect interactions. For example, the nodes in a community of common interest tend to contribute more than those of an irrelevant community. Let $\omega_{i,j}$ be the common relationship weight between nodes i and j, typically reflecting common-distance, common-neighbors, etc., given by

$$\omega_{i,j} = \begin{cases} \mathit{Dis}\,(i,j) \,/\, \max_{i',j' \in \mathcal{G}_g} \mathit{Dis}\,(i',j'), & \text{co-distance,} \\ \Psi, & \text{co-neighbors,} \end{cases} \quad (5)$$

where $\mathit{Dis}(i,j)$ represents the distance between nodes i and j, and Ψ represents the number of common neighbors. By associating $\omega_{i,j}$, the indirect trust score can be calculated as

$$\begin{aligned} IT_{i \xrightarrow{x} j} &= \left(IT_{b,i \xrightarrow{x} j}, IT_{d,i \xrightarrow{x} j}, IT_{u,i \xrightarrow{x} j} \right), \\ &= \begin{cases} DT_{i,x}, & \text{if } \ i == v, \\ DT_{x,j}, & \text{if } \ j == v, \end{cases} \end{aligned} \quad (6)$$

where $v \in \arg\min_{\{i,j\}} (\bar{\omega}_{i,x} DT_{b,i,x}, \bar{\omega}_{x,j} DT_{b,x,j})$. $\bar{\omega}_{i,x}$ and $\bar{\omega}_{x,j}$ are the normalized common relationship weights respectively.

So far we have obtained indirect trust score $IT_{i \xrightarrow{x} j}$ from multiple interaction paths $i \xrightarrow{x} j$. Next, the fusion mechanism is needed to combine multiple indirect interaction paths. To achieve this, Dempster-Shafer's rule can be used to effectively tackle the combination problem of multiple indirect trust [30]. Formally, the aggregated indirect trust score is given by $IT_{i,j} = (IT_{b,i,j}, IT_{d,i,j}, IT_{u,i,j})$. Following the Dempster-Shafer's rule, $IT_{i,j}$ can be expressed as

$$IT_{i,j} = \begin{cases} IT_{b,i,j} = \frac{\sum_{l_1 \cap l_2 \cdots \cap l_\Psi = \{b\}} IT_{l_1,i \xrightarrow{1} j} \cdots IT_{l_\Psi,i \xrightarrow{\Psi} j}}{1 - \sum_{l_1 \cap l_2 \cdots \cap l_\Psi = \emptyset} IT_{l_1,i \xrightarrow{1} j} \cdots IT_{l_\Psi,i \xrightarrow{\Psi} j}}, \\ IT_{d,i,j} = \frac{\sum_{l_1 \cap l_2 \cdots \cap l_\Psi = \{d\}} IT_{l_1,i \xrightarrow{1} j} \cdots IT_{l_\Psi,i \xrightarrow{\Psi} j}}{1 - \sum_{l_1 \cap l_2 \cdots \cap l_\Psi = \emptyset} IT_{l_1,i \xrightarrow{1} j} \cdots IT_{l_\Psi,i \xrightarrow{\Psi} j}}, \\ IT_{u,i,j} = \frac{IT_{u,i \xrightarrow{1} j} \cdots IT_{u,i \xrightarrow{\Psi} j}}{1 - \sum_{l_1 \cap l_2 \cdots \cap l_\Psi = \emptyset} IT_{l_1,i \xrightarrow{1} j} \cdots IT_{l_\Psi,i \xrightarrow{\Psi} j}}. \end{cases} \quad (7)$$

Finally, substituting the total interaction outcome $\mathrm{a}_i = (a_{b,i}, a_{d,i}, a_{u,i})$ into the general trust score T and $c(\mathrm{a}')$, the trust score of node i can be calculated as T_i, i.e., $T_i = (T_{b,i}, T_{d,i}, T_{u,i})$, expressed in (8).

V. Aid to Decision-Making Using Trust

By inferring trust to aid decision-making, TPM helps the system offer effective and efficient services by selecting trustworthy workers to generate reliable transaction data. Next, we will elaborate on how trust decision is made during *transaction generation and dissemination*, *transaction evaluation and block generation*, and *block validation* processes.

A. Transaction Generation and Dissemination

In order to issue a transaction *Tx* and let the other nodes validate it, each transaction publisher first creates a storage unit to store sensed data and transmits *Tx* to the associated leader node.

Considering that individual participants collect and share sensing data at the edge of mobile networks, the impact of channel fading and interference hinders the transaction delivery ratio. To measure the level of delivered transaction, we represent the transaction delivery probability as the ratio of transaction data that are successfully delivered to the destination from the source. Given the transaction data size $\mathrm{tx}_i^{\mathrm{size}}$, the transaction delivery probability $\mathcal{P}_{i,j}^{\mathrm{tx}}$ can be approximately calculated as

$$\begin{aligned} P_{i,j}^{\mathrm{tx}} &= \left(1 - \int_0^\infty P_{i,j}^{\mathrm{ber}}(\gamma)\, p(\gamma)\, d\gamma \right)^{\mathrm{tx}_i^{\mathrm{size}}} \\ &= \left(1 - \frac{\widehat{\alpha}_M}{2} \left[1 - \sqrt{\frac{\tilde{\gamma}\widehat{\beta}_M}{2 + \tilde{\gamma}\widehat{\beta}_M}} \right] \right)^{\mathrm{tx}_i^{\mathrm{size}}}, \end{aligned} \quad (9)$$

where γ is the signal-to-interference plus noise ratio (SINR), $p(\gamma)$ is the probability of density function (PDF) of γ, and $P_{i,j}^{\mathrm{ber}}(\gamma)$ is the bit error probability for a given γ. Note that $P_{i,j}^{\mathrm{ber}}(\gamma)$ is determined by the specific modulation and

$$T_i = \Big(\underbrace{c(\mathrm{a}'_i) \frac{\tau_b a_{b,i} + \tau_b a'_{b,i}}{\sum_{o \in \{b,d,u\}} (\tau_o a_{o,i} + \tau_o a'_{o,i})}}_{T_{b,i}}, \underbrace{c\left(\mathrm{a}'_i\right) \frac{\tau_d a_{d,i} + \tau_d a'_{d,i}}{\sum_{o \in \{b,d,u\}} (\tau_o a_{o,i} + \tau_o a'_{o,i})}}_{T_{d,i}}, \underbrace{1 - T_{b,i} - T_{d,i}}_{T_{u,i}} \Big). \tag{8}$$

coding schemes, such as M-PAM, M-PSK and M-QAM. Generally, $P_{i,j}^{\mathrm{ber}}(\gamma)$ can be the represented as a generic form, i.e., $P_{i,j}^{\mathrm{ber}}(\gamma) = \widehat{\alpha}_M \mathrm{Q}(\sqrt{\widehat{\beta}_M \gamma})$, where $\widehat{\alpha}_M = \alpha_M / \log M$ and $\widehat{\beta}_M = \beta_M / \log M$ [31]. Here, α_M and β_M are the modulation-specific constants, and $\mathrm{Q}(\cdot)$ is the Gaussian Q-function [31], [32]. Furthermore, we consider that fading channels follow a Rayleigh distribution, and thus γ is exponentially distributed with mean value $\tilde{\gamma}$, such that $\tilde{\gamma} = \mathrm{E}[|h_{i,j}|^2 p_j / (N_0 + \mathcal{I})]$, where $h_{i,j}$ is the channel gain, N_0 is the noise power, p_j is the received power of the target node j, and $\mathcal{I}$ is the interference power.

