drones

MDPI

Article

A Hierarchical Blockchain-Based Trust Measurement Method for Drone Cluster Nodes

Jinxin Zuo [1,2], Ruohan Cao [2,3,*], Jiahao Qi [1,2], Peng Gao [2,3], Ziping Wang [1,2], Jin Li [1,2], Long Zhang [1,2] and Yueming Lu [1,2]

[1] School of Cyberspace Security, Beijing University of Post and Telecommunications, Beijing 100876, China; zuojx@bupt.edu.cn (J.Z.); qijiahao@bupt.edu.cn (J.Q.); wangziping2022@bupt.edu.cn (Z.W.); li_jin@bupt.edu.cn (J.L.); zhang-long@bupt.edu.cn (L.Z.); ymlu@bupt.edu.cn (Y.L.)
[2] The Key Laboratory of Trustworthy Distributed Computing and Service (BUPT), Ministry of Education, Beijing 100876, China; gp@bupt.edu.cn
[3] School of Information and Communication Engineering, Beijing University of Post and Telecommunications, Beijing 100876, China
* Correspondence: caoruohan@bupt.edu.cn

Abstract: In response to the challenge of low accuracy in node trust evaluation due to the high dynamics of entry and exit of drone cluster nodes, we propose a hierarchical blockchain-based trust measurement method for drone cluster nodes. This method overcomes the difficulties related to trust inheritance for dynamic nodes, trust re-evaluation of dynamic clusters, and integrated trust calculation for drone nodes. By utilizing a multi-layer unmanned cluster blockchain for trusted historical data storage and verification, we achieve scalability in measuring intermittent trust across time intervals, ultimately improving the accuracy of trust measurement for drone cluster nodes. We design a resource-constrained multi-layer unmanned cluster blockchain architecture, optimize the computing power balance within the cluster, and establish a collaborative blockchain mechanism. Additionally, we construct a dynamic evaluation method for trust in drone nodes based on task perception, integrating and calculating the comprehensive trust of drone nodes. This approach addresses trusted sharing and circulation of task data and resolves the non-inheritability of historical data. Experimental simulations conducted using NS3 and MATLAB demonstrate the superior performance of our trust value measurement method for unmanned aerial vehicle cluster nodes in terms of accurate malicious node detection, resilience to trust value fluctuations, and low resource delay retention.

Keywords: drone cluster; hierarchical blockchain; resource constraint; trust measurement

Citation: Zuo, J.; Cao, R.; Qi, J.; Gao, P.; Wang, Z.; Li, J.; Zhang, L.; Lu, Y. A Hierarchical Blockchain-Based Trust Measurement Method for Drone Cluster Nodes. *Drones* **2023**, *7*, 627. https://doi.org/10.3390/drones7100627

Academic Editor: Shiva Raj Pokhrel

Received: 24 August 2023
Revised: 30 September 2023
Accepted: 4 October 2023
Published: 8 October 2023

1. Introduction

In recent years, drone clusters have found increasingly widespread applications in both military and civilian fields, such as collaborative reconnaissance, real-time monitoring, and aerial base stations [1]. Unmanned aerial vehicle mobile ad hoc networks (UAV MANETs) represent a common communication network for drone clusters, which operate independently of ground-based communication infrastructure and establish distributed networks among the UAVs [2]. The trust degree of UAV nodes is an important influence point for drone cluster mission coordination, independent decision-making, and independent judgment [3]. The trust value of UAV nodes can not only help to select suitable UAVs to participate in tasks but also support resource allocation and task scheduling. However, when the drone cluster is far away from the management platform, facing a variety of network attacks and information interference, the UAV nodes may face cross-cluster reorganization, high dynamic entry and exit of trust clusters, etc. [4]. The trustworthiness inheritance of dynamically entering and exiting nodes and the trust re-evaluation of dynamic clusters become crucial.

The rapid changes in the topology of UAV MANETs result in the instability of communication delay and link status [5,6]. UAV nodes cannot continuously monitor neighbor nodes, but the comprehensive trust degree of UAV nodes can be calculated through the combination of node direct trust evaluation and recommended trust evaluation. Blockchain technology possesses characteristics of decentralization, immutability, and traceability, enabling secure data sharing. In the context of trust management mechanisms, blockchain can be employed to record the trustworthiness information of UAV nodes within the network. However, as the number of UAVs in the network increases, the blockchain also grows, leading to increased communication latency and operational burdens on the network [7]. In order to reduce the propagation delay of blocks and the communication overhead in UAV MANETs, a hierarchical blockchain approach is adopted for trustworthy historical data flow and inheritance [8]. This allows for cross-domain data sharing among UAVs, enhances the credibility of data transmission, improves the accuracy of UAV node trust measurement, supports autonomous mission coordination and collaborative decision-making capabilities, and ensures reliable task completion.

Within the realm of UAV trust measurement research, the literature [9] employs data packet forwarding history and recommendation information from neighboring nodes to calculate the trustworthiness of other nodes. However, this trust mechanism exhibits a limitation in that it relies on a single trust assessment factor and cannot effectively address issues such as link failures. Another study [10] proposed a secure clustering scheme based on fuzzy classification trust, utilizing multi-criteria for classification and optimization to assess node trust. However, this scheme incurs higher energy consumption. Additionally, it assumes a cluster-level hierarchical structure for the network, which results in a higher probability of network failure. Alternatively, the literature [11] introduces a random repeat trust computation scheme that takes into account remaining energy and channel quality. This scheme directly specifies the weight for trust evaluation factors and computes node trust using a weighted average.

Further, the literature [12] adopts a fuzzy C-Means clustering approach to categorize nodes into three groups. Simultaneously, it adjusts the evaluation factor weights and calculates trust degrees based on the clustering centers, thereby rewarding or penalizing nodes within the cluster. Addressing security concerns, the literature [13] employs fuzzy variables to characterize attributes associated with black holes, flooding, and packet discarding attacks prevalent in ad hoc networks. A fuzzy Petri network divides node trust levels into five tiers, facilitating fuzzy security verification among nodes. In terms of trust aggregation, an adaptive fuzzy trust aggregation network is proposed by the literature [14] to compute node trust evaluations. This approach integrates trust factors such as the data packet forwarding rate, trusted interaction degree, and detection packet receiving rate to calculate a node's direct trust degree. It further combines the indirect trust degree of trusted neighboring nodes to formulate a comprehensive trust degree.

The literature [15] establishes trust values by comparing a node's message forwarding count with its energy consumption rate and presents a distributed trust management scheme. Nonetheless, this approach incurs a notable communication overhead and poses a considerable trust acquisition cost. In contrast, the literature [16] devises the provenance-based trust model framework, which traces the source to achieve accurate point-to-point trust evaluation within resource-constrained network settings. This framework seeks to optimize correct message delivery to the intended node while minimizing message delay and communication costs. Yet, it faces challenges concerning algorithm efficiency and the precision of trust recognition.

The above research has laid a good foundation for the research on the trust evaluation model of UAV nodes, but some of the literature ignores the influence of trust timeliness on trust evaluation in the direct trust degree calculation [12,13], which can not reflect well on node behavior, which reduces the accuracy of trust calculation. The trust evaluation of UAVs in the other literature still has shortcomings such as the single trust evaluation factor [9,15], relatively fixed trust evaluation weight [11], high communication

overhead [10,14,15], unreasonable trust evaluation [16], etc. Furthermore, while much of the current research is centered around assessing the reputation of UAVs, there is a notable lack of attention directed toward establishing robust mechanisms for credible reputation management. Table 1 summarizes the main contributions and limitations of the aforementioned literature.

Table 1. Comparison of research work.

Literature	Research Scheme	Main work	Disadvantage
[9]	Light-weight trust-quality routing protocol	Calculate trust values based on historical information in the network	Trust evaluation factor is single
[10]	Fuzzy classification trust-based secure clustering scheme	Multi-criteria for classification and optimization to evaluate nodes' trust	Communication overhead is large and high network failure rate
[11]	Random Repeated Trust Computing Trust Management	Further considers the weight of multiple trust evaluation factors such as residual energy and channel quality	Trust evaluation weight is fixed
[12]	A New Trust Model Based on Fuzzy Logic	Adjust the evaluation factor weights according to the cluster center and calculate the trust degree	The timeliness of trust is ignored, and the packet loss rate is high
[13]	Secure Mobile Ad Hoc Network Routing Protocol FPN-SAODV	Divide trust levels to realize fuzzy security verification between nodes	The timeliness of trust is ignored, and the packet loss rate is high
[14]	Adaptive Fuzzy Trust Aggregation Network	Multiple trust evaluation factors, introducing trust fluctuation penalty factors, and correcting node trust	Long network delay and high communication overhead
[15]	Decentralized trust management scheme DTMS	Comparing the relationship between the forwarding number of the node and the energy consumption rate to obtain the trust value	Trust evaluation factor is single, and the communication overhead is large
[16]	A trust model based on traceability-PROVEST	Reversely infer the trust value of the message-generating node or operating node according to the integrity of the message	Only use the trust value of the predecessor node to judge whether the task message has been tampered

To counter the potential threat of malicious nodes targeting UAVs during data collection and transmission, Ge et al. [17] have proposed a distributed solution using blockchain technology. This approach introduces a novel lightweight blockchain architecture, which

reduces computational and storage costs while simultaneously offering advantages in terms of privacy and security. A blockchain-powered trusted drone cluster communication scheme was devised by [18]. This scheme employs attribute-based algorithms for UAV authentication, verifying and adding data to the blockchain based on the attributes of the UAVs. However, this approach faces the challenge of significant underlying blockchain communication latency. The literature [19] employs a hierarchical blockchain approach with main and sub-chains to address the issue of secure identity verification for drone clusters across trust domains and diverse network environments. However, this method's efficiency for authentication is affected to some extent as the number of authentication nodes increases, leading to an increase in data transmission volume. A distributed key management solution tailored for heterogeneous UAVs was introduced by [20], leveraging the capabilities of blockchain technology. With this scheme, UAVs can autonomously distribute cluster keys, update their public-private key pairs, migrate between clusters, and securely revoke malicious UAVs. However, there is a limitation in this approach as it is challenging to detect malicious behavior in the head node of UAVs.

