Integration of Blockchain and Mobile Crowdsensing by Trust-Preserving Mechanism

Long Zhang[1], Shuang Qin[1*], Gang Feng[1], Xiaoqian, Li[1], and Yao Sun[2]
[1]National Key Laboratory of Science and Technology on Communications, and Yangtze Delta Region Institute (Huzhou), University of Electronic Science and Technology of China, Chengdu, China
[2]James Watt School of Engineering, the University of Glasgow, Glasgow, UK

***Abstract*—Blockchain has been regarded as one of the promising technologies to address trust concern in data-driven mobile crowdsensing (MCS), due to its auditability, immutability and decentralization. However, simply applying blockchain in MCS while ignoring possible abnormal saboteurs hidden in numerous devices may mislead the normal operation of blockchain, resulting in untruthful off-chain data and untrustworthy on-chain interactions. Consequently, it is highly desirable to build a trust-preserving mechanism (TPM) to bridge the gap between MCS and blockchain. To this end, in this paper we first resort to a probabilistic trust assessment model inferred from auditable interaction outcomes in blockchain, to incentivize normal nodes to maintain trustworthiness of on-chain interactions. Assisted by the trust assessment, trust decision is further made to filter untrusted nodes from participating in blockchain process and improve the authenticity of off-chain data. Simulation experiments are conducted to validate the effectiveness and efficiency of the proposed TPM-enabled blockchain in terms of contribution rate and consensus accuracy.**

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

I. Introduction

Leveraging ubiquitous devices for big data sensing, mobile crowdsensing (MCS) [1] provides data analysis and computation to customers with common interests under centralized coordination. As a large amount of data is generated from numerous devices, existing MCS system usually needs a centralized authority to provide authentication of participating devices [1], [2]. This centralized mechanism brings trust concern about data monopoly and disclosure, since an individual is hard to trace and manage decisions throughout the system lifecycle [3]. Additionally, malicious manipulation further hinders data trading, analyzing and sharing across systems.

Recently, the emerging blockchain-based distributed ledger technology [4] has been becoming a promising solution to eliminate the necessity for centralized authority in data-driven MCS system. According to specific consensus protocols, interaction nodes can effectively and reliably publish transactions to exchange data among them without a third party. By allowing interaction nodes to verify the integrity of data, blockchain records any changes of data in the global ledger and ensures its integrity and immutability. Although blockchain can ensure the authenticity of sensed data once it is confirmed on the chain, it still lacks an effective supervision over data sources off the chain. Meanwhile, abnormal behaviors of nodes inevitably affect the normal operation process on the chain. For example, malicious nodes may launch byzantine faults [5] or do not respond to other nodes in time, thus impeding the consensus process. It means that the authenticity and trustworthiness of any data and interactions cannot be guaranteed before they are confirmed by blockchain.

Incorporating trust-preserving mechanism (TPM) into blockchain is a way not only to create a chain of blocks that records sensed data in an unforgeable manner but also to establish a chain of trust of data and interaction itself. In this paper, we focus on designing a TPM for blockchain-assisted MCS, with aim of providing guidance for trust decision-making in blockchain process. Specifically, the proposed TPM consists of the trust assessment for measuring the belief level of nodes and the trust decision for filtering untrusted nodes. In this way, TPM-supported trust decision helps prevent abnormal nodes from participating in the blockchain, whereas blockchain ensures the authenticity of history interactions, thereby effectively supervising the data and interactions on and off the chain. In particular, to accurately quantify the trust assessment, we develop a probabilistic multi-class trust model to assess the outcomes of interaction in blockchain as a multinomial distribution, and express the trust assessment as ternary outcomes, i.e., belief, disbelief and uncertainty. Then, we derive the knowledge defects affecting trust assessment to measure the limitations of interaction experiences and calculate direct and indirect trust based on the Dirichlet model and the Dempster-Shafer's combination rule.

II. Framework of TPM in blockchain-assisted MCS

As shown 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. The main functions deployed in the framework include *trust assessment and decision*, *transaction generation and dissemination*, *block generation*, and *block validation*, and the corresponding workflow of blockchain interaction process and TPM is illustrated in Fig. 1.