B. Transaction Evaluation and Block Generation

In each consensus group, the leader node generates a block by iteratively executing PoW, until a nonce that satisfies the difficulty requirements is found. Once the leader is elected, other nodes in the associated consensus group, called followers, must trust any requests from the leader. However, the leader election brings a concern that abnormal nodes may pose threats to the consensus process. To ensure the randomness and democracy of leader selection, any node can start the leader election, but abnormal nodes can slow down the system progress or even interrupt the current consensus process. To reduce the adverse impact of abnormal behavior on block generation, trust assessment can guide nodes to make trust decision, so that highly-trusted nodes have more opportunities to be elected leaders.

For leader-based consensus protocols, PBFT [10] and Raft [33] are efficient ways to achieve consistency of distributed nodes for consortium and private networks. Because Raft has high transaction throughput and low communication complexity compared with PBFT [34], we resort to Raft to perform trusted-leader election in this paper. Note that trusted-leader election can also be applied to PBFT with appropriate modifications. Unlike randomized leader elections based on Raft and PBFT, a majority rule can be used to decide to elect a trusted leader, denoted as

$$j^* = \underset{i,j \in \mathcal{G}_g}{\mathrm{majority}}(v_{i,j}), \tag{10}$$

where $v_{i,j}$ is a vote from the i-th follower to the j-th candidate leader, $\mathrm{majority}(\cdot)$ is a function that indicates whether candidate leader j obtains a majority of votes, and j^* denotes the election result, i.e., candidate leader j is elected as the leader.

In addition, one or more candidate nodes attempt to trigger leader election using randomized election timeouts for fairness. Let the timeout interval be $[t_1, t_2]$, the timeout of each node t can be randomly set in $t \in [t_1, t_1 + (1 - T_{b,j})^{\varepsilon}(t_2 - t_1)]$, where ε is a constant used to scale down ($\varepsilon > 1$) or scale up ($0 < \varepsilon < 1$). Obviously, this simple way makes trustworthy nodes have the larger probability to be candidate nodes, while ensuring randomness.

The elected leader regularly sends heartbeats to the associated followers to maintain authority. Whenever a follower receive a heartbeat, it should reset an election timeout to a random value. In summary, the followers start the leader election process based on the following steps:

Step 1: If any follower does not receive heartbeats after a timeout, the node that finishes the timeout first becomes the candidate leader, votes itself and sends a voting request to other followers.

Step 2: When the followers receive the voting request, they close the local timeout. Meanwhile, the followers validate the consistency and integrity of DAG snapshot of candidate leader. If the DAG is verified successfully, each follower calculates the trust assessment $T_{i,j}$ based on the interaction history. According to validation results and trust score, each node votes with a ternary-opinion $\langle 1, 0, -1 \rangle$, expressed as

$$v_{i,j} = \begin{cases} 1, & \text{if } T_{i,j} \geq \tau, \\ 0, & \text{if } T_{i,j} < \tau, \\ -1, & \text{if validation fails}, \end{cases} \tag{11}$$

where τ is a trust score threshold, which can be determined by the average trust score of $T_{i,j}$.

Step 3: Once the candidate leader obtains the majority of votes, it wins the election and sends heartbeats to other followers. In the subsequent duration, the leader node needs to perform transaction evaluation to include eligible transactions in the block. Specifically, the evaluation function on the leader node computes the trust score T_i of transaction publishers and selects the eligible transactions with high trust score. If a transaction is successfully added to the ledger, the interaction outcomes can be updated to $\mathrm{a}_{b,j,i} = \mathrm{a}_{b,j,i} + 1$ and $\mathrm{a}_{b,i} = \mathrm{a}_{b,i} + 1$ accordingly by the leader node. Otherwise, $\mathrm{a}_{d,j,i} = \mathrm{a}_{d,j,i} + 1$ and $\mathrm{a}_{d,i} = \mathrm{a}_{d,i} + 1$.

After the term of leader node ends, the evaluation function on transaction publishers computes the trust score of the leader by verifying the newly generated block. Depending on the outcome of blockchain interaction, transaction publishers update the interaction outcome to $\mathrm{a}_{b,i,j} = \mathrm{a}_{b,i,j} + 1$ and $\mathrm{a}_{b,j} = \mathrm{a}_{b,j} + 1$ (or $\mathrm{a}_{d,i,j} = \mathrm{a}_{d,i,j} + 1$ and $\mathrm{a}_{d,j} = \mathrm{a}_{d,j} + 1$) accordingly. Note that lazy leaders in the system can slow down the consensus process, which may not be effectively detected by the blockchain. Therefore, we employ a statistic evaluation function to detect lazy leader nodes. By observing historical delay data of the interaction completed by the leaders, the average delay $\hat{\mu}$ and the variance $\hat{\sigma}^2$ can be counted. If a new observation is greater than $\hat{\mu} + 3\hat{\sigma}$, the leader can be regarded as a lazy node. As a result, the interaction outcomes $\mathrm{a}_{u,i,j}$ and $\mathrm{a}_{u,j}$ can be updated by transaction publishers.

C. Block Validation

To enable a candidate block to be included in DAG, the TPM in blockchain-assisted MCS should process the below stages:

Stage 1: Once a candidate block is generated, the leader node first randomly selects some candidate tips (not exceeding the size of the set of visible tips).

Stage 2: Then the leader node validates the candidate tips, while executing trust assessment for the eligible candidate tips and sorting them in descending order of trust score.

Stage 3: Next the candidate block chooses the top-k tips with the highest trust score from the eligible candidate tips, and references the hash of k tips in DAG.

Stage 4: In addition to containing transactions, timestamp, leader ID and trust assessment of the leader, the hashes of the k tips are added into the candidate block. After that, the candidate block will be propagated to other consensus groups for cross-validation.

Through the above process, the successfully validated block can be added into the DAG as a new tip. As subsequent blocks arrive at the DAG for continuous approvals, the candidate block will eventually become a block until cumulative weight reaches a defined threshold.

It is worth noting that block validation should select a set of visible tips. To improve the diversity and freshness of trust, we regard a tip whose timestamp plus the maximum visible timespan does not exceed the current time as a visible tip. As such, the tips of highly-trusted nodes can be assessed and validated by more nodes within the maximum visible timespan, while the tips of abnormal nodes can be isolated due to the less selection.

