Secure and Scalable Trust Management in IoT: A Hierarchical Blockchain-based Approach

***Abstract*—The diverse and resource-constrained nature of Internet of Things (IoT) devices make them vulnerable against various security attacks. Effective trust management within the IoT ecosystem is crucial for reliable data collection and sharing, as well as the detection of malicious nodes. Centralized trust management methods are inefficient due to several challenges, including single point of attack/failure, unauthorized manipulation of trust data, resource limitations of smart devices, and scalability issues. Blockchain technology provides a suitable solution for trust management due to its decentralization, transparency, and immutability features. However, deploying blockchain for IoT devices is not simple due to the low performance and high computational costs of consensus algorithms, limited resources of smart devices, and the large volume of transactions created by nodes. In this paper, a hierarchical trust management approach based on blockchain is proposed. The proposed approach evaluates each node's reputation and organizational trust at both intra-organizational and inter-organizational levels. At the internal level, a lightweight blockchain is used to evaluate and store the trust score of the nodes. At the inter-organizational level, interactions between organizations and their trust level are recorded in the public blockchain. Two methods are proposed, i.e. probing-based and evidence-based, for evaluating the reputation of each node and the trust level of each organization. The evaluation results show that with a maximum of 35% malicious nodes within an organization, the proposed method can correctly identify the malicious and honest nodes. The recall and specificity measures obtained are both greater than 0.9. Additionally, organizations with more than 35% of malicious nodes are blacklisted and suspended.**

***Index Terms*—Hierarchical Trust Management, Blockchain, Internet of Things**

I. INTRODUCTION

In recent years, the rapid growth of Internet of Things (IoT) technology had significant impact in our daily lives. Smart devices in the IoT ecosystem collect data from the environment, process it, and analyze it to extract valuable information, enabling the provision of various services [1]. These connected devices usually have specific characteristics, such as low computing power, limited storage space, and heterogeneity [2], [3]. They have the ability to communicate and share information, playing an important role in various applications, such as smart healthcare [4], smart transportation [5], smart homes [6], and smart cities [7]. Researcher have shown that IoT devices are prone to numerous security challenges, such as the need for secure mechanisms to authenticate nodes during onboarding, robust access control mechanisms to manage authorization, and effective privacy mechanisms [8], [9], [10]. While authentication and access control mechanisms can mitigate external threats, they may fall short in ensuring the trustworthiness of authorized nodes, as authenticated nodes can potentially turn malicious post-authentication, posing challenges in detecting internal attacks. Therefore, trust management and continuously evaluating the trustworthiness of each node in IoT is an important challenge [11], [12].

The motivation for implementing a trust management mechanism is to safeguard IoT nodes against malicious and compromised nodes. Malicious nodes have the capability to execute various attacks and disseminate false recommendations. Therefore, trust evaluation mechanisms are introduced to alleviate security concerns. Through trust management, each node assesses its level of trust towards other nodes before engaging in a communication. If the trust level meets the specified threshold, the node proceeds with the communication; otherwise, it disregards those nodes with lower trust levels [13].

Trust management mechanisms are categorized into two main types: centralized and distributed [2]. In centralized trust management systems, trust evaluation and data storage are handled by a central entity, typically a cloud server. Awan et al. [12] proposed a multi-level central authority-based inter-domain trust management model called HoliTrust, tailored for interactions among Internet of Things (IoT) nodes. This model incorporates various central authorities, such as community servers, domain servers, and trust servers, to ensure correctness, efficiency, and reduced computational load on IoT nodes. However, due to the large number of nodes, the limited computational capabilities of smart objects, and the dynamic network environment, Chen et al. [14] introduced IoTrust, an architecture integrating Software Defined Networking (SDN) with IoT. IoTrust comprises five layers, including the object layer, node layer, SDN control layer, organization layer, and reputation management layer. The reputation management layer serves as the central node responsible for calculating and storing trust data, albeit this centralized approach may lead to a single point of failure and hinder scalability.

In distributed trust management systems, nodes are tasked with computing and maintaining trust information locally. Zheng et al. [15] proposed a network security mechanism based on distributed trust management to address threats in wireless sensor networks. This model employs a distributed network structure to compute the local trust degree of nodes based on their interaction behavior, subsequently deriving a global trust degree that reflects overall trust levels and aids in identifying malicious nodes. Recognizing the significance of healthcare-related information and the constraints of IoT, Ebrahimi et al. [16] proposed a decentralized and secure trust management model based on the Dempster-Shafer evidence theory and social relationships among object owners. This model aims to establish a trust framework while considering factors such as reliability and privacy. Alghofaili et al. [17] proposed a trust management model to secure IoT devices and services. Their model leverages two techniques: the Simple Multi-Attribute Rating Technique (SMART) and Long Short-

Term Memory (LSTM). SMART calculates trust scores based on delay, data loss, and throughput, while LSTM, a deep learning technique, identifies malicious behavior by analyzing data patterns. Awan et al. [13] introduced RobustTrust, a robust cross-domain distributed trust management system designed to address challenges across different domains. This mechanism divides trust into three components: knowledge, reputation, and experience, each with subcomponents for evaluating trust relative to other nodes. However, in distributed trust management systems, nodes face various challenges, such as the potential for local trust manipulation, and resource limitations of each node for calculating and storing trust values. Researchers suggest that Blockchain technology can help overcome these challenges [18].