Through research on the above literature, there are still some problems and challenges in the combination of blockchain technology and UAVs. The drone cluster network is typically organized into communication clusters based on a hierarchical structure. Data exchange between distinct clusters is facilitated through the cluster head node responsible for forwarding the information. However, the peer-to-peer broadcast approach inherent in blockchain applications can engender excessive inter-cluster bandwidth consumption. This, in turn, could potentially trigger network congestion issues at the cluster head nodes. PoW (Proof of Work), PoS (Proof of Stake), and PBFT (Practical Byzantine Fault Tolerance) are the most common consensus algorithms in the blockchain. However, these mechanisms do not take into account the resource constraints of UAVs, such as storage and computing power; therefore, developing efficient consensus algorithms is another challenge for drone clusters based on blockchain technology. In addition, UAVs cannot store blockchains that grow due to large-scale drone networks, due to storage capacity limitations. The PoW consensus mechanism can only process several transactions per second, and the block confirmation delay is about 60 min [21]. The adoption of entrusted Proof of Stake (DPoS) and PBFT consensus mechanisms can improve the consensus speed of blockchain systems. However, these consensus mechanisms require frequent interaction of devices and require a lot of communication resources. At the same time, only individual nodes are allowed to maintain the blockchain, and the distributed performance is low [22,23]. Reference [24] adopts the structure of a two-layer blockchain consensus to achieve the purpose of verifying multiple blocks at the same time and improve the efficiency of the blockchain chain. However, the dual-layer structure adopts a PoW consensus, which requires too much computing power resources for wireless devices. The PBFT-PoW two-layer consensus structure in our paper adopts a PBFT consensus in a small range within the cluster, which can improve the efficiency of the blockchain and reduce the computing power requirements of the system.

Within the framework of trust management, the blockchain functions as a reliable ledger for capturing and consolidating node perspectives. In the work of [25], the blockchain serves as a repository for recording node viewpoints contributed by miners, thus furnishing the necessary data for trust assessment. However, the scheme uses the traditional PoW consensus algorithm, and the drone cluster has high network latency and computational complexity. The literature [26] studied a blockchain-based vehicle-to-everything (BIoV) system, in which roadside units (RSUs) act as miners in the network and generate blocks to record data transmission between vehicles. To prevent data tampering between RSUs and malicious vehicles, the system selects the miners who generate the blocks through voting among all vehicles. In the context of trust management, the work presented in [27] introduces a trust management framework based on a series of blockchain timestamps, where the behavior of these unmanned aerial vehicles is periodically monitored by a group of distributed observers (DOs). DOs calculate relative trust scores for each UAV and record these scores in a transparent and secure ledger. In the context of enhancing internal security

for UAVs, a blockchain-based trusted self-organizing network mechanism was devised and denoted as BC-UTSON, as outlined in Reference [28]. This pioneering work utilizes Practical Byzantine Fault Tolerance (U-PBFT) to ensure a lightweight consensus and real-time full-node trustworthiness assessment within the layered self-organizing network structure of UAV clusters. It also designs a blockchain-based multi-weight subjective logic (BMWSL) scheme to identify malicious UAV nodes based on trustworthiness assessments. Furthermore, this scheme augments the trusted path quality awareness through the Dynamic Routing mechanism, effectively thwarting data from being inadvertently routed through compromised nodes. While the blockchain serves as a credible ledger to record opinions, the evaluation outcomes could still be influenced by misleading information. To counter this, the scheme employs a blockchain consensus to neutralize the impact of misleading information interference, ultimately achieving higher evaluation accuracy and sensitivity.

Throughout the design of the above approaches, it is imperative to consider the balance between block generation speed and block propagation delay among users. Delays in updating user reputation values can lead to prolonged difficulties in identifying malicious users, consequently elevating the vulnerability of other users to potential attacks.

Given the above-mentioned deficiencies and problems, this paper proposes a trustworthy evaluation management model for drone clusters based on hierarchical blockchain. The main contributions of this paper are listed below.

(1) To address the issues of a single trust evaluation factor and inadequate trust computation in the assessment of trustworthiness for UAV nodes, this paper proposes a UAV node dynamic trust evaluation method, which utilizes multiple trust factors. The model incorporates a dynamic trust decay factor to effectively leverage the historical trustworthiness of UAV nodes. It employs an information entropy-based optimization method for calculating the trustworthiness weight and computes the reputation value for each node through a reputation fusion algorithm.

(2) To address the need for real-time trust evaluation of newly joined nodes during the dynamic composition of drone formations, we propose a trust management mechanism for UAVs based on a hierarchical blockchain. According to the characteristics of a cross-domain combination of UAVs, the layered blockchain is used to manage and record its trust value to improve the efficiency of the blockchain's transaction validation process. Additionally, considering that regular UAV nodes may have limited computational power, we leverage the high computational power of cluster head nodes to offload the consensus validation tasks of UAV nodes, reducing the consensus latency and enhancing the overall efficiency of the blockchain system.

(3) We build a simulated drone cluster scene through the NS3 and MATLAB simulation platform, install the reputation value evaluation algorithm on the UAV node, and compare it with other existing schemes to verify the effectiveness of the evaluation algorithm in this paper. By comparing the blockchain delays of different resource allocation schemes, the efficiency of the blockchain system scheme in this paper is verified.

The rest of the paper is organized as follows. Section 2 introduces the hierarchical blockchain-based trust measurement model for drone cluster nodes proposed in this paper. Section 3 verifies the effectiveness of the algorithm through experiments. Section 4 gives the discussion and future work and Section 5 concludes this paper.

2. Trust Metric Model of Unmanned Aerial Vehicle Cluster Nodes Based on Hierarchical Blockchain

2.1. The Proposed Model

UAV swarms have the characteristics of flexible self-organization of network topology and high dynamics of task data changes. UAVs can replace manual tasks through mutual coordination and information sharing by forming a network. Applications are becoming more and more intelligent. UAV formations often dynamically cluster when performing tasks, and work efficiency is improved through multi-cluster cooperative execution of tasks.

However, when UAV cluster nodes dynamically combine formations, the high dynamics of nodes entering and leaving the trust cluster lead to low accuracy of node trust evaluation. The typical application environment of the UAV network is shown in Figure 1. There are multiple trust management domains in the environment. All UAV nodes belonging to the same task domain form a single trust management domain. A single management domain is controlled by the trust management node TMN_x and other common nodes N_x.

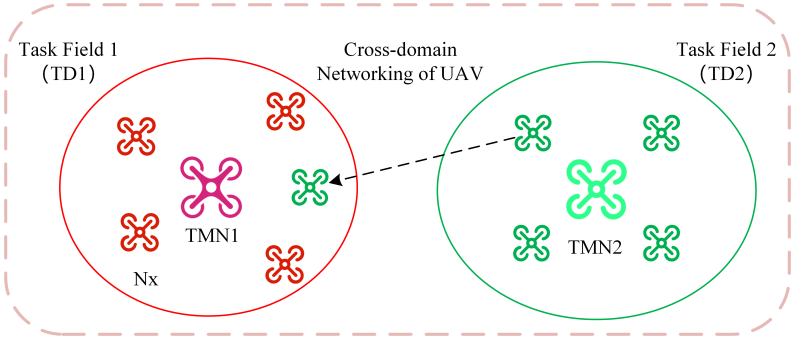

Figure 1. Typical application environment of UAV network.

Aiming at the problem that UAV cluster nodes are highly dynamic in and out of trust clusters, resulting in low accuracy of node trust assessment, this paper proposes a trust measurement model for UAV cluster nodes based on hierarchical blockchain, as shown in Figure 2. All nodes in a single trust management domain jointly maintain a first-level blockchain, and trust management nodes in all management domains maintain a second-level blockchain. Each drone node in the environment has a unique identifier, and for the drones in the domain, the trust relationship of nodes is maintained by the management node, and the transfer of trust between domains is guaranteed by the secondary blockchain. Based on building a layered blockchain, this model ensures the credible evaluation, sharing, and management of the trust degree of UAV nodes. A trusted trust management model based on a layered blockchain includes a demand analysis layer, evaluation layer, and management layer.

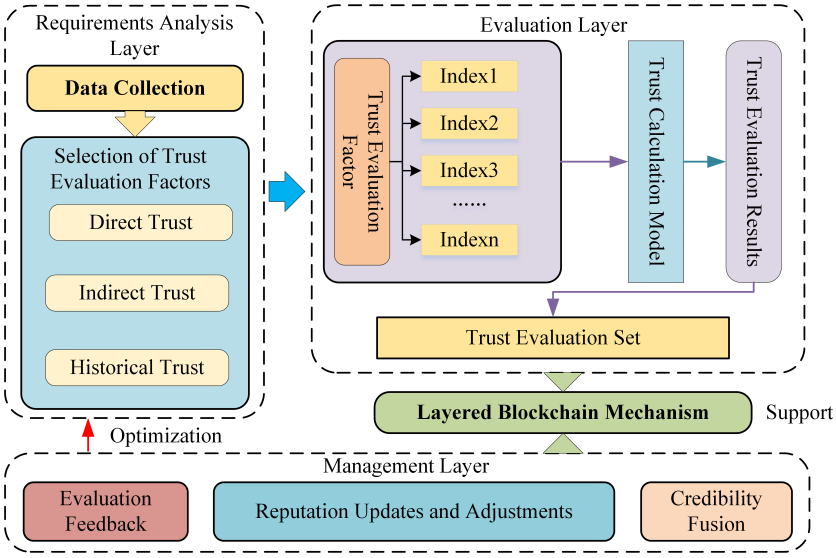

Figure 2. Trust metric model of unmanned aerial vehicle cluster nodes based on hierarchical blockchain.