D. Group Transaction Throughput

During the interaction process of blockchain, the leader can easily become a communication bottleneck. Therefore, successful interactions play a vital role not only in building trust among nodes, but also in improving group transaction throughput. In blockchain, the maximum transaction throughput is mainly determined by block size $\mathrm{B}^{\mathrm{size}}$, transaction number $\mathrm{B}^{\mathrm{num}}$ included in a block, block interval ξ^{invl}. Specifically, block size $\mathrm{B}^{\mathrm{size}}$ is determined by the size of sensed data, block interval ξ^{invl} is the required time that a leader node to publish a block. Hence, the transaction throughput of the g-th group at time t is given by

$$\mathrm{TPS}_g\left(t\right)=\frac{\min\left\{\mathrm{B}^{\mathrm{size}},\mathrm{Tx}_g\left(t\right)\right\}}{\mathrm{tx}^{\mathrm{size}}\xi^{\mathrm{invl}}}, \tag{12}$$

where $\mathrm{Tx}_g\left(t\right)$ is the communication throughput of the g-th group at time t, and $\mathrm{tx}^{\mathrm{size}}$ is the average transaction size.

The communication throughput $\mathrm{Tx}_g\left(t\right)$ can be expressed as

$$\mathrm{Tx}_g\left(t\right)=\sum_{i\in\mathcal{G}_g}\lambda_i\mathrm{tx}_i^{\mathrm{size}}\mathcal{P}_{i,j}^{\mathrm{tx}}, \tag{13}$$

where λ_i is the transaction generation rate of the i-th node in unit time.

For block interval ξ^{invl}, it is mainly determined by transaction computation delay ξ^{task} and block generation delay ξ^{puzzle}.

1) Transaction computation delay ξ^{task}: Upon receipt of transactions, the leader node should process sensed data in each transaction. Let c_i be the computation density (CPU cycles per transaction), and the computational capability of the associated leader be f_j (CPU cycles per second). ξ^{task} is calculated by $\xi^{\mathrm{task}}=\sum_{i\in\mathcal{G}_g}\frac{\lambda_i c_i}{f_j}$.

2) Block generation delay ξ^{puzzle}: To have the right to create a candidate block, the leader node should perform a hash operation to solve a cryptographic puzzle. The required time is probabilistically determined and depends on the target difficulty value D and hash rate $hashrate_j$ (the number of hash operation per second). The average time for successfully creating a candidate block ξ^{puzzle} is exponentially distributed with block generation rate $rate_j^{\mathrm{block}}=hashrate_j/D$ [35], [36].

In summary, given the maximum block interval $\mathrm{B}^{\mathrm{invl}}$, ξ^{invl} can be approximately expressed as

$$\xi^{\mathrm{invl}}=\min\left(\mathrm{B}^{\mathrm{invl}},\xi^{\mathrm{task}}+\xi^{\mathrm{puzzle}}\right). \tag{14}$$

VI. Trust Decision For Transaction Size Optimization

Different from the lightweight data in financial field, blockchain-assisted MCS packages sensed data into transactions, which poses a challenge to resource-constrained mobile devices and wireless networks. To address the issue, this section aims to make a trust decision to optimally plan transaction size by exploiting the priority of trust assessment. As such, this trust decision lowers down the throughput share of unreliable transaction data in DAG, thereby reducing excessive latency and computational overhead.

A. Transaction Size Optimization

As noted, the tips observed by each node at the current time need to be approved by incoming blocks at the next time. To measure the backlog level of tips, the dynamic equation of the throughput of tips can be expressed as

$$C_g\left(t\right)=\max\left[C_g\left(t-1\right)-D_g\left(t\right),0\right]^{+}+A_g\left(t\right), \tag{15}$$

where $C_g\left(t\right)$ is the throughput of unapproved tips at time t, $D_g\left(t\right)$ is the throughput of approved tips at time t, $A_g\left(t\right)$ is the throughput of incoming tips at time t, and $A_g\left(t\right)=\min\left\{\mathrm{B}^{\mathrm{size}},\mathrm{Tx}_g\left(t\right)\right\}$.

In fact, the backlog level of tips can be used to measure the dynamic changes of transactions in the system. To adjust transaction throughput adaptively based on backlog level, Lyapunov optimization theory is an effective modeling method, which can deal with the long-term average optimization problem in a stochastic system. Hence, a transaction size optimization problem based on Lyapunov optimization and trust assessment is formulated in the paper to jointly optimize transaction throughput and backlog level. To measure the system's instability, we define a quadratic function with respect to tip throughput at each time, which can be regarded as a Lyapunov function [37], expressed as $L\left(t\right)=\frac{1}{2}\sum_{g=1}^{\mathcal{G}}\left[C_g\left(t\right)\right]^2$.

Following the Lyapunov optimization framework, now we can define a one-unit Lyapunov drift $\Delta L(t)=\mathbb{E}[L\left(t+1\right)-$

$L(t)]$ to represent the difference in the Lyapunov function from time t to $(t+1)$. Considering the gain after taking control actions, the Lyapunov drift-minus-gain function can be represented as

$$\Delta L(t) - V \cdot \sum_{g=1}^{\mathcal{G}} \mathbb{E}\left[\mathrm{TPS}_g(t)\right], \tag{16}$$

where V is a non-negative control parameter to measure the importance of transaction throughput maximization compared with the tip stability. With the control parameter V, we can balance the transaction throughput maximization and the tip stability by minimizing the drift-minus-gain function. Furthermore, the upper bound of the drift-minus-gain function can be derived based on Lemma 1 [37].

Lemma 1: (The upper bound of drift-minus-gain function).

Under the feasible transaction data, the maximum throughput of incoming tips $A_g^{\max}$ and the maximum throughput of approved tips $D_g^{\max}$, we have the upper bound of drift-minus-gain function as follows

$$\begin{aligned}\Delta L(t) - V \cdot \mathbb{E}\left[\mathrm{TPS}_g(t)\right] \le B - V \cdot \sum_{g=1}^{\mathcal{G}} \mathbb{E}\left[\mathrm{TPS}_g(t)\right] \\ - \sum_{g=1}^{G} \mathbb{E}\left[C_g(t-1) D_g(t)\right] + \sum_{g=1}^{G} \mathbb{E}\left[C_g(t-1) A_g(t)\right],\end{aligned} \tag{17}$$

where $B = \frac{1}{2}\sum_{g=1}^{\mathcal{G}} \left(D_g^{\max}\right)^2 + \frac{1}{2}\sum_{g=1}^{\mathcal{G}} \left(A_g^{\max}\right)^2$.

Proof: In (17), we can observe that the proof of upper bound of drift-minus-gain function in (17) is equivalent to that of the Lyapunov drift $\Delta L(t)$. Therefore, we only need to prove $\Delta L(t) \le B - \sum_{g=1}^{\mathcal{G}} \mathbb{E}[C_g(t) D_g(t)] + \sum_{g=1}^{\mathcal{G}} \mathbb{E}[C_g(t) A_g(t)]$.