Blockchain technology is a distributed ledger system where verified transactions are stored in tamper-proof blocks and are transmitted across a decentralized, transparent peer-to-peer network, which could be helpful in overcoming trust evaluation issues in IoT [19]. Blockchain can protect the system against unauthorized data manipulation and improve scalability by removing central authorities. However, its high computational demands, bandwidth usage, and latency make it unsuitable for many IoT applications [20]. A key challenge lies in scaling data collection and storage, as IoT devices have limited resources. Additionally, blockchain alone cannot guarantee the accuracy of the source data, as malicious devices could inject invalid information that becomes uncorrectable once stored [21]. Lightweight blockchain platforms have been proposed in the literature to address the resource limitations of IoT [22], [23], [24], [25], [22]. However, with the large volume of transaction and the growing blockchain size, monitoring the accuracy of the system becomes difficult. To address these issues, two-layer blockchain architectures have been proposed in the literature to manage trust and privacy in the IoT [26], [27], [28].

This paper proposes a hierarchical trust management system for smart cities. In the proposed model, a smart city comprises multiple organizations, each of which own numerous smart objects that request or provide services to the community. We assume that each organization uses a local blockchain for internal trust management. We consider two organizational trust levels in our proposed model, i.e. intra-organizational and inter-organizational levels. The intra-organizational level leverages a light blockchain platform to store trust values for smart objects within the organization. The inter-organizational level employs a global, reputation-based blockchain for interactions between organizations. When an organization requests a service, a suitable service provider is chosen based on the developed smart contracts and the requested service type and the requesting organization's category. The chosen organization's smart object responds to the request, and the service provider's trust value will be updated on the global blockchain. The key contributions of this paper are:

1) A scalable and privacy-preserving two-level blockchain-based trust management model is proposed. This model uses a private and lightweight blockchain within each organization (intra-organizational layer) to streamline trust calculations and transaction confirmation. Additionally, a global blockchain in the inter-organizational layer is used to store each organization's reputation information and the trust scores.
2) Trust values for individual IoT nodes is calculated by comparing observations from neighboring nodes, achieved through probing and evidence-based methods.
3) Organizational trust value incorporates both direct and indirect trust calculation, creating to a three-tier classification system: whitelisted, graylisted, and blacklisted organizations.

This paper is structured as follows. Section II provides a concise overview of the relevant trust management mechanisms. Section III details the proposed trust mechanism. Section IV presents the performance evaluation results of the proposed model. Section V examines the scalability of the proposed trust management model and Section VI analyzes the resilience of the proposed methods against security attacks. Section VII concludes the paper by summarizing the key findings.

II. RELATED WORK

This section explores the application of single-layer and two-layer blockchain architectures in trust management systems.

A. *Single-layer blockchain trust management*

With the expansion of smart cities and the increase of network traffic, edge computing has improved the quality of service. However, it also faces security challenges, such as central node failure and insider attacks. Wang et al. [29] propose a blockchain-based trust management model for edge devices in the smart city. Due to storage and computing limitations in edge devices, common consensus algorithms are difficult to implement. To solve this problem, a reputation-based consensus algorithm is proposed which divides the network into two parts: the edge servers and the edge devices. The solution often struggles with the heterogeneity, mobility, and scalability demands of the Internet of Things (IoT). To address these challenges, Kouicem et al. [30] proposed a BC-Trust, a decentralized and hierarchical blockchain-based trust management protocol with mobility support. In BC-Trust, mobile devices assess trust recommendations about other devices without relying on a central authority. Additionally, the protocol leverages cloud nodes and service providers to maintain the blockchain, alleviating resource-constrained IoT devices from storing trust information and performing complex computations.

Due to limited interaction experience, nodes may require recommendations to assess other nodes. However, these recommendations can be unreliable. To address this, Wu et al. [11] proposed a blockchain-based distributed trust management mechanism for the IoT called BBTM. In BBTM, trust computation is published as a crowdsourcing task, where a sensor node broadcasts a request to calculate the trustworthiness of its cooperating partner. Nearby edge nodes then receive this task and provide their feedback about the requesting nodes. Layered Blockchain-based Trust Management

(LBTM) and its distributed variant DLBTM [28] propose multi-tier trust architectures where local nodes evaluate peers and periodically synchronize with a higher-level blockchain. These models improve scalability over single-chain designs and reduce evaluation latency. However, they primarily rely on static trust computation formulas (e.g., fuzzy logic) and do not incorporate hybrid trust evaluation methods such as combining evidence-based and probing-based analysis. Furthermore, they lack an explicit mechanism for smart contract-based aggregation or adaptive parameter tuning under dynamic adversarial behavior.

To address concerns about data reliability at the source in the Internet of Things (IoT), Dedeoglu et al. [31] proposed a layered model. This model separates data trust evaluation (sensor observations) from block verification (blockchain layer) using trust and reputation modules within gateways. In the Internet of Social Things (IoST), Amiri et al. [32] proposed LBTM, a lightweight blockchain-based trust management system. LBTM leverages social connections between IoT devices to incorporate valuable social information alongside traditional feedback data, enhancing trust evaluation during interactions (data transfer, service provision). Devices can directly interact with the system or use edge nodes.