The demand analysis layer selects the node trust evaluation factor according to the characteristics of the UAV group itself and flexibly selects the appropriate credibility evaluation factor and calculation method, which can meet the needs of actual application

scenarios. This paper uses direct trust, indirect trust, and historical trust to design evaluation factors for node credibility, and evaluates the overall behavior of nodes during the monitoring cycle.

The evaluation layer is the core layer of node credible trust evaluation. Based on the trust evaluation factors determined by the demand analysis layer, the direct trust degree of the node is first calculated by using the linear weighting method. Considering that the evaluation of the direct trust degree mainly comes from its own experience, which is subjective, and through the indirect adoption of other entities, experience can reduce the influence of subjective evaluation, and then use the recommendation of neighbor nodes to calculate the indirect trust degree of the node to be evaluated. Then, considering the timeliness of the historical trust degree, this paper proposes to express the timeliness of trust based on the dynamic trust decay factor, and finally calculate the comprehensive trust degree of nodes by combining the direct trust degree, indirect trust degree, and historical trust degree.

The management relies on the layered blockchain mechanism to manage the trust value, which mainly includes the record of the historical trust degree, the integration of the trust degree, and the update and adjustment of the trust degree. The accuracy and reliability of the model can be improved through validation and feedback in real situations. By comprehensively considering multiple factors and adopting appropriate data processing and model methods, a more accurate and reliable calculation model of UAV trust degree can be established.

2.2. A Dynamic Evaluation Method for Drone Node Trust Based on Task Perception

The trust level associated with nodes within an unmanned swarm denotes the assessment made by a given node about the prospective service capability or likelihood of cooperative conduct from its neighboring nodes in the future. This estimation is quantified as a trust value, representing the quantitative expression of the evaluating node's anticipation, regarding the likelihood of cooperation by the neighboring node. Of particular note is the direct trust value, which signifies the expected probability of cooperation and service proficiency of a specific entity. This value is determined by amalgamating the historical data of direct interactions between the evaluating node and the subject in question. This direct trust value from node i to node j is denoted as T_{ij}^{d}.

Complementing this, the recommended trust degree stands as a projection of the targeted node's anticipated service capacity or probability of cooperation. This projection hinges on the trust value bestowed upon the target node by other participating nodes. The direct trust degree attributed from node i to node j is recorded as T_{ij}^{re}.

2.2.1. Direct Trust Evaluation

The calculation method of the direct trust degree is to evaluate the trust degree of the node by periodically observing the node behavior and combining the four trust evaluation factors of packet loss rate, data packet forwarding rate, trusted interaction degree, and detection packet reception rate. The definition of each impact factor is as follows.

Packet loss rate (PLR) refers to the ratio of the total number of lost data packets to the total number of sent data packets.

Packet forwarding rate (PFR) refers to the ratio of packets received by a node to those it forwards.

Trustworthiness of interaction (TOI) refers to the degree of interaction between a node and other nodes. The number of interactions between a trusted node and an untrusted node can reflect the degree of trustworthiness of the node. When a node interacts with more trusted nodes and interacts less with untrusted nodes, the degree of trusted interaction will be lower. The weight of the trust evaluation factor can be calculated by the AHP and the Delphi method. Here, we use the Delphi method to calculate the weight of the trust factor of the direct trust degree. Then, after obtaining the actual data of the trust factor, the linear weighting method is used to calculate the direct trust value of the UAV node.

2.2.2. Recommendation Trust Evaluation

The recommendation trust degree refers to the direct trust value recommended by the neighbor node set shared by node i and node j. By synthesizing the recommendation information of public nodes, it can reflect the recommended trust level more reliably, truly, and accurately. The formula for calculating the recommended trust value T_{ij}^{re} is

$$T_{ij}^{re} = \frac{\sum_{k=1}^{n-1} T_{ik}^{d} \times T_{kj}^{d}}{\sum_{k=1}^{n-1} T_{ik}^{d}}. \tag{1}$$

2.2.3. Comprehensive Trust Calculation and Update

(1) Integrated trust calculation

The calculation of comprehensive trustworthiness requires the fusion of direct trust and indirect trust. Evaluating a node's trustworthiness involves considering its historical service trust sequence $T\left(T^{t_1}, T^{t_2}, \ldots, T^{t_n}\right)$, with historical trust records decaying based on dynamic trust factors. Setting the weights of direct trust and recommended trust for a node based on empirical knowledge would introduce subjectivity into the calculation of comprehensive trustworthiness. Information entropy, as a method to measure the utility values of various indicators, can be used to determine the corresponding weights of the indicators to overcome the limitations of empirically determining weights.

The calculation of the historical trust value decreases with the increase in time, and the record of the recent trust value can better reflect the cooperation probability of nodes. Therefore, the timeliness of trust should also be taken into account when calculating direct trust.

In this paper, the dynamic trust attenuation factors that present the timeliness of trust can be expressed as

$$FR(\lambda, t_k) = e^{-\lambda \cdot L(t - t_k)}, \tag{2}$$

where λ and $L(t - t_k)$ are two independent variables. λ is the rate adjustment factor, and $0 < \lambda \leq 1$. The value of λ can be adjusted according to the specific application scenario. $L(t - t_k)$ is the time update function, representing the time elapsed from the occurrence of the k-th historical trust record to the current time t. t_k denotes the moment when the trust value of the node is evaluated for the k-th time.

The comprehensive trustworthiness value T_j^{total} of node j is calculated using

$$T_j^{total} = w_d \cdot T_{ij}^{d} + w_{re} \cdot T_{ij}^{re} + \sum_{i=1}^{n} FR(\beta_i, t_i) \cdot T^{t_i}, \tag{3}$$

where w_d and w_{re} are the adaptive weights of direct trust and recommended trust, respectively. The calculation method can be expressed as

$$w_d = \frac{1 - \left(\frac{H\left(T_{ij}^{d}\right)}{lb\ T_{ij}^{d}}\right)}{\left[1 - \left(\frac{H\left(T_{ij}^{d}\right)}{lb\ T_{ij}^{d}}\right)\right] + \left[1 - \left(\frac{H\left(T_{ij}^{re}\right)}{lb\ T_{ij}^{re}}\right)\right]}, \tag{4}$$

and

$$w_{re} = \frac{1 - \left(\frac{H\left(T_{ij}^{re}\right)}{lb\ T_{ij}^{re}}\right)}{\left[1 - \left(\frac{H\left(T_{ij}^{d}\right)}{lb\ T_{ij}^{d}}\right)\right] + \left[1 - \left(\frac{H\left(T_{ij}^{re}\right)}{lb\ T_{ij}^{re}}\right)\right]}, \tag{5}$$

where $H\left(Tij^{d}\right)$ and $H(Tij^{re})$ are the information entropy of direct trust and recommended trust, respectively.

(2) Trust renewal

When the UAV group reaches the trust update time, the trust record is updated according to the following rules.

a. The evaluation node first calculates the direct trust degree and indirect trust degree of the node to be evaluated.
b. The evaluation node queries the historical trust degree of the node to be evaluated from the first-level blockchain of the trust management area.
c. The evaluation node calculates the comprehensive trust based on direct trust, indirect trust, and historical trust.
d. After evaluating the trust of all surrounding nodes, the trust record is stored in the tier-1 blockchain.

(3) Trust calculation process

The pseudo-code flow of the evaluation algorithm can be expressed as in Algorithm 1.

Algorithm 1 Trust Evaluation Algorithm

Input: Number of nodes, heartbeat interval
Output: the Latest fused trustworthiness of nodes

Initialize parameters θ
Cluster head node C_0 starts the first heartbeat
Ordinary nodes $C = \{C_1, C_2, \ldots, C_t\}$ receive heartbeat from cluster head node and start computing T_{ij}^{d}
for each node C_i in C **do**
    C_i sends a communication packet to C_j
    C_j receives the packet and checks if it needs to be forwarded:
    if forwarding is required **then**
        Forward to the next relay node
    else
        Return response packet
    end if
    C_i receives the response packet from C_j and computes C_j's PLR, PFR, and TOI
    C_i calculates C_j's T_{ij}^{d}
end for
for each node C_i in C **do**
    C_i sends a request packet for recommending trustworthiness of C_j to C_n and C_m
    C_n and C_m receive the request packet and return T_{nj}^{d} and T_{mj}^{d} of C_j
    C_i calculates C_j's T_{ij}^{re} after receiving the response packets
end for
for each node C_i in C **do**
    Calculate the current round's FR
    Calculate T_{ij}^{d} and T_{ij}^{re}
    Calculate C_j's T_{j}^{total}
end for
Package all data and send to C_0
C_0 collects all comprehensive trustworthiness result set T
for each node C_i in C **do**
    Calculate C_i's $P(e_j|C_j)$
end for

2.2.4. Trust Fusion Algorithm

After the UAV cluster head node receives the trust degree evaluation list sent by the common node, it needs to fuse the trust degree evaluation of multiple UAV nodes to the same UAV node into a comprehensive trust degree. This fusion process can be calculated using the following message summary formula based on Bayesian inference, which can more accurately calculate the result of fusion trust. The specific calculation formula can be expressed as

$$P(e_j \mid C_j) = \frac{P(e_j) \times \prod_{k=1}^{q} p\left(c_k^j \mid e_j\right)}{P(e_j) \times \prod_{k=1}^{q} p\left(c_k^j \mid e_j\right) + P(\bar{e}_J) \times \prod_{k=1}^{q} p\left(c_k^j \mid \bar{e}_J\right)}. \tag{6}$$

The cluster head node stores the data in C after receiving all of the credibility lists, such as $C = \{c_1, c_2, c_3, \ldots, c_j\}$. According to the formula, $P(e_j)$ is the prior probability of event e_j occurring. $P(\neg e_j)$ is the prior probability that event e_j does not occur (that is, the prior probability of the complement of event e_j). $P\left(c_j^k \mid e_j\right)$ is the conditional probability that category c_j^k occurs under the condition that event e_j occurs. $P\left(c_j^k \mid \neg e_j\right)$ is the conditional probability of category c_j^k occurring under the condition that event e_j does not occur. q is the number of nodes. We can define the event e_j as the event that the UAV node is evaluated as credible, and the category c_j^k represents the corresponding comprehensive trust evaluation. Given a list of trust evaluations, we can calculate $P(e_j)$ and $P(\neg e_j)$ for each drone node. At the same time, for each evaluation index c_j^k, we can calculate $P\left(c_j^k \mid e_j\right)$ and $P\left(c_j^k \mid \neg e_j\right)$, and substituting these values into the formula, we can calculate that each UAV node is evaluated is the credible posterior probability $P\left(c_j^k \mid e_j\right)$. Then, according to the calculated posterior probability, we can adopt a weighting strategy to fuse the trust evaluations of multiple nodes to obtain a comprehensive trust evaluation for the same UAV node.