Sensing Layer: The interaction nodes at sensing layer are responsible for realizing *transaction generation and dissemination*. At this layer, some interaction nodes, such as heterogeneous smart devices, act as working nodes to publish transactions. We consider that the interaction nodes in the system constitute G non-overlapping and non-empty groups, which can be formed according to the regions where nodes are

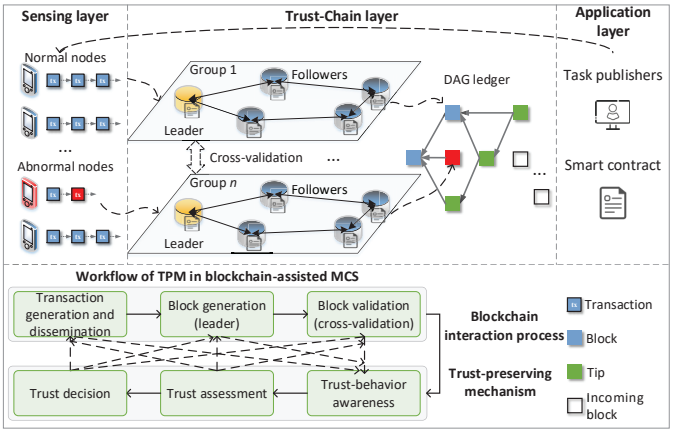


Fig. 1: The framework of TPM in blockchain-assisted MCS, where red dashed box represents the malicious transactions published by abnormal nodes.

located [6]. 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 assessment of the publisher, $Data$ is the transaction data, and $Timestamp$ is the update time of transaction generation. In the subsequent transaction dissemination stage, the signed transaction should be transmitted to the associated leader node within a group for further validation and computation.

Trust-Chain Layer: The Trust-Chain layer, constituted by leader and follower nodes, is served by verifiers to validate transactions. This layer is mainly responsible for *trust assessment and decision*, *block generation* and *block validation*. For *trust assessment and decision*, each node infers the trust level according to TPM and makes trusted decisions to encourage nodes to participate in the blockchain. The *trust assessment and decision* functionality ensures that only the trusted transactions of well-behaved nodes could be added into a block, so as to decrease interaction behaviors of abnormal nodes.

For *block generation*, according to the votes of follower nodes in each group, a trusted leader first performs transaction validation, and then endows the right to create a block for received transactions, and initiates the consensus process. In this paper, we can reasonably expect that the trusted nodes have more chances to be elected.

For *block validation*, the verifiers in a group validate the correctness of received blocks, including hash digest, trust assessment, 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 directed acyclic graph (DAG)-based blockchain to drive MCS. In DAG, some published and yet unapproved candidate blocks (called tips) should be approved by incoming blocks, yielding a forking-chain topology. After that, the incoming block also becomes a tip waiting for subsequent approvals. With the continuous arrival of incoming blocks, the cumulative weight of a tip eventually becomes a valid block when a predefined threshold is met. Note that the cumulative weight of a tip is calculated as the sum of the weight of the tip itself and the weights directly and indirectly approved by all blocks in DAG. According to the TPM in blockchain-assisted MCS, the incoming block prefers to approve Top-k tips with the highest trust from a set of visible tips. In Fig. 1, we depict the DAG ledger when there exist 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.

Application Layer: The application layer is at the top and uses application programming interfaces (APIs) to allow publishers to access. According to the requirements of APIs, nodes will be identified and interface the appropriate group and corresponding smart contracts. With the initialization operation, the smart contract will be installed and instantiated at the targeted leaders and conduct sensing tasks.

III. Trust Assessment of TPM in Blockchain-Assisted MCS

By interacting with each other, nodes are interlinked by their trust relationship. The resulted interaction outcome (typically positive/negative) can be used as the first-hand observation to infer a trust assessment. For example, to mitigate counterparty risk, Bitcoin-otc marketplace infers the trust level of a user by aggregating the number of positive and negative ratings[1]. Considering the uncertain interaction outcomes, we express the general trust assessment as a ternary trust $T = (T_b, T_d, T_u)$ in $[0, 1]$, where T_b is the belief degree of normal interaction outcome b, T_d is the disbelief degree of abnormal interaction outcome d and T_u is the uncertainty degree of uncertain interaction outcome d [7].

A. Trust Assessment Process

In order to accurately assess trust, the TPM operates in two phases: trust behavior-awareness and trust assessment.

1) Trust behavior-awareness: The trust behavior-awareness provides a series of interaction behaviors required for conducting trust assessment. Owing to the auditability and immutability of blockchain, *trust behavior-awareness* can identify interaction outcomes b, d and u from *transaction generation and dissemination*, *block generation* and *block validation*.