BC-Trust [30] is a foundational work that uses a single-layer blockchain to store and manage trust values. Trust scores are computed based on direct interactions and stored immutably to prevent tampering. However, the model depends on a proof-of-work (PoW) consensus mechanism, making it computationally expensive and less suitable for resource-constrained IoT devices. Moreover, the single-layer architecture results in poor scalability as the number of transactions grows, limiting its deployment in large-scale networks such as smart cities.

RobustTrust is a non-blockchain statistical trust framework that uses data-driven outlier detection and weighted voting to assess trustworthiness. It offers flexibility and robustness in small and medium-sized networks. Nevertheless, the lack of a secure distributed ledger means that trust data can be spoofed or modified by attackers, reducing its resilience to tampering and long-term audits. Additionally, the model lacks a clear mechanism to verify source authenticity in distributed environments.

T-Broker [?] introduces a broker-based trust negotiation model where IoT devices select trustworthy services based on historical interactions and contextual policies. While flexible in service selection, T-Broker is not blockchain-based and therefore lacks transaction traceability and tamper-resistance. It also assumes relatively powerful devices capable of executing negotiation protocols, making it impractical for highly constrained IoT sensors or real-time systems such as vehicular networks.

Collaboration between devices from different domains in the Industrial Internet of Things (IIoT) raises concerns about trust in device-to-device interactions. Wu et al. [33] address this by proposing PPTMA, a blockchain-based privacy-preserving trust management architecture for cross-domain trust assessment. PPTMA utilizes federated learning to train task-specific trust models and employs a game theory-based incentive mechanism to encourage honest and active participation by IIoT devices in sending trust data to the blockchain, ultimately improving trust calculation accuracy. Additionally, a parallel consensus protocol is proposed to enhance consensus efficiency.

B. Two-layer blockchain trust management

Two of the most important challenges of the Internet of Things are the protection of smart objects and the need to guarantee their independence. To solve these challenges, Corradini et al. [27] have presented a two-layer blockchain-based trust management mechanism. In this model, intelligent objects are appropriately grouped into communities. Each community is linked to a local blockchain and is used to record probing transactions and evaluate the trust of one object to another from the same community or another community. To reduce the volume of transactions, a lightweight blockchain is used which aggregates transactions periodically. If the reputation value of the smart object is less than the threshold, it will be removed from the community. In the global blockchain, the list of objects belonging to each community, the amount of reputation of smart objects of each community and the trust of each community in other communities are calculated and updated using smart contracts.

In the Internet of Vehicles (IoV), cars collect data from their environment and share it through the Internet of Things (IoT). However, vehicles can transmit false information, jeopardizing trust within the network. To address this and overcome the limitations of single-layer blockchains, Ruan et al. [28] propose a two-level blockchain-based trust management mechanism called DLBTM. DLBTM utilizes separate blockchains for vehicles and Road Side Units (RSUs). The vehicle blockchain involves vehicles and a limited number of RSUs participating in consensus. The RSU blockchain consists solely of RSU nodes and miners. RSU nodes handle tasks like block generation, calculating final vehicle trust (combining vehicle message trust, vehicle evaluation trust, and vehicle past behavior trust), delivering messages to requesting vehicles, and storing both vital and temporary messages. Vehicle nodes focus on generating, evaluating, and requesting messages. Through logistic regression, DLBTM calculates the final trust value by combining these three trust components. This model boasts an accuracy of at least 90% in identifying malicious nodes over time. Fischer et al. [34] incorporated the direct and indirect feedback of a vehicular nodes towards others to estimate the reputation of vehicles in a network. Huang et al. [35] computed the node's reputation by employing weighted voting algorithm. The importance of nodes with higher weight would have higher preferences and has been computed on the basis of nodes presence near the event. Yang et al. [36] proposed a blockchain based decentralized trust management system based on positive and negative votes and incorporates the RSU for vote aggregation. The system is decentralized but it has not provided any mechanism for ensuring vote privacy of vehicles. Further, the system does not consider the trustworthy weight of participating vehicles. Ray et al. [37] incorporated a data-analytics approach to estimate the trusted behaviour of nodes in a vehicular ad-hoc network (VANET). The system weighs each consider the trust weight of feedback provider and well-defined rules to estimate the trust of the vehicles. A centralized system can be used to collect the feedback, feedback aggregation and relying the aggregated feedback to other participants of the network. Muhammad et al. [38] adopted crowdsourcing approach for aggregating the feedback from participating vehicles in a completely decentralized fashion by employing a homomorphic encryption. The approach also incorporated trust weights while evaluating the trustworthiness of the vehicles. Haitham et al [39] proposed a blockchain-based reputation system which computes reputation of vehicles in a privacy-preserving, secure and decentralized way.

Awan et al. [40] propose a model for secure routing in sensor networks using blockchain technology. Their model allows for choosing reliable paths between source and destination nodes by calculating trust values and detecting malicious behavior. The system utilizes a combination of public and private blockchains to store identities of collector nodes (ANs) and sensor nodes (SNs), respectively, and perform authentication. Additionally, the model identifies and removes malicious sensor nodes based on three parameters: forwarding rate, response time, and delayed transmission. Furthermore, it ensures secure routing by considering the remaining energy and confidence values of sensor nodes.

C. Comparison and Limitations of Existing Work

Table I provides a high-level comparison of the discussed models across six dimensions: blockchain usage, consensus mechanism, scalability, trust evaluation method, hierarchical design, and resilience to malicious attacks.

As seen, none of the existing models simultaneously address the challenges of scalability, hybrid trust evaluation, real-time adaptability, and blockchain-based aggregation in a hierarchical structure. These limitations form the core motivation for our proposed framework.