2.3. Blockchain Data Sharing Mechanism under Resource Constraints

In this section, we first introduce the collaborative block generation process of the double-layer blockchain. Then, we present the overall system model, including the consensus mechanism process, block structure design, and communication interaction model. Finally, we utilize the computational resources of cluster head nodes to offload the verification tasks of regular nodes. We propose a computational optimization problem to save the computational resources of regular nodes and improve the system's transaction throughput.

2.3.1. Integrated Blockchain and UAV System

Figure 3 illustrates the overall framework of reputation management for UAVs using blockchain. This framework is based on the clustering structure of UAVs, divided into two layers: the first blockchain for the PBFT consensus within clusters and the second blockchain for the PoW consensus between clusters. Let $M = \{1, \ldots, m, \ldots, M\}$ represent the set of clusters, and $U_m = \{1, \ldots, i, \ldots, U_m\}$ represent the set of UAV nodes within cluster m. The first blockchain within clusters adopts the PBFT consensus mechanism. Nodes collect and record interaction data with neighboring nodes, including packet loss rate, packet forwarding rate, trusted interaction degree, and probe packet reception rate. They evaluate the comprehensive trustworthiness of surrounding nodes and send the evaluation data to the consensus leader node. Consensus is reached among nodes within the cluster, and block generation (we call it PBFT-block) is completed. The cluster head node updates the reputation values of nodes within the cluster based on the data in the first blockchain. For the second blockchain between clusters, considering the control requirements for the reputation update cycle and the distance factor between cluster head nodes, the PoW consensus mechanism is adopted to record the collected interaction data within the cluster on the blockchain. In addition to packaging PBFT blocks, the second

blockchain block (we call it PoW-block) also records real-time reputation data of nodes within the cluster to achieve the goal of reputation data sharing among all network nodes.

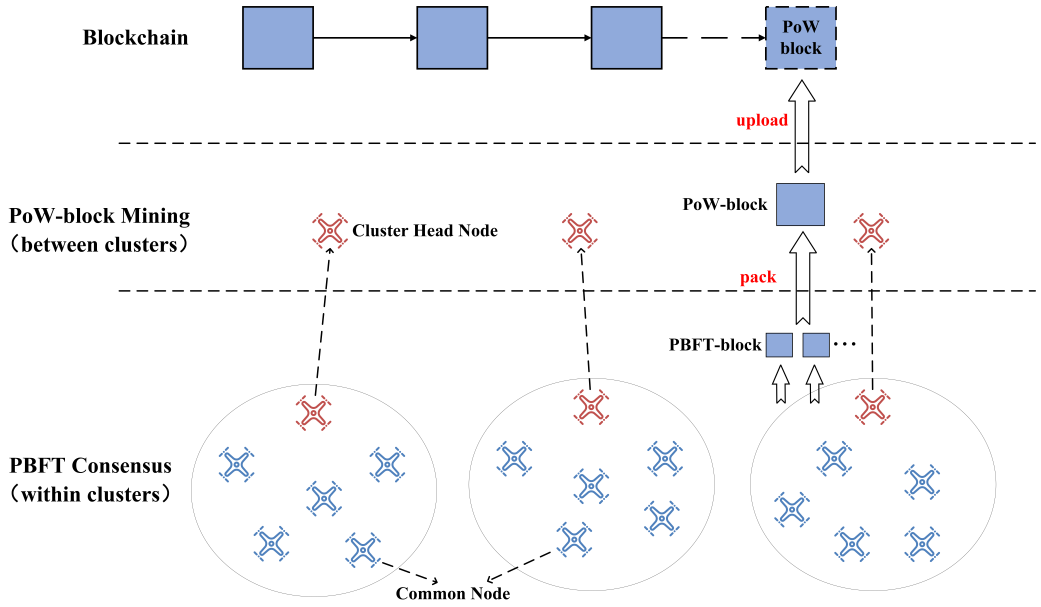

Figure 3. Consensus Framework.

In this framework, we have made improvements to the structure of two types of blocks, namely PBFT-block and PoW-block, to facilitate the integration of a two-tier blockchain. As shown in Figure 4, the structures of the PBFT-block and PoW-block are similar to regular blocks. The header contains the hash, the hash of the previous block, and the timestamp. The body includes the body hash value, the digitally signed signature encrypted with the private key, and the public key, among other key fields. However, for ease of lookup, we have added metadata as a marker in the PBFT-block header, and in the body section of the PoW-block, we have included a list containing multiple PBFT-block metadata. The reputation value data in the PBFT block is hashed to form the trunk unit, and the hash values are successively taken upwards to construct a Merkle tree, with the Merkle root serving as a sub-block in the block header.

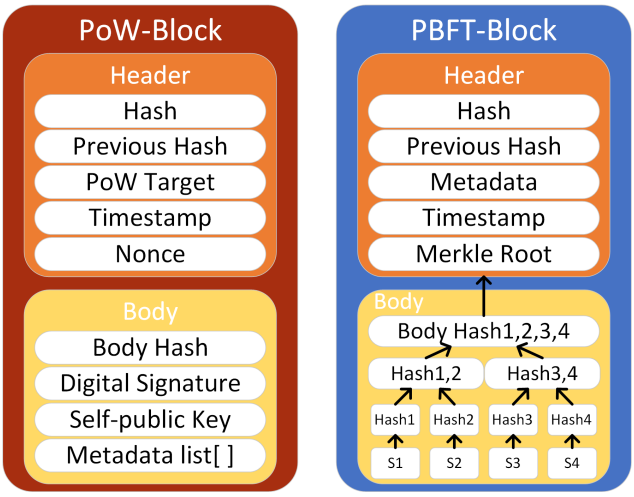

Figure 4. The block structure.

2.3.2. Election of the Consensus Leader Node

During PBFT consensus within each cluster, it is necessary to select a leader node to guide the overall consensus process. To ensure fairness in block generation among nodes

and reduce the threat of malicious nodes, while selecting nodes with higher computational power as leader nodes, this section designs an election mechanism for the consensus leader node. The algorithm is deployed on the cluster head node, and the cluster head node selects the leader node of each blockchain consensus according to the algorithm. Based on the node's own computational power and block generation behavior, the behavior factor RF_i and computational power factor FF_i are proposed to optimize the election mechanism for the leader node in the PBFT consensus of drone clusters. A score reset strategy is also employed for nodes with excessively high or low scores to enhance the decentralization characteristics of consensus.

Behavior factor: $RF_i \in [0,1]$, the initial value is uniformly set to r. Let $RF_i(t)$ denote the reputation value of node i in the t round of consensus. The calculation formula for the reputation value $RF_i(t+1)$ of node i in the $t+1$ round of consensus can be expressed as

$$RF_i(t+1) = \begin{cases} RF_i(t) + \alpha(1 - RF_i(t)), & normal \\ \beta RF_i(t), & abnormal \\ RF_i(t)e^{-\lambda \Delta b}, & offline \\ 0, & byzantine \end{cases}. \tag{7}$$

Normal behavior includes block-producing nodes packaging valid blocks, the leader node leading all network nodes to reach consensus, and the slave nodes participating in consensus and eventually synchronizing their results with the majority of nodes. Abnormal behavior includes block-producing nodes packaging invalid blocks, the master node failing to produce blocks, and slave nodes synchronizing with results that differ from the majority of nodes. Offline nodes refer to nodes that do not participate in consensus, and their reputation gradually decreases over time. Byzantine nodes refer to nodes that send inconsistent messages to different nodes. Among them, the coefficient $\alpha \in (0,1)$ is used to control the growth rate of reputation, and $\beta \in (0,1)$ is the punishment coefficient used to control the decline rate of reputation. λ is the decay factor, and Δb is the difference between the block height at which the node last participated in consensus and the current block height. The node with the highest reputation is elected as the leader node, and nodes with a reputation lower than the threshold b are prohibited from participating in consensus. When the reputation value of a node is higher than the threshold r, it is reset to r at the beginning of the next cycle to prevent the centralization tendency caused by excessively high reputation values. When the reputation value of a node is lower than b and it is prohibited from participating in consensus, its reputation value is restored to b in the next cycle.