For $\Delta L(t)$, we have

$$\begin{aligned}\Delta L(t) &= \mathbb{E}[\frac{1}{2}\sum_{g=1}^{\mathcal{G}} \left[C_g(t+1)\right]^2 - \frac{1}{2}\sum_{g=1}^{\mathcal{G}} \left[C_g(t)\right]^2] \\ &= \mathbb{E}[\frac{1}{2}\sum_{g=1}^{\mathcal{G}} (\left[D_g(t+1)\right]^2 + \left[A_g(t+1)\right]^2 + \\ &2\left(A_g(t+1) - D_g(t+1)\right) C_g(t) - 2 D_g(t+1) A_g(t+1))]\end{aligned} \tag{18}$$

$$\le B + \mathbb{E}[\sum_{g=1}^{\mathcal{G}} C_g(t)\, \mathbb{E}\left[A_g(t+1) - D_g(t+1)\right]] \tag{19}$$

The proof is completed.

Compared to minimizing the drift-minus-gain function in (16), minimizing the upper bound of drift-minus-gain function in (17) is more tractable. Hence, we can minimize the upper bound of drift-minus-gain function to jointly ensure tip stability and achieve a higher transaction throughput. Given the states of incoming and approved tips, an adaptive transaction size generation decision can be made based on the following Lyapunov optimization:

$$\begin{aligned}\mathbf{P1}: \min_{\left\{\mathrm{tx}_i^{\mathrm{size}}\right\}} & B - V \sum_{g=1}^{\mathcal{G}} \mathrm{TPS}_g(t) - \\ & \sum_{g=1}^{\mathcal{G}} C_g(t-1) D_g(t) + \sum_{g=1}^{\mathcal{G}} C_g(t-1) A_g(t), \end{aligned} \tag{20}$$

$$\text{s.t.} \quad C_1: \mathrm{tx}^{\min} \le \mathrm{tx}_i^{\mathrm{size}} \le \mathrm{tx}^{\max}, i \in \mathcal{G}_g, \tag{20-1}$$

$$C_2: \frac{\min\left\{\mathrm{B}^{\mathrm{size}}, \sum_{i\in\mathcal{G}_g} \mathrm{tx}_i^{\mathrm{size}}\right\}}{\mathrm{tx}^{\mathrm{size}}} \le \mathrm{B}^{\mathrm{num}}, \tag{20-2}$$

$$C_3: T_{b,i_1} \ge T_{b,i_2} \ge T_{b,i_3} \cdots, \tag{20-3}$$

where constraint C_1 specifies the size of transaction data, constraint C_2 specifies the region of $\mathrm{tx}_i^{\mathrm{size}}$ given the maximum transaction size $\mathrm{B}^{\mathrm{num}}$, and constraint C_3 indicates the descending order of trust score $T_{b,i}$ for selected transactions.

B. Optimal Transaction Size

From (P1), we can observe that there is no coupling relationship between $\mathcal{G}$ groups with respect to $\mathrm{tx}_i^{\mathrm{size}}$. Hence, (P1) can be decomposed into $\mathcal{G}$ sub-problems to solve. Through equivalent conversions, the optimization problem in (P1) can be converted into the following form:

$$\mathbf{P2}: \min_{\left\{\mathrm{tx}_i^{\mathrm{size}}\right\}} I_1 - I_2 + I_3 \sum_{i\in\mathcal{G}_g} \lambda_i \mathrm{tx}_i^{\mathrm{size}} I_4^{\mathrm{tx}_i^{\mathrm{size}}}, \tag{21}$$

$$\text{s.t.} \quad C_1, C_2, \text{ and } C_3, \tag{21-1}$$

where I_1, I_2, I_3 and I_4 are constants related to system parameters: $I_1 = \frac{1}{2}((D_g^{\max})^2 + (A_g^{\max})^2)$, $I_2 = C_g(t-1)D_g(t)$, $I_3 = C_g(t-1) - \frac{V}{\mathrm{tx}^{\mathrm{size}}\xi^{\mathrm{invT}}}$, and $I_4 = (1 - \frac{\widehat{\alpha}_M}{2}[1 - \sqrt{\frac{\tilde{\gamma}\widehat{\beta}_M}{2+\tilde{\gamma}\widehat{\beta}_M}}])$.

Taking the derivative of the optimization objective in (P2) with respect to $\mathrm{tx}_i^{\mathrm{size}}$, its first-order derivative is equal to $\lambda_i I_3 I_4^{\mathrm{tx}_i^{\mathrm{size}}} \left(1 + \mathrm{tx}_i^{\mathrm{size}} \ln I_4\right)$. The optimal transaction size can be obtained by investigating the following two cases:

Case 1. $I_3 < 0$: The optimization objective in (P2) first decreases monotonically and then increases monotonically. Hence, the optimal transaction size could be obtained at stationary point $-\frac{1}{\ln I_4}$ or the upper boundary point, expressed as

$$\mathrm{tx}_i^{\mathrm{size}*} = \begin{cases} -\frac{1}{\ln I_4}, & \text{if } -\frac{1}{\ln I_4} \le \mathrm{tx}^{\max} \\ \mathrm{tx}^{\max}, & \text{if } -\frac{1}{\ln I_4} > \mathrm{tx}^{\max} \end{cases}. \tag{22}$$

Case 2. $I_3 > 0$: Similarly, the optimal transaction size can be obtained at the lower boundary point, expressed as $\mathrm{tx}_i^{\mathrm{size}*} = \mathrm{tx}^{\min}$.

VII. Numerical Results

In this section, we validate the effectiveness of the proposed TPM in blockchain-assisted MCS and evaluate several critical performance metrics, including contribution rate, consensus accuracy, transaction throughput and tips stability. Specifically, the contribution rate is a measure of effectiveness, defined as the proportion of approved transactions under a threshold in

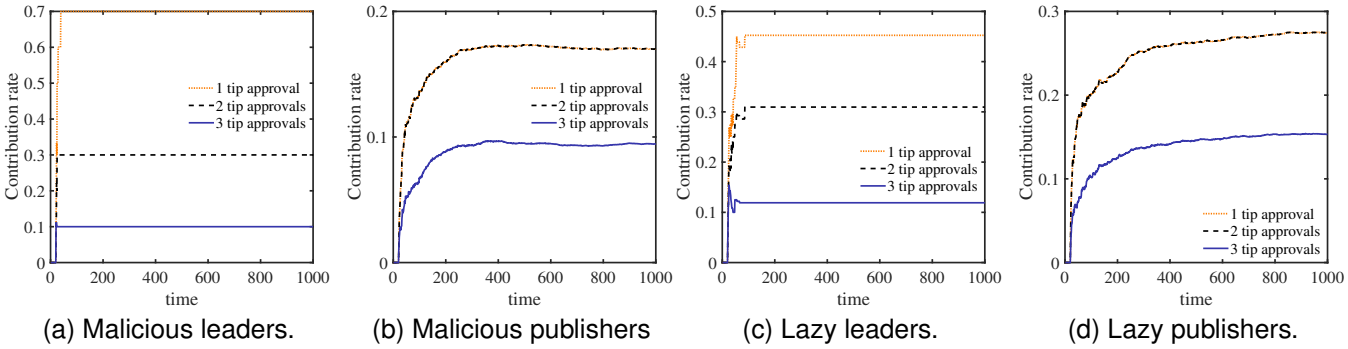


Fig. 2. Impact of tip approvals on contribution rate of abnormal nodes.

the total number of published transactions. To reflect the trustworthiness of a leader, the consensus accuracy is statistically computed by measuring the proportion of normal nodes in the total number of elected nodes. Based on (12), the transaction throughput of a group can be calculated. In addition, the tip stability can be measured by the average number of tips of all nodes in the system.