D. Summary of Contributions

Unlike the above models, our approach introduces a scalable two-layer architecture combining local trust aggregation and global consensus through lightweight blockchains. It supports hybrid trust evaluation (probing- and evidence-based), utilizes efficient PoA consensus, and ensures robustness against multiple attack types through redundancy, adaptive pruning, and secure smart contracts. The effectiveness of our model is validated through extensive simulations, demonstrating superior scalability, accuracy, and resilience in comparison with state-of-the-art alternatives.

III. PROPOSED MODEL

As explained in the introduction section, in the proposed model, we assume that each organization owns numerous smart objects that request or provide services. We also assume that each organization is connected to either a lightweight or a local blockchain, forming an intra-organizational layer. The inter-organizational layer interfaces with the global/public blockchain and stores information, such as the list of smart objects associated with each organization, the reputation score of each object, and the trust level of each organization. When an organization requests a service within this model, a smart contract is activated, recommending an organization based on the requested service and its category. Subsequently, the service request is forwarded to the designated organization, and a smart object from that organization responds to the request. The object's response is transmitted to the organization via a secure communication channel. Following this, the organization's trust value is computed through a smart contract and stored in the global blockchain.

The proposed hierarchical trust model is based on the following key assumptions:

- Honest Majority: More than 50% of the nodes in both intra- and inter-organizational layers are assumed to behave honestly. This threshold is critical, as the correctness of most consensus-based mechanisms (e.g., Proof of Authority (PoA)) and aggregation-based trust models depends on the presence of a majority of honest nodes. If a dishonest majority (i.e., $\geq 50\%$) emerges, reputation scores can be manipulated, thereby compromising the accuracy of trust evaluations.
- Service Redundancy: Each service within an organization has at least 2—3 redundant smart objects capable of providing the same service. This redundancy enables effective probing and evidence-based validation. It also allows pruning of outlier responses for more accurate reputation aggregation. In a 1000-node smart city, dividing nodes into 100 partitions with 2–3 redundant devices per partition ensures sufficient honest majority and fault tolerance.

  Assumption 2 asserts that the maximum number of colluding nodes t within any partition p_{ij} remains less than the partition size $|p_{ij}|$. This is justified by the redundant design of service partitions, where each partition includes multiple nodes from independent subgroups or trust zones, limiting coordinated compromise. For example, in a 10-node partition with 3 redundant nodes per service, even if 3 nodes are malicious, the trust evaluation remains robust due to the dominance of honest majority and the pruning of anomalous outputs. Additionally, node authentication using digital certificates mitigates Sybil attacks and constrains entry of unauthorized participants, further supporting this assumption.

A. Intra-organizational layer

In interactions among smart objects, some nodes may exhibit malicious behavior. We propose two methods for calculating the trust score and reputation value of each smart object, i.e. probing-based and evidence-based.

1) Probing-based method: The probing-based method, inspired by [27], involves a smart object seeking services from another smart object within the same or a different organization. This mechanism assesses the reliability and reputation of objects through probing. Nodes undergo testing via probe queries regarding their service capabilities, with their responses compared to those of other nodes offering the same

TABLE I: Comparison of Existing Blockchain-Based Trust Models in IoT

Model	Blockchain	Consensus	Scalability	Trust Eval.	Hierarchy	Attack Resilience
BC-Trust	Yes	PoW	Low	Direct	No	Medium
RobustTrust	No	–	Medium	Statistical	No	Low
LBTM	No	–	Medium	Fuzzy Logic	Partial	Medium
DLBTM	Yes	BFT	Medium	Fuzzy Logic	Yes	Medium
T-Broker	No	–	Medium	Negotiation	No	Low
Proposed	Yes	PoA	High	Hybrid (Prob/Evid)	Yes	High

service. This comparison enables the calculation of the object's reliability. All transactions involved in evaluating smart object reliability are recorded in a lightweight blockchain with a smart contract. In this system, the consensus mechanism used is Proof-of-Authority (PoA). In this approach, only trusted entities are allowed to produce new blocks, and nodes must authenticate their identities. Compared to other algorithms such as Proof-of-Work, this mechanism requires less energy and time, making it more suitable for fast and reliable transaction validation.

As depicted in Figure 1, the proposed model comprises multiple organizations, with each object belonging to precisely one organization capable of inter-communication. The model consists of two layers: the intra-organizational layer and the inter-organizational layer. In the intra-organizational layer, each organization is composed of several smart objects that both request and provide services. A lightweight blockchain is employed to manage transaction volume, record probing transactions, and oversee the trustworthiness of each smart object within its respective organization. Periodically, once the local blockchain reaches a predetermined length threshold, these transactions are aggregated, and the resultant values are stored in the global blockchain.