The computation factor can be expressed as Equation (8), where $FF_i \in [0,1]$. Considering that some verification tasks in the consensus cannot be offloaded and need to be performed locally, and the leader node itself needs to perform more verification tasks than ordinary nodes, the higher the computational power of the leader node, the shorter the consensus delay. We utilize the computational power of all nodes to calculate the computational power factor for each node.

$$FF_i = \frac{f_i}{\max_{j \in U_n, n \in M} f_j} \tag{8}$$

In conclusion, the scoring of UAV nodes in the election of consensus leader nodes can be expressed as

$$R_i = \Xi_1 RF_i + \Xi_2 FF_i, \tag{9}$$

where Ξ_1 and Ξ_2 are corresponding weights and $\Xi_1 + \Xi_2 = 1$. The cluster head node evaluates the nodes based on their historical behavior and the situation of unmanned aerial vehicles within the cluster after each block is generated. The node with the highest score is selected as the leader node for the next round of consensus, and the election results are broadcast within the cluster. In order to obtain appropriate weight parameter values,

we set three groups of parameters [0.25,0.75], [0.5,0.5], and [0.75,0.25] for [Ξ_1 ,Ξ_2], and conducted consensus block-out simulation tests under each parameter condition, compared the block-out delay size of each group of parameters, and finally concluded that the shortest block-out delay was obtained under the parameter [0.5,0.5]. Considering that this is not the main innovation point of this paper, we do not further optimize this weight parameter.

2.3.3. Consensus Process

In this section, we consider the integration between the first blockchain and the second blockchain and design a collaborative block generation process for the two-tier blockchain. As shown in Figure 5, the block generation process consists mainly of the following steps.

S1 Within each cluster, the cluster head node updates the reputation evaluation based on the historical block generation behavior of the nodes. At the same time, the cluster head node sets the mining task difficulty based on its computational power used to solve the PoW nonce problem and the reputation update cycle requirement for the second blockchain.

S2 After completing the mining difficulty assessment, the cluster head node starts solving the PoW problem by finding a valid nonce to ensure the security of the block header. The cluster nodes collect interaction information.

S3 The slave nodes, upon receiving the reputation evaluation, are led by the leader node to reach the PBFT consensus.

S4 After completing one PBFT-block generation, the cluster nodes switch to the next leader node for the next PBFT consensus.

S5 During the mining process, the cluster head node continuously receives PBFT blocks and records the metadata of valid blocks in the block list.

S6 After finding the valid nonce, the cluster head node broadcasts it among the cluster heads to achieve consensus among other clusters.

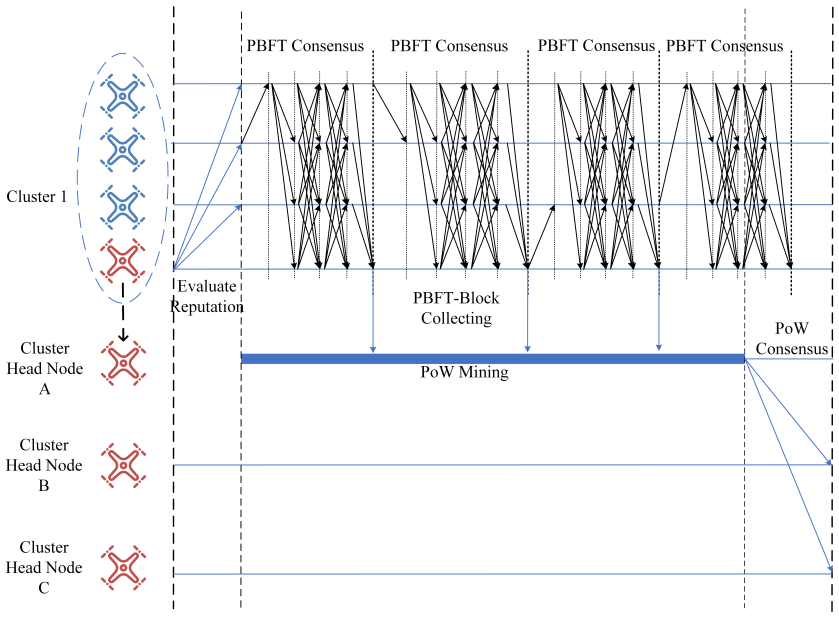

Figure 5. Consensus process.

2.3.4. Latency Analysis

PBFT consensus has the advantage of low block latency, but each step in the consensus process requires verification confirmation messages from a certain number of nodes. If some nodes experience long verification message delays, it will affect the overall latency of the consensus. Therefore, to meet the timeliness of reputations being recorded on the blockchain for UAVs, we utilize the high computing power of cluster-head UAVs to offload the verification tasks of low computing power nodes within the cluster. Considering

the cryptographic operation calculation, let f_m^{max} represent the total allocatable computing power of cluster head node m, and $f_{i,m}^{ver}$ represent the allocated computing power for UAV to perform offloaded PBFT consensus verification tasks. Thus, the total allocatable computing power of cluster head node m can be expressed as

$$f_m^{max} = \sum_{i=1}^{i} f_{i,m}^{ver}. \tag{10}$$

In the transmission process of task offloading, the system adopts a universal frequency reuse scheme, which means that all cluster head nodes use the same radio resources. Each cluster head node and the UAVs use orthogonal spectra for block transmission, which means there is no interference between the UAVs connected to the same cluster head node. Let $a_{i,m}$ represent the bandwidth percentage allocated by cluster head node m to UAV i. The useful signal power ratio CINR is denoted as $\gamma_{i,m}$, and the total bandwidth is B_m. According to Shannon's theory, the offloading rate between cluster head node m and UAV i is calculated as

$$u_{i,m} = a_{i,m} B_m \log_2(1 + \gamma_{i,m}). \tag{11}$$

After task offloading, the analysis of the latency in each step of the PBFT consensus for the drone cluster is as follows.

S1 Request: The block-generating node sends an unverified content block to the leader node. The content block includes the signature of the transaction user, and the block-generating node signs the content block. After the leader node verifies the signature, it proceeds to the next step.

S2 Latency Analysis: The delay in the request phase mainly comes from the leader node's verification of the block-generating node's signature. Assuming that the CPU cycles required for signature verification are δ_1, the offloaded verification computational power of the leader node is $f_{i,m}^{ver}$, the size of the received message is μ_1, and the offloading rate is $u_{i,m}$; the time required to complete this step can be expressed as

$$t_1 = \frac{\delta_1}{f_{i,m}^{ver}} + 2\frac{\mu_1}{u_{i,m}}. \tag{12}$$

S3 Pre-Prepare: Nodes sign the unverified content block and send the signed pre-prepared message to other nodes.

S4 Latency Analysis: The delay in the pre-prepare phase mainly comes from the leader node signing the received message and multicasting it to all other slave nodes. Assuming that the CPU cycles required for signing and verifying the received message digest are δ_2, the size of the received message is μ_2, and the broadcast rate of the leader node is u_i; the time required for the leader node to complete this step is calculated as

$$t_2 = \frac{\delta_2}{f_i} + \frac{\mu_2}{u_i}. \tag{13}$$

S5 Prepare: In the prepare phase, slave nodes validate the signature of the leader node to ensure that no one has forged the message. After checking the integrity of the message, the signature indicates agreement to validate and acknowledge receipt. The signed prepared message is then sent to other nodes. Each node receives $2f$ (f being the number of tolerated Byzantine nodes) and prepares messages before proceeding to the next step.

S6 Latency Analysis: In the prepare phase, the leader node needs to generate its signature for the prepare message and send it to the secondary nodes. It also needs to validate $2f$ to prepare messages from other slave nodes. Let us assume that the CPU cycles

required for signature verification of the prepared message are δ_3, and the size of the prepared message is μ_3. The leader node requires

$$t_3{}' = \lceil 1 + 2f \rceil \frac{\delta_3}{f_i} + \frac{\mu_3}{u_i}, \tag{14}$$

where f_i represents the CPU speed of the leader node and u_i represents the network bandwidth of the leader node. On the other hand, the slave nodes need to validate the signature on the received message from the leader node, generate their own prepared message, and multicast it to other secondary nodes. Therefore, they require

$$t_3{}'' = \frac{\delta_2}{f_{i,m}^{ver}} + 2\frac{\mu_2}{u_{i,m}} + \lceil 1 + 2f \rceil \frac{\delta_3}{f_i} + \frac{\mu_3}{u_i}. \tag{15}$$

It should be noted that the final delay required for this step depends on the time taken by the last node to complete the task. Thus, the array T^3 is created by arranging the time required for each node to complete the task in ascending order, giving us $t_3 = T_N^3$.

S7 Commit: In the commit phase, all nodes verify the integrity of the content. Slave nodes also verify if the content block is from the block-producing node. After verification, each node sends confirmation messages to other nodes. When a node receives $2f$ confirmation messages, it knows that the data are correct and proceeds to the next step.

S8 Latency Analysis: In the commit phase, all nodes need to verify the integrity of the block content, sign it, and multicast the commit message to each slave node. Slave nodes also need to verify the signature of the block-producing node. Let δ_4 represent the CPU cycles required for verifying the integrity, δ_5 represent the CPU cycles required for signing and verifying the commit message, μ_4 represent the size of unloaded information, and μ_5 represent the size of the broadcast message. The latency for the leader node can be calculated as

$$t_4{}' = \frac{\delta_4}{f_{i,m}^{ver}} + 2\frac{\mu_4}{u_{i,m}} + \lceil 1 + 2f \rceil \frac{\delta_5}{f_i} + \frac{\mu_5}{u_i}. \tag{16}$$

The latency for slave nodes can be calculated as

$$t_4{}'' = \frac{\delta_1 + \delta_4}{f_{i,m}^{ver}} + 2\frac{\mu_4}{u_{i,m}} + \lceil 1 + 2f \rceil \frac{\delta_5}{f_i} + \frac{\mu_5}{u_i}. \tag{17}$$

Similarly, the array T^4 contains the time required for each node to complete the task, sorted in ascending order as $t_4 = T_N^4$.

S9 Reply: All nodes send their signed commit messages to the blockchain, where the data becomes a pending transaction awaiting inclusion in a new block.