- 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 assessment and decision* to *block validation*. This can be detected as $\texttt{Hash}(PreHash, Merkle, Nonce) \leq Target$, where $\texttt{Hash}(\cdot)$ is the hash operation, $PreHash$ is the hash value of the previous block, $Merkle$ is the root of the Merkle tree containing transactions in the block, and $Target$ is a numeric value that a valid block must be less than or equal to.
- The interaction behavior results in outcome d if any of the functions from *trust generation* to *block validation*

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

fails. The abnormal interactions are typically caused by Byzantine failures, i.e., some of the nodes fail in responding or interact maliciously. This can be detected as $\texttt{Hash}(PreHash, Merkle, Nonce) > Target$.

- The interaction behavior results in outcome u if the generated transaction cannot be included in DAG ledge due to some uncontrollable reasons. In this regard, we consider that some leaders cannot complete an interaction during their term due to lazy behavior, resource constraints, etc, recorded as outcome u.

In the following, we use $\mathrm{a}_{i,j} = (a_{b,i,j}, a_{d,i,j}, a_{u,i,j})$ to denote the 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 trust level between a pair of interaction nodes from history interactions. 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 heterogeneous 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 behaviors and prevent the proportion of normal outcomes b from increasing rapidly, τ_d and τ_u are usually larger than τ_b.

To assess trust for different interaction outcomes, the Dirichlet distribution can be used to map the multi-class interactions into a probability distribution [8]. Hence, 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 $\Theta = (\Theta_b, \Theta_d, \Theta_u)$. Based on the above analysis, we can express the probability density function (PDF) of the Dirichlet distribution [9] 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 $E_{D(\Theta_o|\mathrm{a}')}\left(\Theta_o\right) = \int_{\Theta_o} \Theta_o D\left(\Theta_o|\mathrm{a}'\right) d\Theta_o$.

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 an insufficient number of observations to assess trust can easily lead to knowledge defects, resulting in inaccurate trust assessment. To remedy this defect, we consider the impact of imperfect interaction knowledge on trust assessment by exploiting the definition of certainty in [10]. For the multinomial distribution considered in this paper, we further derive its certainty related to knowledge defects $c\left(\mathrm{a}'\right)$ 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 the certainty $c\left(\mathrm{a}'\right)$ is to use mean absolute deviation (MAD) to count the number of increases and decreases from mean value 1 [10]. Given the observed interaction space $\mathrm{a}' = (a'_b, a'_d, a'_u)$ and corresponding probability $\Theta = (\Theta_b, \Theta_d, \Theta_u)$, the certainty $c\left(\mathrm{a}'\right)$ 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 can get the certainty related to knowledge defects $c\left(\mathrm{a}'\right)$ as

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

Based on $E_{D(\Theta_o|\mathrm{a}')}\left(\Theta_o\right) = \frac{\tau_o a_o + \tau_o a'_o}{\sum_{o\in\{b,d,u\}}(\tau_o a_o + \tau_o a'_o)}$ and $c\left(\mathrm{a}'\right)$, the general trust assessment 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 $T_u = 1 - T_b - T_d$.

B. Direct and Indirect Trust Assessment

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 assessment T and certainty related to knowledge defects $c\left(\mathrm{a}'\right)$, we can calculate the direct trust assessment as $DT_{i,j} = (DT_{b,i,j}, DT_{d,i,j}, DT_{u,i,j})$ with the corresponding certainty $c\left(\mathrm{a}'_{i,j}\right)$, where $DT_{b,i,j} = \frac{c\left(\mathrm{a}'_{i,j}\right)\left(\tau_b a_{b,i,j} + \tau_b a'_{b,i,j}\right)}{\sum_{o\in\{b,d,u\}}\left(\tau_o a_{o,i,j} + \tau_o a'_{o,i,j}\right)}$, $DT_{d,i,j} = \frac{c\left(\mathrm{a}'_{i,j}\right)\left(\tau_d a_{d,i,j} + \tau_d a'_{d,i,j}\right)}{\sum_{o\in\{b,d,u\}}\left(\tau_o a_{o,i,j} + \tau_o a'_{o,i,j}\right)}$ and $DT_{u,i,j} = 1 - DT_{b,i,j} - DT_{d,i,j}$.