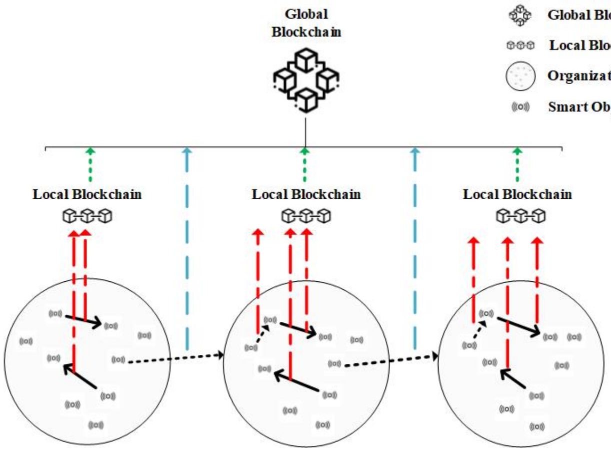


Fig. 1: Probing-based trust evaluation model

a) Assessment of Trust in Smart Objects: Regarding the services provided by objects within the organization, it is assumed that a certain level of service redundancy exists to prevent service interruptions (multiple objects within the same organization can provide the same service). If half or more of the nodes are malicious, however, trust values cannot be accurately assigned. Consequently, both the probing-based and evidence-based methods assume that over 50% of the nodes are honest. Gateway nodes are selected based on computational capacity and pre-authenticated via a trusted third-party certificate authority during network setup.

A request initiated by a trustor tr_i to a trustee te_j within an organization is denoted as req_{i_j}. Subsequently, a partition p_{i_j} comprising smart objects in the organization capable of responding to req_{i_j} is identified. A pruning process is then applied to p_{i_j} to eliminate smart objects with outlier data, resulting in the pruned partition $\widehat{p_{i_j}}$. To assess the trustworthiness of T_{i_j}, based on the output out_j provided by te_j and the outputs from smart objects within $\widehat{p_{i_j}}$, it is computed using the equation 1.

$$T_{i_j} = \frac{1}{|\widehat{p_{i_j}}|} * \Sigma_{o_k \in \widehat{p_{i_j}}} \mathcal{F}_k^j \tag{1}$$

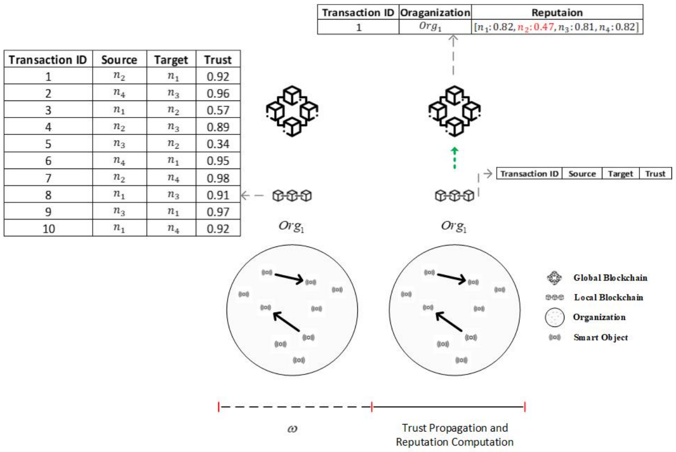
In equation 1, $|\widehat{p_{i_j}}|$ represents the count of smart objects in the pruned partition, while the function $\mathcal{F}_k^j$ defines the criteria for validating the outputs of smart objects. This validation involves comparing the function with a threshold value, which is determined based on specific application requirements and calculated using equation 2.

$$\mathcal{F}_k^j = \begin{cases} 1 & \text{if out}_k \text{ supports out}_j, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

b) Calculation of Smart Object's Reputation Value : The reputation of a smart object reflects the collective perception of the entire organization towards that object, as illustrated in Figure 2. Reputation calculation serves as a method to maintain blockchain efficiency by minimizing its size, and it is computed via a smart contract when the local blockchain reaches its length threshold w. Within this threshold, all probing transactions are aggregated. TrS_j represents the set of nodes that have engaged in at least one probing transaction with te_j. Upon reaching the threshold w, the reputation of te_j is determined using Equation 3

$$R_j^w = \begin{cases} \alpha \cdot R_j^{w-1} + (1-\alpha)\overline{T_j^w} & if \quad TrS_j \neq \varnothing, \\ R_j^{w-1} & otherwise. \end{cases} \tag{3}$$

Where, R_j^{w-1} represents the reputation value of te_j from the previous aggregation. The parameter α serves as a weight parameter to assess the significance of past data relative to new data. In critical scenarios necessitating a high level of security assurance, the value of α is set to a lower value. We designate



Transaction ID	Oraganization	Reputaion
1	Org_1	$[n_1{:}0.82, n_2{:}0.47, n_3{:}0.81, n_4{:}0.82]$

Transaction ID	Source	Target	Trust
1	n_2	n_1	0.92
2	n_4	n_3	0.96
3	n_1	n_2	0.57
4	n_2	n_3	0.89
5	n_3	n_2	0.34
6	n_4	n_1	0.95
7	n_2	n_4	0.98
8	n_1	n_3	0.91
9	n_3	n_1	0.97
10	n_1	n_4	0.92

Transaction ID	Source	Target	Trust

Fig. 2: Reputation calculation aggregates probing transactions when the local blockchain reaches threshold w.