A. Experimental Settings

We consider a blockchain-assisted MCS underlaying a wireless network scenario. Assume that the 10 groups are constructed randomly and independently, the network coverage of each group is set to 150 square meters, and 100 nodes are randomly located in this area, where the transmit power of each node is 20 dBm, the noise power is -104 dBm, and path loss exponent is 2.5 [38]. In addition, K is set to -31.54 dB, $\widehat{\alpha}_M$ and $\widehat{\beta}_M$ are 1 and 2, respectively [31]. To simplify, each node is allocated 1 Mhz bandwidth, and CPU frequency of each node and computation density are randomly generated in $[1, 2]$ GHz [39]. In addition, the SHA-256 hash function is used to generate data hash in this paper.

We classify the node status into normal and abnormal nodes, where abnormal nodes include malicious and lazy ones. More specifically, malicious leaders may poison the received transaction data, while a lazy leader may slow down the consensus process. In addition, malicious publishers issue more useless transactions in a short time to obtain more rewards, while lazy publishers publish fewer or even no transactions. Since abnormal nodes may behave normally to defraud trust, we assume that abnormal nodes publish transactions or blocks with a probability p. In this paper, we set the probability p to $2/3$ and the number of abnormal nodes to 30, unless stated otherwise.

In the process of TPM in blockchain-assisted MCS, we set the rate at which each node publishes transactions to 0.5 transactions per unit time. The generated transaction data size is limited to $[10, 100]$ Kb, and the maximum number of transactions in a block is 50. To ensure that the tips from trustworthy nodes get more approvals, we set the maximum visible timespan to 20 units. In the visible timespan, the new incoming blocks should select 10 tips to authenticate, and two of them will be referenced by incoming blocks.

B. Performance Comparisons

In this subsection, we conduct three experiments to compare the performance of TPM in DAG-based blockchain (called TPM-BlockDAG) with three baseline schemes as follows:

- Dirichlet-based BlockDAG (Dirichlet-BlockDAG): In [27], the authors employ a blockchain to record historical trust information. By classifying the behavior of participants into several ranks, the Dirichlet distribution is used to infer the trust score for a behavior at a specified rank.
- Poof of reputation-based BlockDAG (PoR-BlockDAG): In [40], the authors propose a reputation-based consensus protocol to promote successful interactions in blockchain. Essentially, the PoR adopts a sigmoid function to infer the trust score and thus elects a leader who has the highest trust score.
- Poof of work-based BlockDAG (PoW-BlockDAG): It is an original PoW-based BlockDAG without relying on a trust/reputation-based incentive mechanism [41].

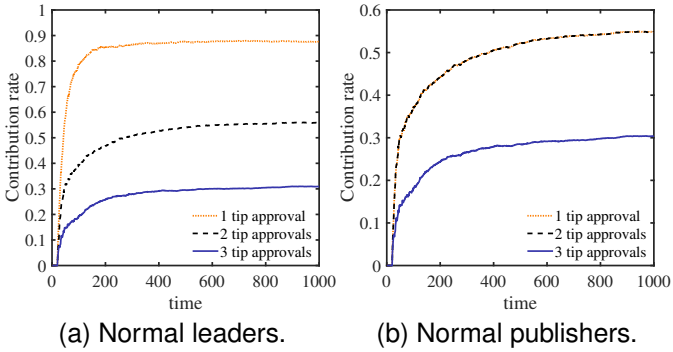


Fig. 3. Impact of tip approvals on contribution rate of normal nodes.

1) Impact of the number of tip approval times on contribution rate: In the process of TPM BlockDAG, we regard the blocks whose tip approval times are less than a certain number as untrusted blocks, and these blocks will be considered isolated without any contribution. In fact, we expect that the blocks published by normal nodes can get more approval times, so as to have a greater contribution rate.

In this experiment, we evaluate the impact of the number of tip approval times on the contribution rate. As shown in Fig. 2, we can observe that the contribution rate of abnormal

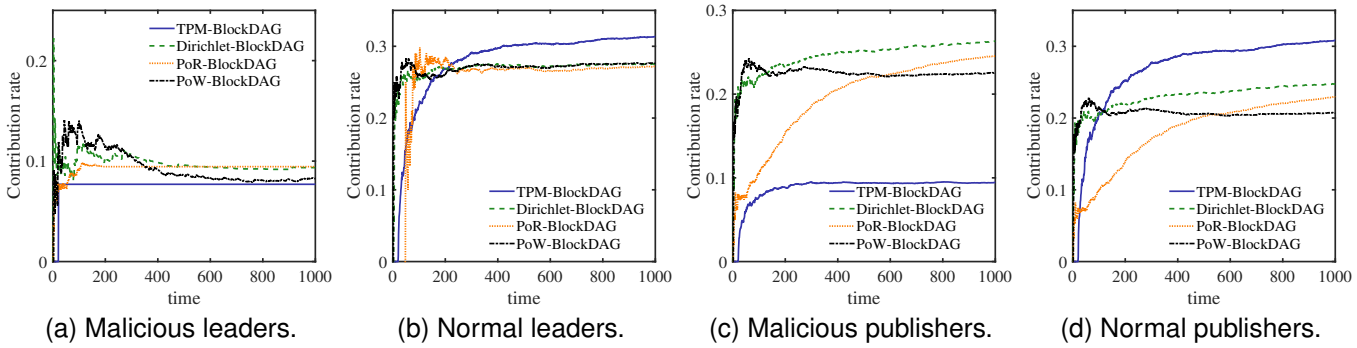


Fig. 4. Comparisons of contribution rate under malicious behavior.