After obtaining direct trust assessment, the indirect trust assessment $IT_{i\xrightarrow{x}j}$ 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. For fairness and motivation, the common relationship between nodes should be considered in indirect interaction paths. 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

$$\omega_{i,j} = \begin{cases} Dis\,(i,j) \,/\, \max_{i',j' \in \mathcal{G}_g} Dis\,(i',j'), & \text{co-distance,} \\ \Psi, & \text{co-neighbors,} \end{cases} \tag{3}$$

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

$$IT_{i\xrightarrow{x}j} = \begin{cases} DT_{i,x}, & \text{if } \ v == i, \\ DT_{x,j}, & \text{if } \ v == j, \end{cases} \tag{4}$$

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 weights respectively.

So far we have obtained indirect trust assessment $IT_{i\xrightarrow{x}j}$ from multiple interaction paths $i \xrightarrow{x} j$ $(x \in \Psi_{i,j})$. 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 [11]. Following the Dempster-Shafer's rule, the aggregated indirect trust assessment $IT_{i,j}$ can be calculated as

$$IT_{i,j} = \begin{cases} IT_{b,i,j} = \dfrac{\sum\limits_{l_1 \cap l_2 \cdots \cap l_\Psi = \{b\}} IT_{l_1,i\xrightarrow{1}j} \cdots IT_{l_\Psi,i\xrightarrow{\Psi}j}}{1 - \sum\limits_{l_1 \cap l_2 \cdots \cap l_\Psi = \varnothing} IT_{l_1,i\xrightarrow{1}j} \cdots IT_{l_\Psi,i\xrightarrow{\Psi}j}}, \\ IT_{d,i,j} = \dfrac{\sum\limits_{l_1 \cap l_2 \cdots \cap l_\Psi = \{d\}} IT_{l_1,i\xrightarrow{1}j} \cdots IT_{l_\Psi,i\xrightarrow{\Psi}j}}{1 - \sum\limits_{l_1 \cap l_2 \cdots \cap l_\Psi = \varnothing} IT_{l_1,i\xrightarrow{1}j} \cdots IT_{l_\Psi,i\xrightarrow{\Psi}j}}, \\ IT_{u,i,j} = \dfrac{IT_{u,i\xrightarrow{1}j} \cdots IT_{u,i\xrightarrow{\Psi}j}}{1 - \sum\limits_{l_1 \cap l_2 \cdots \cap l_\Psi = \varnothing} IT_{l_1,i\xrightarrow{1}j} \cdots IT_{l_\Psi,i\xrightarrow{\Psi}j}}. \end{cases} \tag{5}$$

C. Final Trust Assessment

To sum up, the final trust assessment $T_{i,j}$ from node i to node j can be further obtained by combining $DT_{i,j}$ and $IT_{i,j}$ based on the Dempster-Shafer's rule, which can be expressed as

$$T_{i,j} = \begin{cases} T_{b,i,j} = \frac{DT_{b,i,j} IT_{b,i,j} + DT_{b,i,j} IT_{u,i,j} + DT_{u,i,j} IT_{b,i,j}}{1 - DT_{b,i,j} IT_{d,i,j} - DT_{d,i,j} IT_{b,i,j}}, \\ T_{d,i,j} = \frac{DT_{d,i,j} IT_{d,i,j} + DT_{d,i,j} IT_{u,i,j} + DT_{u,i,j} IT_{d,i,j}}{1 - DT_{b,i,j} IT_{d,i,j} - DT_{d,i,j} IT_{b,i,j}}, \\ T_{u,i,j} = \frac{DT_{u,i,j} IT_{u,i,j}}{1 - DT_{b,i,j} IT_{d,i,j} - DT_{d,i,j} IT_{b,i,j}}. \end{cases} \tag{6}$$

Similarly, substituting the total interaction outcome $\mathrm{a}_i = (a_{b,i}, a_{d,i}, a_{u,i})$ into the general trust assessment T and certainty $c(\mathrm{a}')$, the trust assessment of node i can be calculated as T_i, i.e., $T_i = (T_{b,i}, T_{d,i}, T_{u,i})$, where $T_{b,i} = \frac{c(\mathrm{a}'_i)(\tau_b a_{b,i} + \tau_b a'_{b,i})}{\sum_{o \in \{b,d,u\}} (\tau_o a_{o,i} + \tau_o a'_{o,i})}$, $\frac{c(\mathrm{a}'_i)(\tau_d a_{d,i} + \tau_d a'_{d,i})}{\sum_{o \in \{b,d,u\}} (\tau_o a_{o,i} + \tau_o a'_{o,i})}$ and $T_{u,i} = 1 - T_{b,i} - T_{d,i}$.