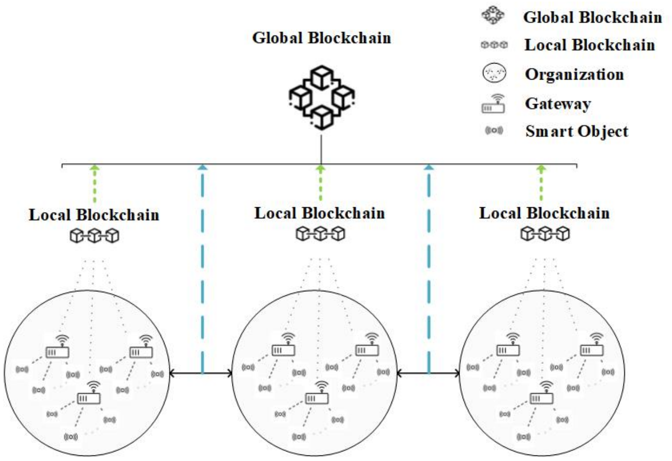


Fig. 3: Evidence-based trust evaluation method

the initial reputation value for all smart objects within the organization at system startup as 0.7, which corresponds to the Secondary Reputation Threshold (SRT). $\overline{T_j^w}$ denotes the average confidence attained by summing the confidence values of te_j across w iterations, calculated using Equation 4.

$$\overline{T_j^w} = \frac{\sum_{tr_i \in TrS_j} T_{i_j}}{|TrS_j|} \tag{4}$$

Through the smart contract's reputation value calculations, smart objects with reputations lower than the Initial Reputation Threshold (IRT) are expelled from the organization, set at $IRT = 0.5$. If an output node lacks support from over 50% of the partition nodes, its trust falls below 0.5, impacting its reputation. Nodes falling between the primary and secondary reputation thresholds are categorized as medium. Nodes whose output values estimate support from 50-70% of partition nodes, indicating less precise outputs, are classified as average and logged in the global blockchain. Smart objects are grouped using the relationship given in Equation 5.

$$C_{obj} = \begin{cases} C_{weak} \quad if R_j^w < IRT, \\ C_{medium} \quad if IRT \leq R_j^w < SRT, \\ C_{perfect} \quad if R_j^w \geq SRT. \end{cases} \tag{5}$$

2) Evidence-based method: As depicted in Figure 3, each organization comprises a set of gateway nodes $G = G_1, G_2, \ldots, G_n$ forming a lightweight blockchain. Each gateway node G_i encompasses k smart object nodes $O_i = O_{i_1}, O_{i_2}, \ldots, O_{i_k}$ and is tasked with data collection from objects. The observed data is hashed and can be stored off-chain, while aggregated transactions and interactions are recorded on the blockchain. Smart objects affiliated with the same gateway are adjacent, facilitating related observations. Furthermore, smart objects linked to a gateway offer a degree of redundancy in service provision. Every node in the network possesses a unique public-private key pair. During the network setup phase, nodes register using their public key, and transactions are signed using their private keys. Gateway nodes, holding access to registered nodes' public keys, verify these signatures.

a) Lightweight-blockchain: In [31], a lightweight blockchain based on the trust architecture of the Internet of Things is proposed, which we adhere to in our provided private blockchain. As illustrated in Figure 4, gateway nodes are responsible for block generation, block validation, and distributed consensus within the private blockchain. According to the presented method, the gateway nodes are recognized by the network and granted permission to produce blocks, eliminating the need to compete for block production using computationally expensive block mining mechanisms such as proof-of-stake and proof-of-work. Gateway nodes generate blocks at periodic intervals. Upon receiving all related smart object transactions, the gateway node validates these transactions, calculates the evidence and reputation of the node, and creates a block with transactions containing the observation data, public key, and signature of the smart object, as well as the trust value of the evidence and the reputation of the smart object. Subsequently, the block is transmitted to other blockchain nodes for validation. Block validation encompasses: 1) validation of data transactions by verifying the public keys of the smart objects and their signatures within the transactions, and 2) validation of the trust and reputation values assigned by the gateway node through recalculation with the data within the generated block.

To manage and minimize the length of the blockchain, a lightweight blockchain named Sensor-chain is introduced in [23]. In line with this approach, once the blockchain's length surpasses a predefined threshold, a gateway node is designated as the aggregator node. Subsequently, the blocks are consolidated, resulting in an aggregate block serving as the genesis block. Eventually, the reputation values of the smart objects associated with that organization are updated in the Global blockchain. Any smart object whose reputation value falls below the initial threshold is subsequently removed from the organization.

b) Calculation of trust and reputation of smart object: The observations made by neighboring smart objects can serve as evidence regarding the reliability of the observations made by a given smart object. The trust value of a smart object

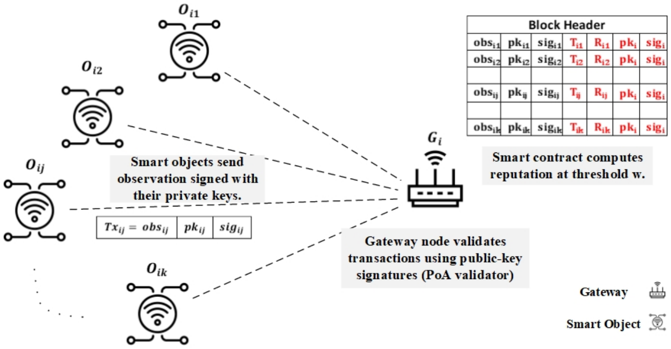

O_{i1}
O_{i2}
O_{ij}
O_{ik}
G_i
Block Header
obs_{i1} pk_{i1} sig_{i1} T_{i1} R_{i1} pk_i sig_i
obs_{i2} pk_{i2} sig_{i2} T_{i2} R_{i2} pk_i sig_i
obs_{ij} pk_{ij} sig_{ij} T_{ij} R_{ij} pk_i sig_i
obs_{ik} pk_{ik} sig_{ik} T_{ik} R_{ik} pk_i sig_i
Smart objects send observation signed with their private keys.
$Tx_{ij} = obs_{ij}$ pk_{ij} sig_{ij}
Smart contract computes reputation at threshold w.
Gateway node validates transactions using public-key signatures (PoA validator)
Gateway
Smart Object