S10 Latency Analysis: In the reply phase, all nodes include their signatures in the reply messages and send them to the blockchain. The time required for each node to complete this task can be represented as

$$t_5{}' = \frac{\delta_6}{f_i} + \frac{\mu_6}{u_i}. \tag{18}$$

By arranging the completion times of all nodes in ascending order, we obtain an array T^5, and the final latency, denoted as $t_5 = T_N^5$.

The total block generation time of the PBFT consensus within a drone cluster is calculated as

$$T_{k,m}^{PBFT}(a_{i,m}, f_{i,m}^{ver}) = t_1 + t_2 + t_3 + t_4 + t_5. \tag{19}$$

Therefore, the optimization problem can be formulated as

$$Q: \min_{\boldsymbol{a},\boldsymbol{f}} \bar{\Gamma}(\boldsymbol{a},\boldsymbol{f}) = T_{k,m}^{PBFT}\left(a_{i,m}, f_{i,m}^{ver}\right) \tag{20}$$

$$s.t.\ a_{i,m} \in [0,1], \quad \forall i,\ \forall m \tag{20a}$$

$$0 \leq \sum_{i \in U_m} a_{i,m} \leq 1, \quad \forall m \tag{20b}$$

$$f_{i,m}^{ver} > 0, \quad \forall i,\ \forall m \tag{20c}$$

$$\sum_{i \in U_m} f_{i,m}^{ver} \leq f_m^{max}, \quad \forall m, \tag{20d}$$

where Equations (20a) and (20b) represent that the total bandwidth consumption of all offloaded tasks in cluster head node m does not exceed B_m, Equation (20c) represents the computational resource constraint of the cluster head node, and Equation (20d) ensures that the allocated resources for PBFT verification tasks do not exceed the total CPU resources of cluster head node m. The problem is decoupled into two optimization problems for variables $\boldsymbol{a}$ and $\boldsymbol{f}$, and an alternating optimization approach is used to solve this optimization problem, aiming to improve the information writing speed of the blockchain system.

3. Simulation and Results

3.1. Construction of the UAV Simulation Experiment Environment Based on the NS3 Platform

The network topology of the UAV cluster built on the NS3 platform is shown in Figure 6, where different numbers are used to identify different drone nodes. Based on this experiment, the network performance data of the UAV cluster on a secondary blockchain are obtained, and the specific simulation parameters are shown in Table 2.

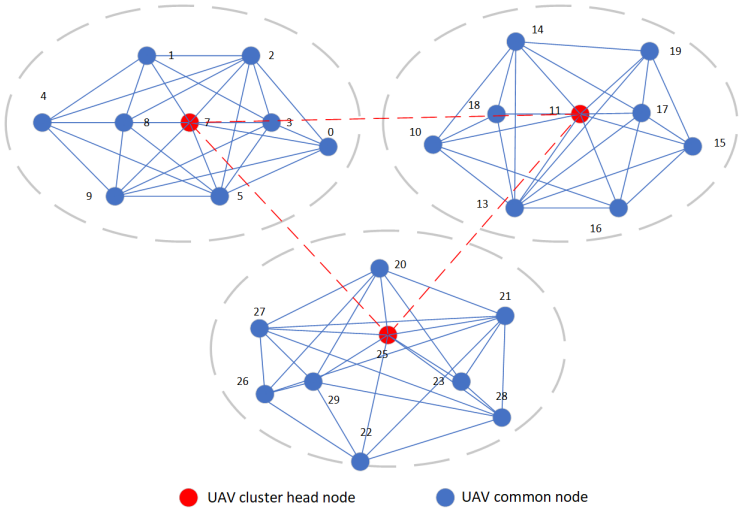

Figure 6. UAV cluster topology.

(1) Number of drones in each cluster: A UAV cluster in three simulated airspaces is selected, with 10 UAVs in each airspace, and the communication requirements among UAVs in the cluster are considered.

(2) Data transmission rate: To simulate the rate of data transmission, we set the appropriate data transmission rate. According to the communication technology and transmission requirements used in the UAV cluster, we set the data rate of 11 Mbps transmission per second to ensure the reliability of the data transmission performance of the UAV cluster.

(3) Routing protocol: To transmit data between drones, the ad hoc on-demand distance vector (AODV) routing protocol is selected as the communication protocol. AODV

is a commonly used wireless ad hoc network routing protocol, which is suitable for the dynamic network environment in UAV clusters. At the same time, the wireless device is set to AdhocWifiMac, and the networking mode of the drone cluster is set to ad-hoc, which is more in line with the data networking behavior of real drones.

(4) Wireless scenario: Using the appropriate wireless WIFI scene in the simulation model, to simulate the real environment of the wireless communication environment, the GroupMobilityModel mobile model and FriisPropagationLossModel path loss model are chosen to accurately describe the wireless transmission characteristics between the unmanned aerial vehicles (UAVs). GroupMobilityModel is a group movement model that is used to simulate simulation nodes to form groups and carry out cooperative movement according to certain rules, and can simulate the cooperative task movement of UAV clusters. The FriisPropagationLossModel fries propagation loss model, also called the free space propagation loss model, is a kind based on free space transmission theory, suitable for no obstacle of the simple path loss model of the open space environment, which will simulate UAV cluster signal propagation loss in a three-dimensional space.

(5) Communication distance: In the simulation, we set the communication distance between drones. This is determined according to the communication technology and scenario requirements of the UAV. We set the communication range of each UAV to 50 m to simulate the communication distance in the real environment; when the communication distance is too large, the communication efficiency will rapidly decline.

Table 2. Simulation initial setup.

Simulation Parameters	Value
Number of clusters	3
Number of drones in each cluster	10
Bandwidth	10 MHz
UAV transmission power	DssRate 11 MHz
Routing protocol	AODV
Mobile model	GroupMobilityModel
Loss model	FriisPropagationLossModel
Communication distance	50 m
Initial trust value	0.5 s

3.2. Analysis of UAV Node Trust Evaluation Results

Through simulation experiments, we evaluated the performance of a secondary blockchain-based drone cluster in terms of network performance. Here are some of our performance metrics and observations.

(1) Malicious node judgment accuracy

In order to analyze the effectiveness of the identification of malicious nodes in the experimental process, this paper adds the node result set to the malicious nodes in the UAV node data transmission. The scheme [29] and the Dynamic Evaluation Method for Drone Node Trust Based On Task Perception (DEMDT-TPT) in this paper are introduced to conduct a comparative test, we introduce the method proposed in scheme [29] as the benchmark. Through eight groups of simulated trust value evaluation rounds, the Bayesian inference model is used to judge the correctness of the results of ordinary nodes to malicious nodes, and five experiments are carried out in scenarios with different ratios of malicious nodes, and for the accuracy of the final judgment of malicious nodes, the average value is removed, and the comparison chart is shown in Figure 7. It can be concluded from Figure 7 that the scheme designed in this paper performs better, and can still maintain a high accuracy rate as the proportion of malicious nodes increases.

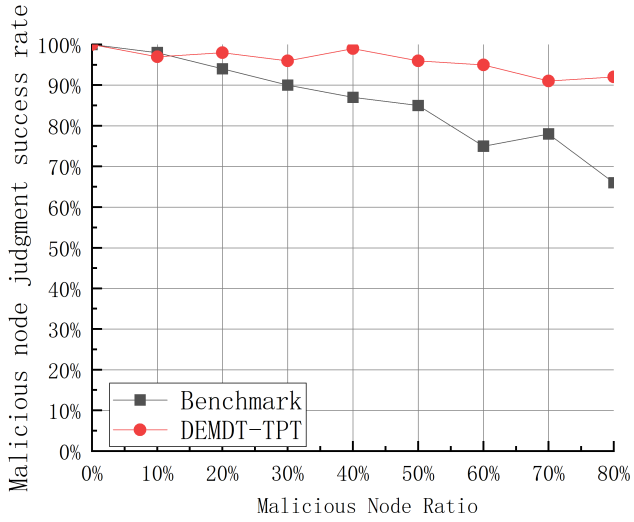

Figure 7. Malicious node judgment success rate.

(2) Trust value change rate

The scheme in this paper has the change in trust value caused by node behavior. To verify the anti-spoofing ability of the experimental scheme against malicious nodes, this paper counts the change rate of the trust value of UAV nodes and introduces the scheme [18] as the benchmark of this paper; the scheme [29] is an experiment on the rate of change of the trust value in the UAV field, which can better reflect the change in the trust value in general situations. Through ten groups of simulation experiments, the change rate of the latest trust value in each trust value evaluation round compared with the trust value of the previous round is recorded, and five simulation experiments for each group are conducted, and the average value of the trust value change rate after the team is taken, and the situation is compared in Figure 8.

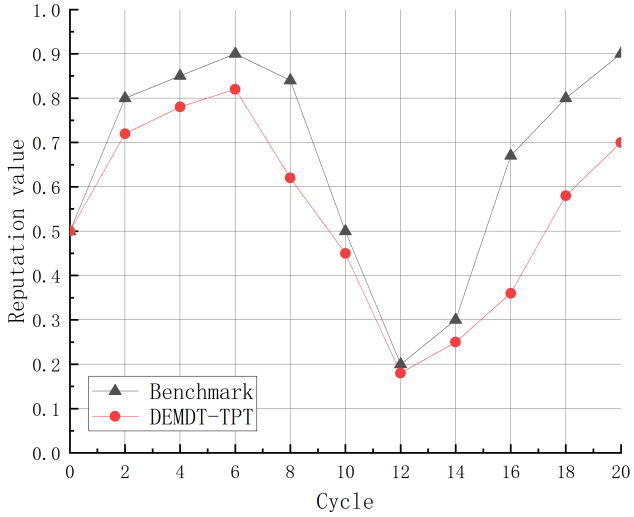

Figure 8. Change rate of node trust value.