IV. Trust Decision in Blockchain

Through the trust assessment, an effective collaborative supervision can be realized first, and then trust decisions can be made to encourage nodes to participate in the blockchain process normally.

A. 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, leader election brings a concern that abnormal nodes may pose threats to consensus process. To ensure randomness and democracy, any node can start 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 behaviors 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 [5] and Raft [12] 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 [13], 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. Different with randomized leader election based on Raft and PBFT, the decision to elect a trusted leader can be made according to the following majority rule:

$$v_j = \underset{i,j \in \mathcal{G}_g}{\text{majority}}(T_{i,j}, v_{i,j}), \tag{7}$$

where $v_{i,j}$ is the voting strategy related to the trust assessment $T_{i,j}$ from the i-th follower to the j-th candidate leader, and v_j is the number of votes won by the j-th candidate leader. As such, the candidate leader with majority votes can be elected as a leader node.

In Raft, 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 $t \in [t_1, t_1 + T_{d,j}(t_2 - t_1)]$. Obviously, this simple way makes highly-trusted nodes have the larger probability to be candidate nodes, while ensuring randomness. In summary, the followers start the leader election process based on the following steps:

Step 1: If any follower does not receive heartbeats from leader 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 assessment, each node votes with a ternary-opinion $\langle 1, 0, -1 \rangle$, expressed as

$$v_{i,j} = \begin{cases} 1, & \text{if } T_{i,j} \geq \tau, \\ 0, & \text{if } T_{i,j} < \tau, \\ -1, & \text{if validation fails}, \end{cases} \tag{8}$$

where τ is a trust assessment threshold, which can be determined by the average trust level over the consensus group.

Step 3: After 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 selects some transactions according to descending order of transaction publishers' trust $T_{i,b}$, and packetizes them into a candidate block.

B. Block Validation

To include a candidate block 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 integrity of the candidate tips, while executing trust assessment for the valid candidate tips and sorting them in descending order of trust.

Stage 3: Next the candidate block chooses the top-k tips with the highest trust assessment from the valid 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 till until cumulative weight reaches a defined threshold. It is worth noting that a set of visible tips should be determined for trust assessment and block validation. To improve the diversity and freshness of trust, we regard the tips that the timestamp of tips plus the maximum visible timespan does not exceed the current time as the visible tips. Based on such mechanism, the tips of highly-trusted nodes can be assessed and validated by more nodes within such a maximum visible timespan, while the tips of abnormal nodes can be isolated due to the less selection.

V. Numerical Results

In this section, we validate the effectiveness of the proposed TPM in blockchain-assisted MCS and evaluate several critical metrics, including contribution rate, consensus accuracy, transaction throughput and tips stability.

A. Experimental Settings

We consider 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. Considering the existence of untrusted nodes, abnormal nodes may poison transaction data by forging data from other nodes, so as to publish malicious transactions and blocks. Since abnormal nodes may behave normally to defraud trust, we assume that abnormal nodes publish malicious transactions or blocks with a probability p. In this paper, we set p to $2/3$ and the number of abnormal nodes to 30.

In the process of TPM in blockchain-assisted MCS, we set the rate at which each node publishes transactions to 1/2 transactions per unit time. To ensure that the tips from trustworthy nodes get more approvals, we set the maximum visible timespan to 20. In the visible timespan, the new incoming blocks should select 10 tips to authenticate, and two of them will be referenced by incoming blocks. In addition, the SHA-256 hash function is used to generate data hash in this paper.

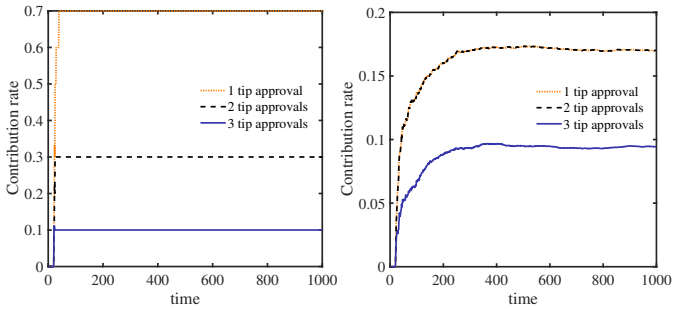
B. Performance Comparisons

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-BlockDAG: In [8], the authors propose to use blockchain to record historical trust information. To represent the trust, the Dirichlet distribution is adopted to classify the behaviors of participants into several ranks and use it as the trust assessment.
- Poof of reputation-based BlockDAG (PoR-BlockDAG): In [14], the authors propose a reputation-based consensus protocol to promote successful interaction. Essentially, PoR in this paper uses the sigmoid function to assess trust and elects a leader who has the highest trust assessment.
- Poof of work-based BlockDAG (PoW-BlockDAG): It is an original PoW-based BlockDAG without relying on a trust/reputation-based incentive mechanism [15].