Fig. 4: Block generation in the PoA-based model. Gateway nodes validate transactions and submit blocks once the local threshold w is reached.

is instantaneous for each observation. Denoted as obs_{i_j}, it represents the observation made by the smart object O_{i_j} belonging to the gateway G_i. N_{i_j} denotes the set of adjacent smart objects to O_{i_j}. Based on the observations of neighboring smart objects, a pruning process is conducted on N_{i_j} to eliminate smart objects with outlier data, resulting in the set $\widehat{N_{i_j}}$. Subsequently, the trust value T_{i_j} of the smart object is calculated using Equation 1.

The level of reputation reflects the long-term performance of the smart object and significantly influences the organization's assessment. The reputation value $R_{i_j}^t$ for the smart object O_{i_j} is determined as follows:

The trust and reputation values associated with the node in transactions involving the smart object are logged in the local blockchain. Initially, all smart objects are assigned a reputation value of 0.7. Once the local blockchain reaches its length threshold w, it is consolidated, and the most recent reputation values of the smart objects within that organization are updated in the global blockchain via the smart contract. Any smart object with a reputation value below the initial threshold is expelled from the organization. Smart objects are categorized based on the relationship defined in Equation 5.

B. Inter-organizational layer

Organizations function as both service providers and requesters within the network. Each organization is registered in the global blockchain with a unique public and private key. Information such as the services offered, the list of associated smart objects, and their reputations is stored in the global blockchain using a smart contract. Inter-organizational trust is established through direct observations made by each organization regarding others, as well as indirect recommendations received from other organizations.

$$R_{i_j}^t = \lambda . R_{i_j}^{t-1} + (1-\lambda) T_{i_j} \tag{6}$$

In the probing-based method, when organization q requests a service, it selects an intelligent object with a high reputation as an intermediary node. This intermediary node then requests the service from the target node and activates the probing mechanism to assess the trustworthiness of the target object based on the response provided by the smart contract in the local blockchain. The resulting trust value is then conveyed to the requesting organization. Conversely, in the evidence-based method, the gateway node relays the response to the requesting organization using the values it has already collected from the smart objects.

Whenever organization k requests service s_m at time t, the organizations offering service s_m are initially identified using the smart contract. Subsequently, based on the category of the requesting organization, the corresponding service provider organization is selected. The trust between organization k and organization q is then evaluated and updated with respect to service s_m as follows:

$$T_{k_q}^{s_m}(t) = \alpha . T_{k_q}^{s_m}(t-1) + \beta . TD_{k_q}^{s_m}(t) + \gamma . TR_{k_q}^{s_m}(t) \tag{7}$$

Where, α, β, and γ represent weight parameters within the range [0,1], satisfying the constraint $\gamma + \beta + \alpha = 1$. $TD_{k_q}^{s_m}(t)$ denotes the direct observation of organization k regarding organization q at time t, defined as follows:

$$TD_{k_q}^{s_m}(t) = \begin{cases} \mu . TD_{k_q}^{s_m}(t-1) + (1-\mu) R_{i_j}^q . \Lambda_q^w \\ \quad if TD_{k_q}^{s_m}(t-1) \neq \varnothing, \\ R_{i_j}^q . \Lambda_q^w \quad otherwise. \end{cases} \tag{8}$$

Where, $R_{i_j}^q$ represents the reputation value of smart object O_{i_j} affiliated with organization q, which offers the service s_m. The parameter Λ_q^w is calculated as follows:

$$\Lambda_q^w = \begin{cases} \theta . \Lambda_q^{w-1} + (1-\theta) \mathcal{J}(q^w, q^{w-1}) \quad if \Lambda_q^{w-1} \neq \varnothing, \\ 1 \quad otherwise. \end{cases} \tag{9}$$

The index $\mathcal{J}(q^w, q^{w-1})$ measures the degree of change in organization q compared to the previous aggregation. It is computed by evaluating the Jaccard coefficient between the sets of smart objects in organization q during aggregations w and $w-1$, as defined in Equation 10.

$$\mathcal{J}(q^w, q^{w-1}) = \frac{|q^w \cap q^{w-1}|}{|q^w \cup q^{w-1}|} \tag{10}$$

The indirect recommendation, denoted as $TR_{k_q}^{s_m}(t)$, stems from other organizations that have previously interacted with organization q. It is recalculated and updated using Algorithm 1 following each interaction. Additionally, the overall trust value of the organization is determined by its collective service provision, computed through the smart contract. Based on the resultant total trust value $T_q(t)$, the organization is categorized as follows:

$$C_{org_q} = \begin{cases} BlackList \quad if T_q(t) < 0.35, \\ GrayList \quad if 0.35 \leq T_q(t) < 0.7, \\ WhiteList \quad if T_q(t) \geq 0.7. \end{cases} \tag{11}$$

The smart contract aggregates trust scores from probing transactions, applies a weighted reputation update (Eq. 3), and classifies nodes into Perfect, Medium, or Weak trust levels.