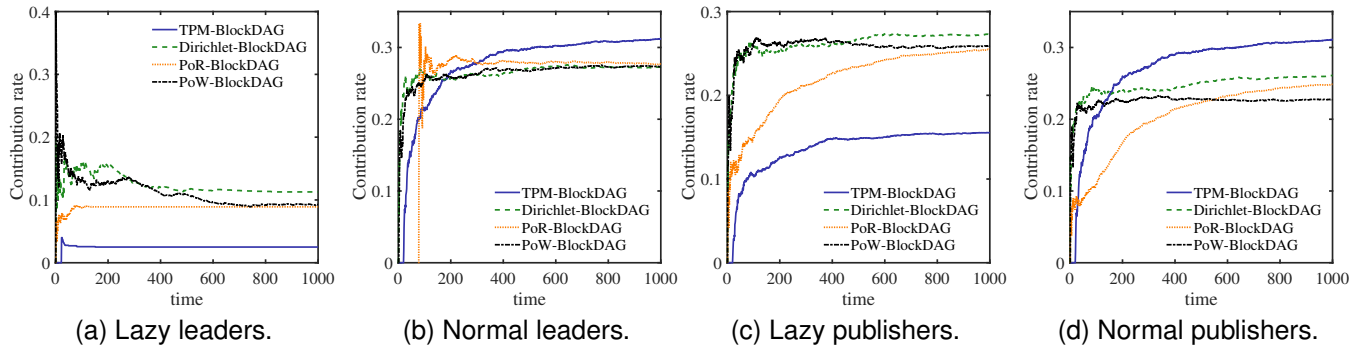


Fig. 5. Comparisons of contribution rate under lazy behavior.

leaders and publishers decreases with the number of tip approvals, where the thresholds for tip approval is set to 1, 2 and 3, respectively. On the one hand, the larger threshold for tip approvals makes it more difficult to approve tips from malicious and lazy nodes, because the proposed TPM can incentivize the tips from trustworthy nodes to get more approvals. On the other hand, we can see that the contribution rate of abnormal leaders and publishers keeps basically stable over time, which indicates that the blocks and transactions published by abnormal leaders and publishers can be isolated as much as possible in the case of 3 tip approvals.

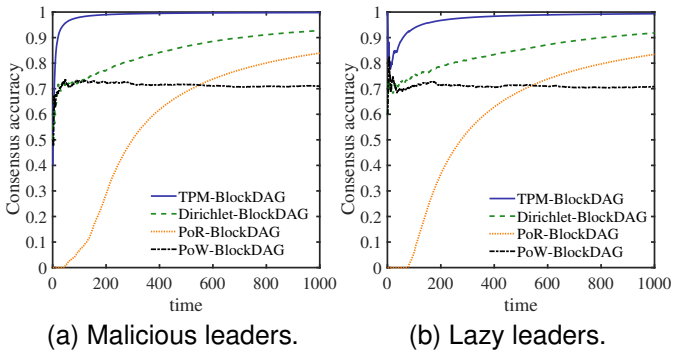


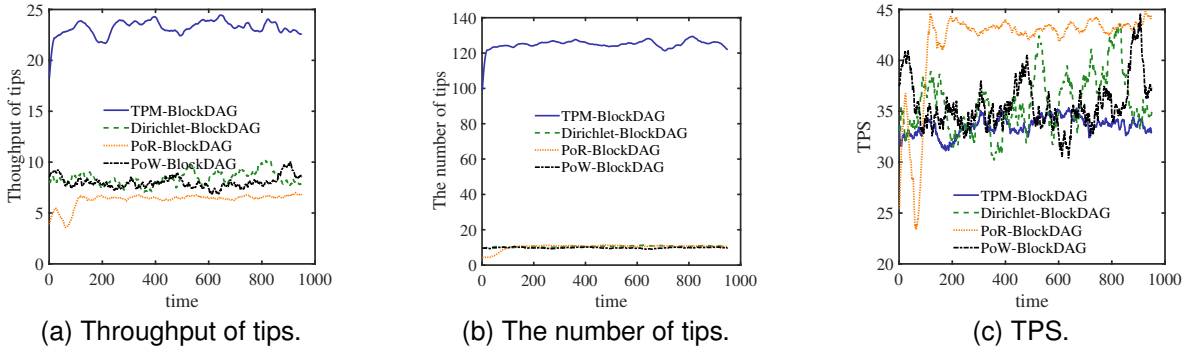
Fig. 6. Consensus accuracy.

Furthermore, Fig. 3 illustrates the contribution rate of normal leaders and publishers. Obviously, the larger number of tip approvals inevitably decreases the contribution rate of normal leaders and publisher. However, the contribution rate of normal leaders and publishers in Fig. 3 is greater than that of abnormal ones in Fig. 2, while the contribution rate of normal leaders and publishers gradually increases monotonically. Therefore, there is a tradeoff between increasing the contribution rate of normal nodes and decreasing the contribution rate of abnormal nodes. In the following experiments, we consider 3 tip approvals scheme to effectively help TPM-BlockDAG resist abnormal nodes.

2) Contribution rate comparisons: Next, we conduct an experiment to compare the contribution rate of the proposed TPM-BlockDAG with benchmarks.

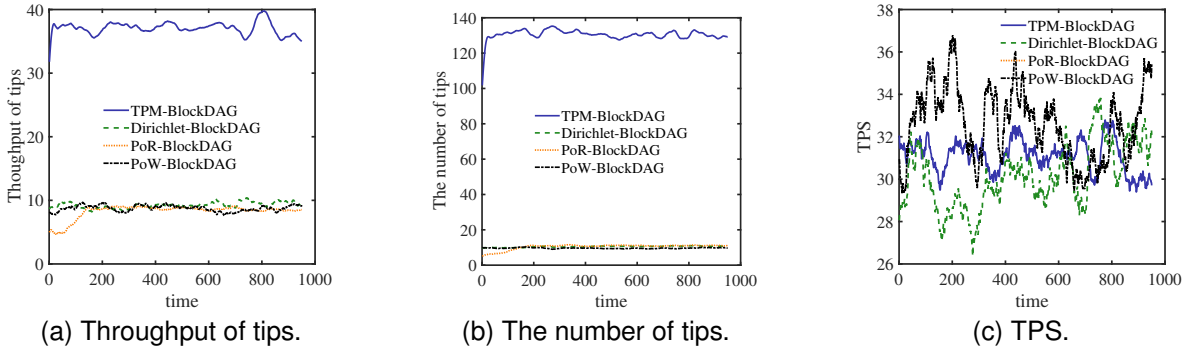
Fig. 4 illustrates the contribution rate under malicious behavior over time. In this experiment, malicious leaders may poison the information in blocks. In Figs. 4a and 4b, TPM-BlockDAG reduces the contribution rate of malicious leaders compared to Dirichlet-BlockDAG, PoR-BlockDAG and PoW BlockDAG, while the contribution rate of normal leaders is greater than that of other schemes. In fact, PoW-BlockDAG can use the hash function to ensure that poisoned blocks cannot be successfully added to the ledger. Obviously, TPM-BlockDAG can enable trustworthy leaders to be elected as the initiators in the process of BlockDAG, resulting in more positive contributions. Similarly, Figs. 4c and 4d show that the contribution rate of TPM-BlockDAG is greater than that of other schemes for malicious and normal publishers. This is because the trust score of malicious publishers is lower than that of normal publishers over time, so that more transactions published by normal publishers can be packaged into blocks.

Furthermore, Fig. 5 shows the contribution rate under lazy



(a) Throughput of tips. (b) The number of tips. (c) TPS.