In the figure, the scheme in this paper has a lower trust value change rate in the period 0–6, because the interaction of the UAV is honest; in the period 6–12, the trust value change rate decreases, and for the UAV, the interaction of malicious nodes appears in the nodes. It can be seen that the trust value of this paper drops faster, and the punishment for malicious nodes is higher than the baseline scheme. After that, in the period 12–20, the UAV nodes begin to reply to the trust value, but in this paper, the recovery speed in the scheme is obviously slower, which can prevent malicious nodes from obtaining high trust

values in a short period, and can significantly improve the anti-spoofing ability of malicious UAV nodes.

Based on the collected experimental data, we conducted an in-depth analysis and discussion of the experimental results. In addition to paying attention to the performance of performance evaluation, packet loss rate, and routing performance, we pay more attention to the data of the correct rate of judging the malicious nodes of drones and the rate of change of the trust value of drone nodes. The following is a summary of some of the experimental results.

First, in terms of performance evaluation, we evaluated the security performance of secondary blockchain-based UAV swarms through simulation experiments. We analyze metrics such as data transfer latency, throughput, and energy consumption to evaluate the system's efficiency and feasibility. Through the analysis of the experimental data, we were able to derive a quantitative assessment of the performance of the system and compare it with the design goals. The AODV routing protocol performs well in UAV swarms and can provide stable communication services.

Second, we evaluate the verdict accuracy against malicious nodes. By using the judgment method we designed, we can judge the malicious nodes in the UAV cluster in most cases. By analyzing the experimental data, we calculated the decision accuracy, that is, the proportion of correctly classified malicious nodes to the total number of malicious nodes. We observed that the judgment accuracy rate varies under different experimental conditions, and the comparison with other experimental schemes can illustrate the feasibility of the experimental scheme in this paper.

Then, in terms of the change rate of the trust value, we observed the change in the trust value of the drone. Through the analysis of the experimental data, we found that the trust value of the drone fluctuated and changed in different periods. We noticed that the rate of change of the trust value can represent an indication of the change in the behavior of the UAV, which is of great significance for verifying the feasibility of the scheme designed in this paper.

In addition, we also evaluate the performance of the blockchain in drone swarms. By analyzing the experimental data, we observed that the blockchain works effectively in data transmission between drones. We pay attention to the performance of blockchain delay, data update, and stability, and compare it with the design requirements.

3.3. Simulation Parameters of the Blockchain System

We conducted simulations on the proposed system model and optimization algorithm from different perspectives using MATLAB R2022a, to evaluate the effectiveness and system performance of the proposed approach. In the system model, we considered a quasi-static scenario where the transmit power of the UAVs and the distances between the UAVs remain constant during each consensus process. Some of the parameters used in the simulations are shown in Table 3.

Table 3. Simulation Parameters of Blockchain System.

Simulation Parameters	Value
f_m^{max}	16 GHz
B_m	5 MHz
δ_1	1×10^6 cycles
δ_2	8×10^6 cycles
$\delta_3, \delta_5, \delta_6$	5×10^5 cycles
δ_4	2×10^7cycles
$\mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6$	1×10^5 cycles
f_i	5 MHz, 15 MHz
u_i	1 Gb/s
Ξ_1, Ξ_2	0.5, 0.5

3.4. Analysis of Blockchain Simulation Results

Figure 9 presents the relationship between the block generation latency of the PBFT consensus algorithm and the number of UAVs per cluster for both the task offloading scheme and the non-offloading scheme.

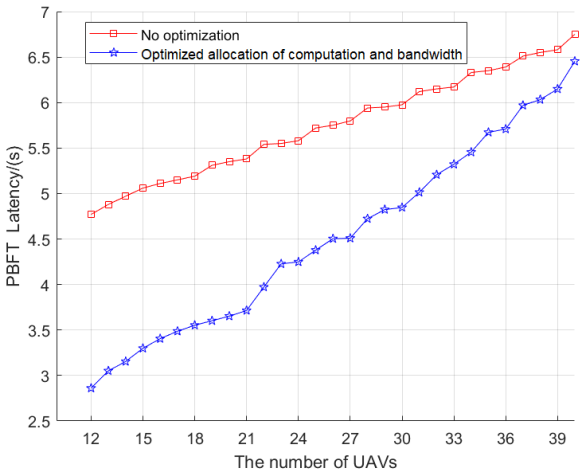

Figure 9. The relationship between block latency and the number of UAVs.

As the number of UAVs increases, the consensus latency for both schemes gradually increases. This is because the verification tasks required in the PBFT consensus process increase with the number of UAV nodes, thereby affecting the verification latency of the nodes. It can be observed from the graph that the block generation latency of the task offloading scheme increases at a faster rate compared to the non-offloading scheme. This is because the cluster head node has limited available computational resources. As the number of UAV nodes increases, the allocated bandwidth and computational power for each UAV decrease, resulting in a greater increase in latency.

In Figure 10, we investigated the relationship between the transaction throughput of the blockchain system and the number of UAVs under different resource allocation schemes. We set the number of PBFT blocks packed by PoW blocks to 30 and controlled the PoW mining delay by setting the mining task based on the block generation delay of PBFT blocks. As shown in Figure 10, the transaction throughput of the blockchain system decreases as the number of UAVs increases, and the decrease rate gradually slows down. This is because the block generation rate of PBFT blocks decreases with the increase in the number of UAVs. From the graph, it can be observed that under this resource allocation scenario, the bandwidth equal distribution method performs better than the computing power equal distribution method. This is because the allocation of computing resources has a greater impact on the latency.

The relationship curve between the block generation latency of PBFT and the allocated computing power of cluster head nodes is shown in Figure 11. As the allocated computing resources of cluster head nodes increase, the block generation time of PBFT gradually decreases. However, as the computing power of UAV nodes for task validation reaches saturation, the rate at which the time delay decreases slows down. Since PBFT requires at least 2/3 validation messages from other nodes to achieve consensus, the computing power average allocation scheme cannot compensate for the validation tasks of nodes with low computing power. Therefore, the block generation time will be significantly higher than the resource optimization scheme proposed in this paper.

In Figure 12, we investigated the relationship between the optimization of the leader node election scheme and the PBFT consensus latency. We set the number of UAVs per cluster to 30 and compared the latency of the leader node election scheme and the task offloading scheme under different average UAV computing power conditions.

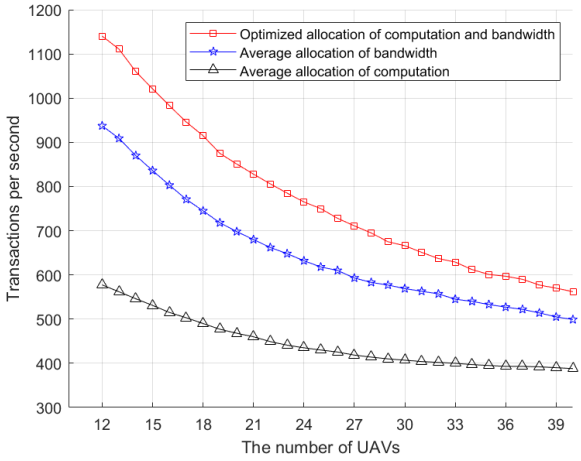

Figure 10. The relationship between the number of UAVs and the performance of our system.

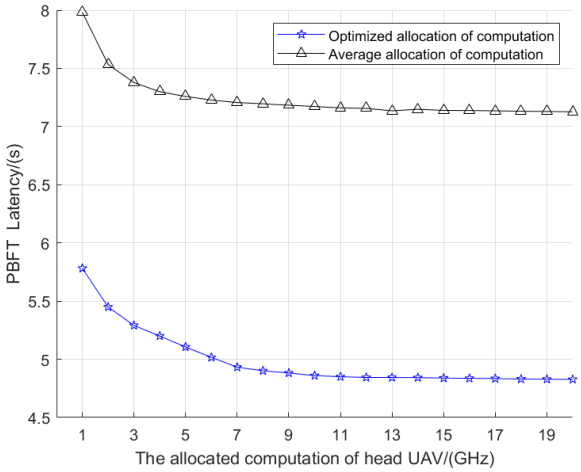

Figure 11. The relationship between latency and the computation of the head node.

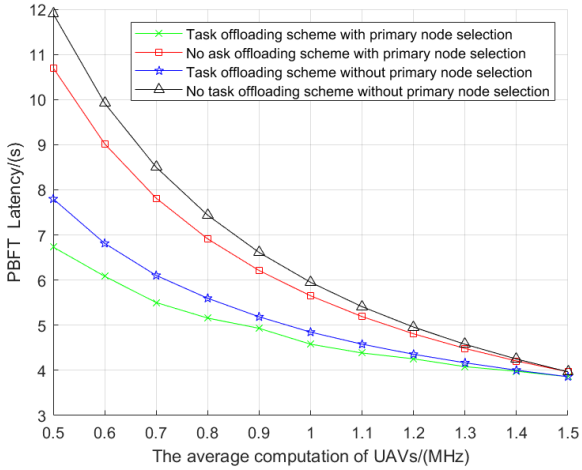

Figure 12. The relationship between the latency and the average computation of UAVs.

In Figure 12, as the average UAV computing power increases, the latency of the tasks that require UAV self-verification decreases. Therefore, the consensus latency of all schemes gradually decreases. However, this portion of the tasks is limited in size, so the reduction rate gradually slows down. Additionally, as the UAV's computing power increases, the advantage of the cluster head's computing power in the task offloading scheme gradually diminishes. As a result, the latency difference between different schemes also gradually

decreases. From the figure, we can see that the performance of the task offloading scheme alone is better than that of the scheme focusing solely on optimizing the leader node election. This is because the impact of improving the verification efficiency of the majority of regular nodes is greater than the impact of solely improving the verification efficiency of the leader node.

4. Discussion and Future Work

On the basis of the research in this paper, the contents that we can improve and continue to study are summarized as follows.