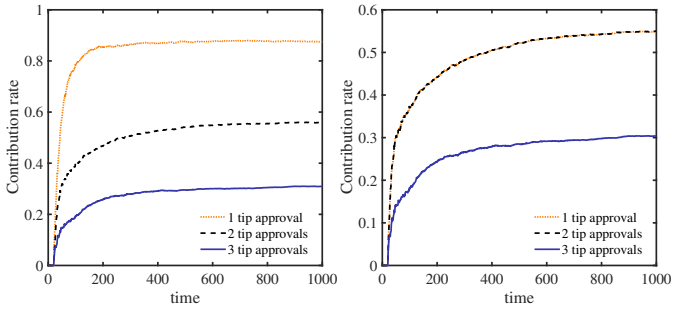
1) Impact of the number of tip approval times on contribution rate: In the process of TPM BlockDAG, we regard the blocks whose tip approval times are less than a certain number as untrusted blocks, and these blocks will be considered isolated without any contribution. Therefore, the contribution rate is evaluated by the proportion of transactions that are approved above a threshold in the total number of published transactions.

As shown in Fig. 2, we can observe that the contribution rate of abnormal leaders and publishers decreases with the number of tip approvals, where the thresholds for the number of tip approvals is set to be 1, 2 and 3, respectively. On the one hand, increasing the threshold for tip approvals makes it more difficult to approve tips from abnormal 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 stable over time, which indicates that the blocks and transactions published by abnormal nodes can be isolated as



(a) Abnormal leaders. (b) Abnormal publishers.

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



(a) Normal leaders. (b) Normal publishers.

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

much as possible in the case of 3 tip approvals. Furthermore, Fig. 3 illustrates the contribution rate of normal leaders and publishers. Obviously, 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.

2) Consensus accuracy comparisons: This experiment evaluates the consensus accuracy, which measures the proportion of normal node in the total number of elected nodes.

Fig. 4 demonstrates the consensus accuracy under abnormal behaviors. Obviously, TPM-BlockDAG significantly outperform other schemes in terms of the 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 trust assessment to motivate normal nodes to be elected as leaders and punish abnormal nodes to some extent, but the proposed TPM can more accurately and comprehensively characterize the trust relationship of nodes. In addition, PoW-BlockDAG chooses a leader randomly, resulting in significantly lower consensus accuracy under malicious and lazy behaviors.

VI. Conclusions

In this paper we have proposed to integrate TPM in blockchain-assisted MCS to solve the problem of off-chain data authenticity and on-chain interactions trustworthiness. The TPM infers trust level of participating nodes for characterizing and motivating the underlying interactions in blockchain-assisted MCS. To achieve trust supervision and promote successful interaction, trust decision is made followed by trust assessment, so as to filter some abnormal node and avoid participating in the blockchain process. The experimental results demonstrate that the proposed TPM can help blockchain resist abnormal behaviors and outperform trust/reputation based blockchains, as well as the blockchain without trust.

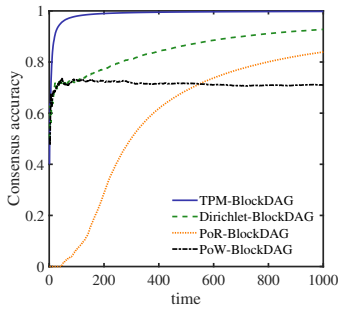


Fig. 4: Consensus accuracy.

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] Y. Zhan, P. Li, K. Wang, S. Guo, and Y. Xia, "Big data analytics by crowdlearning: Architecture and mechanism design," *IEEE Network*, vol. 34, no. 3, pp. 143–147, 2020.

[3] J. Chen, H. Ma, D. Zhao, and L. Liu, "Correlated differential privacy protection for mobile crowdsensing," *IEEE Transactions on Big Data*, vol. 7, no. 4, pp. 784–795, 2021.

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

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

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

[13] 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

[14] 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.

[15] Y. Sompolinsky, Y. Lewenberg, and A. Zohar, "Spectre : Serialization of proof-of-work events : Confirming transactions via recursive elections," 2017.