Fig. 7. Comparisons of throughput under malicious behavior.



(a) Throughput of tips. (b) The number of tips. (c) TPS.

Fig. 8. Comparisons of throughput under lazy behavior.

behavior. Similarly, the trust score of lazy leaders and publishers is lower than that of normal ones over time, resulting in a lower probability of getting approval. Owing to this, the contribution rate of TPM-BlockDAG outperforms that of other schemes for lazy leaders and publishers.

3) Consensus accuracy comparisons: In the consensus process, abnormal nodes may be selected as leaders, which can slow down and even interrupt the consensus process.

Fig. 6 demonstrates the consensus accuracy under malicious and lazy behavior. Obviously, TPM-BlockDAG significantly outperforms other schemes in terms of consensus accuracy. In particular, the consensus accuracy of TPM-BlockDAG can quickly approach 1 compared to other schemes. This is because all schemes can use the inferred trust to motivate normal nodes to be elected as leaders and punish malicious and lazy nodes to some extent, but the proposed TPM can more accurately and comprehensively infer the trust score of nodes. In addition, PoW-BlockDAG chooses a leader randomly, resulting in significantly lower consensus accuracy under malicious and lazy behavior.

4) Throughput comparisons: In addition to the contribution rate and consensus accuracy, we also examine the impact of TPM on throughput performance including throughput of tips, the number of tips and TPS. In this regard, the transaction generation of TPM is of vital importance for improving throughput while ensuring system stability for BlockDAG, as shown in Figs. 7 and 8. Note that all simulation results in this experiment are the moving average of the previous 50 values.

Figs.7a and 8a evaluate the throughput of tips, the number of tips and TPS over time under malicious and lazy behavior, respectively. The results show that the throughput of tips of TPM-BlockDAG is significantly larger than that of other schemes and keeps basically stable over time. This is because TPM-BlockDAG regards all approved blocks within the visible timespan as tips, which increases the number of tips, as shown in Figs. 7b and 8b. In contrast, Dirichlet-BlockDAG, PoR-BlockDAG and PoW-BlockDAG only regard current approved blocks as tips, thus the number of tips is significantly less than that of TPM-BlockDAG. In fact, a reasonable visible timespan can increase the diversity of tips for approval, which is beneficial for approving blocks published by trustworthy leaders.

Furthermore, we can observe that the TPS of TPM-BlockDAG is more stable than other schemes in Figs. 7c and 8c, which verifies the effectiveness of the proposed transaction size optimization. As a result, TPM-BlockDAG can generate the optimal amount of transaction data while stabilizing throughput performance. Although the TPS of Dirichlet-BlockDAG, PoR-BlockDAG and PoW-BlockDAG may be larger than that of TPM-BlockDAG, the malicious transactions generated by untrusted nodes also pollute the blockchain ledger.

VIII. Conclusions

In this paper, we have proposed to integrate a trust assessment approach into blockchain-assisted MCS, aiming to address the trust issue arising from the interaction outside and inside the blockchain. Based on auditable records in

blockchain, we model the normal, abnormal and uncertain interaction outcomes as a probabilistic ternary-trust assessment to characterize trust, distrust and uncertainty, respectively. To achieve trust in decision-making, the trust decision including leader election and transaction generation is made to filter abnormal nodes in blockchain. The designed TPM addresses the problem of how to form a reasonable trust score under deficient interaction outcomes, derives the knowledge defects of trust score, and provides insights for effective integration of MCS and blockchain. The experimental results demonstrate that the proposed TPM can help blockchain resist abnormal behavior and outperform trust/reputation-based blockchains, as well as the blockchain without trust.

REFERENCES

[1] C. Jiang, L. Gao, L. Duan, and J. Huang, "Scalable mobile crowdsensing via peer-to-peer data sharing," *IEEE Transactions on Mobile Computing*, vol. 17, no. 4, pp. 898–912, 2018.

[2] J. Ni, K. Zhang, Q. Xia, X. Lin, and X. S. Shen, "Enabling strong privacy preservation and accurate task allocation for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 6, pp. 1317–1331, 2020.

[3] W. Y. B. Lim, J. S. Ng, Z. Xiong, J. Jin, Y. Zhang, D. Niyato, C. Leung, and C. Miao, "Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 3, pp. 536–550, 2022.

[4] W. Y. B. Lim, Z. Xiong, D. Niyato, X. Cao, C. Miao, S. Sun, and Q. Yang, "Realizing the metaverse with edge intelligence: A match made in heaven," *CoRR*, vol. abs/2201.01634, 2022. [Online]. Available: https://arxiv.org/abs/2201.01634

[5] B. Cao, Y. Li, L. Zhang, L. Zhang, S. Mumtaz, Z. Zhou, and M. Peng, "When internet of things meets blockchain: Challenges in distributed consensus," *IEEE Network*, vol. 33, no. 6, pp. 133–139, 2019.

[6] Y. Li, B. Cao, M. Peng, L. Zhang, L. Zhang, D. Feng, and J. Yu, "Direct acyclic graph-based ledger for internet of things: Performance and security analysis," *IEEE/ACM Transactions on Networking*, vol. 28, no. 4, pp. 1643–1656, 2020.

[7] R. F. Hayat, S. Aurangzeb, M. Aleem, G. Srivastava, and J. C.-W. Lin, "Ml-ddos: A blockchain-based multilevel ddos mitigation mechanism for iot environments," *IEEE Transactions on Engineering Management*, pp. 1–14, 2022.

[8] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A survey of distributed consensus protocols for blockchain networks," *IEEE Communications Surveys Tutorials*, vol. 22, no. 2, pp. 1432–1465, 2020.

[9] A. Yazdinejad, A. Dehghantanha, R. M. Parizi, M. Hammoudeh, H. Karimipour, and G. Srivastava, "Block hunter: Federated learning for cyber threat hunting in blockchain-based iiot networks," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 11, pp. 8356–8366, 2022.

[10] M. Castro, B. Liskov *et al.*, "Practical byzantine fault tolerance," in *OSDI*, vol. 99, no. 1999, 1999, pp. 173–186.

[11] "Recommendation, y.3052: Overview of trust provisioning for information and communication technology infrastructures and services," *ITU-T*, 2017.

[12] N. Mohammed, H. Otrok, L. Wang, M. Debbabi, and P. Bhattacharya, "Mechanism design-based secure leader election model for intrusion detection in manet," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 1, pp. 89–103, 2011.

[13] L. Xiao, Y. Ding, D. Jiang, J. Huang, D. Wang, J. Li, and H. Vincent Poor, "A reinforcement learning and blockchain-based trust mechanism for edge networks," *IEEE Transactions on Communications*, vol. 68, no. 9, pp. 5460–5470, 2020.

[14] J. An, Z. Wang, X. He, X. Gui, J. Cheng, and R. Gui, "Ppqc: A blockchain-based privacy-preserving quality control mechanism in crowdsensing applications," *IEEE/ACM Transactions on Networking*, pp. 1–16, 2022.