(1) For the layered blockchain-based UAV reputation management mechanism proposed in this paper, the PoW consensus mechanism is used. Since the PoW consensus mechanism depends on computing power, there may be a certain burden on UAV energy management. In the subsequent research, we will conduct in-depth research on blockchain technology and select the consensus mechanism more suitable for the UAV scenario, reducing the burden on UAV endurance.

(2) For the DEMDT-TPT proposed in this paper, there may be improper trust management of UAVs under certain circumstances. Considering the resource scheduling problem of trust management in the UAV scenario, we will optimize the processing flow of UAV trust data and improve the algorithm to achieve better performance.

In short, the two mechanisms proposed in this paper can improve the detection of malicious nodes in the trust management of drone clusters and increase the security of drone clusters. With the continuous innovation of relevant technologies, we will follow up more in-depth research to improve the security of drone clusters based on blockchain.

5. Conclusions

In this paper, we propose a hierarchical blockchain-based trust measurement model for drone cluster nodes, which addresses the issues of low accuracy in node trust evaluation and untrustworthy cross-domain sharing of drone trust values caused by high dynamic entry and exit of nodes in the trust cluster. Introducing the PBFT and PoW two-layer blockchain consensus mechanism and the dynamic evaluation method of UAV node trust based on task perception ensures the trustworthy evaluation of trust values and the inheritability of historical trust values. Through the double-layer blockchain system framework, it is possible to achieve multiple blocks at the same time as the packaging chain, improving the efficiency of the blockchain data chain. At the same time, in view of the limited computing power and communication resources of drones, the resource allocation optimization design of layered blockchain architecture under resource-limited conditions is constructed. By formulating the data throughput maximization problem and solving it by convex optimization, the throughput of blockchain data on the chain is improved. In this paper, NS3 and MATLAB simulation platforms are used to build a simulated UAV cluster scenario, and trust value evaluation algorithms are deployed on UAV nodes. Meanwhile, compared with other existing schemes, the effectiveness of the measurement algorithm in this paper is verified through indicators such as the accuracy rate of malicious nodes, robustness of trust value changes, blockchain system time delay, and throughput under different UAV cluster scenarios. The simulation results show that the scheme has good performance.

Author Contributions: Conceptualization, J.Z.; Formal analysis, J.Z.; Investigation, Z.W., J.L. and L.Z.; Software, J.Q. and P.G.; Supervision, R.C. and Y.L.; Validation, J.L. and Y.L.; Writing—original draft, J.Z., J.Q., P.G. and Z.W.; Writing—review and editing, J.Z. and R.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB3104901 and 2021YFB3101903.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

Acronym	Implication
PoW	Proof of Work
PoS	Proof of Stake
PBFT	Practical Byzantine Fault Tolerance
DEMDT-TPT	Dynamic Evaluation Method for Drone Node Trust Based on Task Perception
AODV	Ad hoc On-Demand Distance Vector Routing

References

1. Jia, Y.; Tian, S.; Li, Q. Recent development of unmanned aerial vehicle swarms. *Acta Aeronaut. Astronaut. Sin.* **2020**, *41*, 723738.
2. Noor, F.; Khan, M.A.; Al-Zahrani, A.; Ullah, I.; Al-Dhlan, K.A. A review on communications perspective of flying ad-hoc networks: Key enabling wireless technologies, applications, challenges and open research topics. *Drones* **2020**, *4*, 65. [CrossRef]
3. Ge, C.; Zhou, L.; Hancke, G.P.; Su, C. A Provenance-Aware Distributed Trust Model for Resilient Unmanned Aerial Vehicle Networks. *IEEE Internet Things J.* **2021**, *8*, 12481–12489. [CrossRef]
4. Xin, D.; Tunan, X.; Xianchang, L. Future-oriented Collaborative Incentive Mechanism of UAV Cross-domain Tasks: Based on the Token Idea. *J. Command. Control.* **2023**, *9*, 66–75.
5. Lakew, D.S.; Sa'ad, U.; Dao, N.N.; Na, W.; Cho, S. Routing in flying ad hoc networks: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1071–1120. [CrossRef]
6. Wen, S.; Deng, L.; Shi, S.; Fan, X.; Li, H. Distributed congestion control via outage probability model for delay-constrained flying Ad Hoc networks. *Wirel. Commun. Mob. Comput.* **2020**, *2020*, 1–9. [CrossRef]
7. Alladi, T.; Chamola, V.; Sahu, N.; Guizani, M. Applications of blockchain in unmanned aerial vehicles: A review. *Veh. Commun.* **2020**, *23*, 100249. [CrossRef]
8. Chen, Y.; Jia, L. Two-layer grouped Byzantine fault tolerance algorithm for UAV swarm. *J. Commun.* **2022**, *43*, 96–103.
9. Alaa, M.; Oumayma, A.D.; Mohamad, A.J. Towards Trust Model in Unmanned Aerial Vehicle Ad Hoc Networks. *J. Commun. Softw. Syst.* **2021**, *17*, 213–220. [CrossRef]
10. Singh, K.; Verma, A.K. TBCS: A trust based clustering scheme for secure communication in flying ad-hoc networks. *Wirel. Pers. Commun.* **2020**, *114*, 3173–3196. [CrossRef]
11. Nivedita, V.; Nandhagopal, N. Improving QoS and efficient multi-hop and relay based communication framework against attacker in MANET. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *12*, 4081–4091. [CrossRef]
12. Singh, K.; Verma, A.K. A fuzzy-based trust model for flying ad hoc networks(FANETs). *Int. J. Commun. Syst.* **2018**, *31*, 23–47. [CrossRef]
13. Pouyan, A.A.; Yadollahzadeh Tabari, M. FPN-SAODV: Using fuzzy Petri nets for securing AODV routing protocol in mobile Ad hoc network. *Int. J. Commun. Syst.* **2017**, *30*, e2935. [CrossRef]
14. Zeng, Y.; Zhou, J.; Liu, Y.; Cao, T.; Yang, D.; Liu, Y.; Shi, X. Trustworthy Routing Protocol for Unmanned Aerial Vehicle Ad Hoc Networks. *J. Comput. Sci. Explor.* **2021**, *15*, 2304–2314.
15. Asuquo, P.; Cruickshank, H.; Ogah, C.P.A.; Lei, A.; Sun, Z. A distributed trust management scheme for data forwarding in satellite DTN emergency communications. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 246–256. [CrossRef]
16. Cho, J.H.; Chen, R. PROVEST: Provenance-based trust model for delay tolerant networks. *IEEE Trans. Dependable Secur. Comput.* **2016**, *15*, 151–165. [CrossRef]
17. Ge, C.; Ma, X.; Liu, Z. A semi-autonomous distributed blockchain-based framework for UAVs system. *J. Syst. Archit.* **2020**, *107*, 101728. [CrossRef]
18. Gai, K.; Wu, Y.; Zhu, L.; Choo, K.K.R.; Xiao, B. Blockchain-enabled trustworthy group communications in UAV networks. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 4118–4130. [CrossRef]
19. Chen, A.G.; Zhou, X.C.; Wen, X.Y.; Li, J.H.; Yan, K.; Xie, X.R. A Hierarchical Blockchain-based Identity Authentication Scheme for Drone Clusters. In Proceedings of the 2021 Chinese Automation Congress, Beijing, China, 22–24 October 2021; pp. 378–383.
20. Tan, Y.; Liu, J.; Kato, N. Blockchain-based key management for heterogeneous flying ad hoc network. *IEEE Trans. Ind. Inform.* **2020**, *17*, 7629–7638. [CrossRef]
21. Xiao, Y.; Zhang, N.; Lou, W.; Hou, Y.T. A Survey of Distributed Consensus Protocols for Blockchain Networks. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1432–1465. [CrossRef]
22. Cui, L.; Yang, S.; Chen, Z.; Pan, Y.; Xu, M.; Xu, K. An Efficient and Compacted DAG-Based Blockchain Protocol for Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2020**, *16*, 4134–4145. [CrossRef]
23. Xu, G.; Liu, Y.; Khan, P.W. A Survey of Distributed Consensus Protocols for Blockchain Networks. *IEEE Trans. Ind. Inform.* **2020**, *16*, 4252–4259. [CrossRef]

24. Xu, Y.; Zhang, H.; Ji, H.; Yang, L.; Li, X.; Leung, V.C.M. Transaction Throughput Optimization for Integrated Blockchain and MEC System in IoT. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 1022–1036. [CrossRef]
25. Barka, E.; Kerrache, C.A.; Benkraouda, H.; Shuaib, K.; Ahmad, F.; Kurugollu, F. Towards a trusted unmanned aerial system using blockchain for the protection of critical infrastructure. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e3706. [CrossRef]
26. Kang, J.; Xiong, Z.; Niyato, D.; Ye, D.; Kim, D.I.; Zhao, J. Toward secure blockchain-enabled internet of vehicles: Optimizing consensus management using reputation and contract theory. *IEEE Trans. Veh. Technol.* **2019**, *68*, 2906–2920. [CrossRef]
27. Keshavarz, M.; Gharib, M.; Afghah, F.; Ashdown, J.D. UASTrustChain: A decentralized blockchain-based trust monitoring framework for autonomous unmanned aerial systems. *IEEE Access* **2020**, *8*, 226074–226088. [CrossRef]
28. Yang, J.; Liu, X.; Jiang, X.; Zhang, Y.; Chen, S.; He, H. Toward Trusted Unmanned Aerial Vehicle Swarm Networks: A Blockchain-Based Approach. *IEEE Veh. Technol. Mag.* **2023**, *18*, 98–108. [CrossRef]
29. Chen, J.M.; Li, T.T.; Panneerselvam, J. TMEC: A trust management based on evidence combination on attack-resistant and collaborative internet of vehicles. *IEEE Access* **2018**, *7*, 148913–148922. [CrossRef]