[15] Y. Wang and M. Singh, "Formal trust model for multiagent systems." in *IJCAI International Joint Conference on Artificial Intelligence*, vol. 7, 01 2007, pp. 1551–1556.

[16] L. Wei, Y. Yang, J. Wu, C. Long, and B. Li, "Trust management for internet of things: A comprehensive study," *IEEE Internet of Things Journal*, vol. 9, no. 10, pp. 7664–7679, 2022.

[17] S. Zou, J. Xi, H. Wang, and G. Xu, "Crowdblps: A blockchain-based location-privacy-preserving mobile crowdsensing system," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4206–4218, 2020.

[18] M. Li, J. Weng, A. Yang, W. Lu, Y. Zhang, L. Hou, J.-N. Liu, Y. Xiang, and R. H. Deng, "Crowdbc: A blockchain-based decentralized framework for crowdsourcing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 6, pp. 1251–1266, 2019.

[19] B. An, M. Xiao, A. Liu, Y. Xu, X. Zhang, and Q. Li, "Secure crowdsensed data trading based on blockchain," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2021.

[20] Z. Ning, S. Sun, X. Wang, L. Guo, S. Guo, X. Hu, B. Hu, and R. Kwok, "Blockchain-enabled intelligent transportation systems: A distributed crowdsensing framework," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2021.

[21] A.-R. Sadeghi, M. Selhorst, C. Stüble, C. Wachsmann, and M. Winandy, "Tcg inside? a note on tpm specification compliance," in *Proceedings of the first ACM workshop on Scalable trusted computing*, 2006, pp. 47–56.

[22] M. Huang, S. Cao, X. Li, K. Huang, and X. Zhang, "Defending data poisoning attack via trusted platform module and blockchain oracle," in *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 1245–1250.

[23] D. Chatzopoulos, S. Gujar, B. Faltings, and P. Hui, "Mneme: A mobile distributed ledger," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 2020, pp. 1897–1906.

[24] A. Charapko, A. Ailijiang, and M. Demirbas, "Pigpaxos: Devouring the communication bottlenecks in distributed consensus," in *Proceedings of the 2021 International Conference on Management of Data*, 2021, pp. 235–247.

[25] W. Yang, X. Dai, J. Xiao, and H. Jin, "Ldv: A lightweight dag-based blockchain for vehicular social networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 5749–5759, 2020.

[26] H. Yu, Z. Shen, C. Miao, C. Leung, and D. Niyato, "A survey of trust and reputation management systems in wireless communications," *Proceedings of the IEEE*, vol. 98, no. 10, pp. 1755–1772, 2010.

[27] B. Luo, X. Li, J. Weng, J. Guo, and J. Ma, "Blockchain enabled trust-based location privacy protection scheme in vanet," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2034–2048, 2020.

[28] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian data analysis*. Chapman and Hall/CRC, 1995.

[29] T. C. Silva and L. Zhao, *Machine learning in complex networks*. Springer, 2016, vol. 1.

[30] K. Sentz, S. Ferson *et al.*, *Combination of evidence in Dempster-Shafer theory*. Citeseer, 2002, vol. 4015.

[31] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[32] A. Mahmood and R. Jäntti, "Packet error rate analysis of uncoded schemes in block-fading channels using extreme value theory," *IEEE Communications Letters*, vol. 21, no. 1, pp. 208–211, 2017.

[33] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *2014 Annual Technical Conference*, 2014, pp. 305–319.

[34] L. Zhang, H. Xu, O. Onireti, M. A. Imran, and B. Cao, "How much communication resource is needed to run a wireless blockchain network?" *CoRR*, vol. abs/2101.10852, 2021. [Online]. Available: https://arxiv.org/abs/2101.10852

[35] Y. Li, B. Cao, L. Liang, D. Mao, and L. Zhang, "Block access control in wireless blockchain network: Design, modeling and analysis," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 9258–9272, 2021.

[36] B. Group, "Proof of stake versus proof of work," 2015. [Online]. Available: https://bitfury.com/content/downloads/pos-vs-pow-1.0.2.pdf

[37] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.

[38] Y. Sun, L. Zhang, G. Feng, B. Yang, B. Cao, and M. A. Imran, "Blockchain-enabled wireless internet of things: Performance analysis and optimal communication node deployment," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5791–5802, 2019.

[39] M. Cao, L. Zhang, and B. Cao, "Toward on-device federated learning: A direct acyclic graph-based blockchain approach," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2021.

[40] F. Gai, B. Wang, W. Deng, and W. Peng, "Proof of reputation: A reputation-based consensus protocol for peer-to-peer network," in *International Conference on Database Systems for Advanced Applications*. Springer, 2018, pp. 666–681.
[41] Y. Sompolinsky, Y. Lewenberg, and A. Zohar, "Spectre : Serialization of proof-of-work events : Confirming transactions via recursive elections," 2017.

Long Zhang received the M.E degree in information and communication engineering from Chongqing University of Posts and Telecommunications, Chongqing, China, in 2019. He currently is pursuing his Ph.D. degree at the National Key Laboratory of Wireless Communications, University of Electronic Science and Technology of China, Chengdu, China. His research areas include next generation mobile networks and Internet of Things.

Gang Feng (M'01–SM'06) received his BEng and MEng degrees in Electronic Engineering from the University of Electronic Science and Technology of China (UESTC), in 1986 and 1989, respectively, and the Ph.D. degrees in Information Engineering from The Chinese University of Hong Kong in 1998. At present he is a professor with the National Key Laboratory of Wireless Communications, UESTC of China. His research interests include resource management in wireless networks, next generation cellular networks, etc. Dr. Feng is a senior member of IEEE.

Shuang Qin received the B.E. degree in Electronic Information Science and Technology, and the Ph.D degree in Communication and Information System from University of Electronic Science and Technology of China (UESTC), in 2006 and 2012, respectively. He is currently an associate professor with National Key Laboratory of Wireless Communications in UESTC. His research interests include cooperative communication in wireless networks and data transmission in opportunistic networks.

Xiaoqian Li received the B.Eng. and M.Eng. degrees in communication engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2013 and 2016, respectively, and the Ph.D. degree in communication engineering from the University of Hong Kong in 2020. She is currently a Postdoctoral Researcher with the University of Electronic Science and Technology of China. Her research interests include next-generation Internet, mobile edge computing, and mobile crowd sensing.

Yao Sun is currently a Lecturer at the James Watt School of Engineering, the University of Glasgow, UK. His research interests include semantic communications, intelligent wireless networking, and wireless blockchain system.

Bin Cao (M'14) received the Ph.D. degree (Hons.) in communication and information systems from the National Key Laboratory of Wireless Communications, University of Electronic Science and Technology of China (UESTC), in 2014. He is currently an Associate Professor with the State Key Laboratory of Network and Switching Technology, Beijing University of Posts and Telecommunications (BUPT). His research interests include blockchain systems, the Internet of Things, and mobile edge computing.