Blacklisted organizations will face suspension and restrictions on both providing and requesting services, primarily due to the heightened risk of collusion among malicious smart objects. Greylisted organizations are restricted to receiving services solely from organizations within their category, although

Algorithm 1 Smart contract for calculating the trust value of organization k with respect to organization q

```
Require: $k$: Service requester organization, $q$: Service provider organization, $s_m$: Service, $org_N$: Set of all organizations
$org_N \leftarrow org_N - \{k, q\}$
Compute direct trust $TD_{k_q}^{s_m}$ using Equation (8)
$T_{k_q}^{s_m}(t) \leftarrow \alpha \cdot T_{k_q}^{s_m}(t-1) + \beta \cdot TD_{k_q}^{s_m}(t) + \gamma \cdot TR_{k_q}^{s_m}(t)$
$T_q(t) \leftarrow \nu \cdot T_q(t-1) + (1-\nu) \cdot T_{k_q}^{s_m}(t)$
Classify organization $q$ using Equation (11)
for $org_i \in org_N$ do
    $TR_{i_q}^{s_m}(t) \leftarrow \rho \cdot TR_{i_q}^{s_m}(t-1) + (1-\rho) \cdot T_{k_q}^{s_m}(t)$
end for
```

they retain the ability to provide services to organizations in the same category or those on the whitelist. White list organizations, on the other hand, encounter no limitations in their interactions and are free to both provide and receive services without restriction.

IV. EXPERIMENTAL ANALYSIS

The efficiency and accuracy of the two proposed methods are evaluated using the Confusion Matrix. Simulations are conducted using Python and Solidity programming languages within the Ganache test network. The confusion matrix serves as a valuable instrument for assessing the effectiveness of classification algorithms, particularly in the context of evaluating trust management models. In the two proposed methods, samples are categorized into two groups: reliable nodes and malicious nodes. The parameters of the confusion matrix are outlined as follows [17]:

1) True Positive (TP): Instances where trusted nodes were correctly identified as trusted.
2) True Negative (TN): Instances where malicious nodes were correctly identified as malicious.
3) False Positive (FP): Instances where malicious nodes were incorrectly identified as trusted.
4) False Negative (FN): Instances where trusted nodes were incorrectly identified as malicious.

Based on the obtained parameters, the following evaluation metrics can be computed to assess the performance of the trust management system.

1) Accuracy: This metric assesses the overall correctness of predictions and is computed using Equation 12.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

2) Precision: It represents the proportion of true positive predictions among all positive predictions and is determined by Equation 13.

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

3) Recall: This metric demonstrates the model's ability to correctly identify positive samples and is calculated as per Equation 14.

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

4) Specificity: It indicates the model's capability to accurately recognize negative samples and is computed as per Equation 15.

$$Specificity = \frac{TN}{TN + FP} \tag{15}$$

5) F-measure: This metric serves as a comprehensive evaluation of accuracy and recall and is utilized as a holistic measure to evaluate model performance, calculated using Equation 16.

$$F - measure = \frac{2PrecisionRecall}{Precision + Recall} \tag{16}$$

A. Determining Evaluation Parameters

1) Probing-Based Method: Initially, the optimal reputation weight parameter is identified through analysis of the confusion matrix to ensure accurate detection of malicious nodes within the organization. Subsequently, the optimal value of the probing transaction probability is assessed.

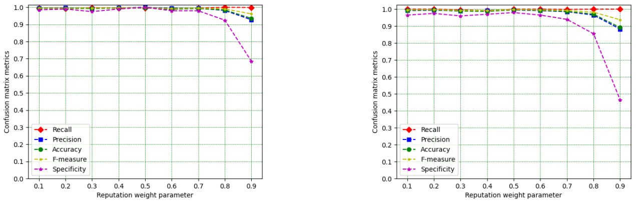
a) Optimal Reputation Weight Parameter: The reputation of the smart object is computed based on formula 3. Simulation parameters, as outlined in Table II, are configured to ascertain the optimal weight parameter. Initially, malicious nodes in the evaluation model are assumed to be consistently malicious, after which they are assessed for on-off attacks. These simulations involve 100 smart objects, among which 20% are considered malicious. Other key parameters, including the number of partitions, initial reputation values, and the probing transaction probability, are adjusted accordingly to assess the optimal reputation weight parameter.

TABLE II: Simulation Parameters for Determining Optimal Reputation Weight in Probing-Based Method

Parameters	Values
Number of Smart Objects	100
Malicious Smart Objects	20%
Partitions	10
Initial Reputation Value	0.7
Initial Reputation Threshold	0.5
Probing Transaction Probability	1
Local Blockchain Length Threshold	5
Number of Runs	10

Figure 5 illustrates the impact of increasing the reputation weight on the confusion matrix metrics when considering permanently malicious nodes and on-off attack modes. A reputation weight value of $\alpha = 0.1$ indicates that past reputation values have less influence on the smart object's reputation outcome, while new trust assessments have a greater impact. Analysis of the results in Tables 5a and 5b reveals that up to a weight of $\alpha = 0.8$ for the permanent destructive mode and $\alpha = 0.7$ for the on-off attack mode, the specificity metric consistently exceeds 0.9. Lower reputation weight for historical data correlates with higher model accuracy.

Figure 6 illustrates the reputation trajectories and the average number of trusted and malicious nodes within the organization per transaction under the persistent malicious



(a) Permanent Malicious Mode

(b) On-Off Attack Mode

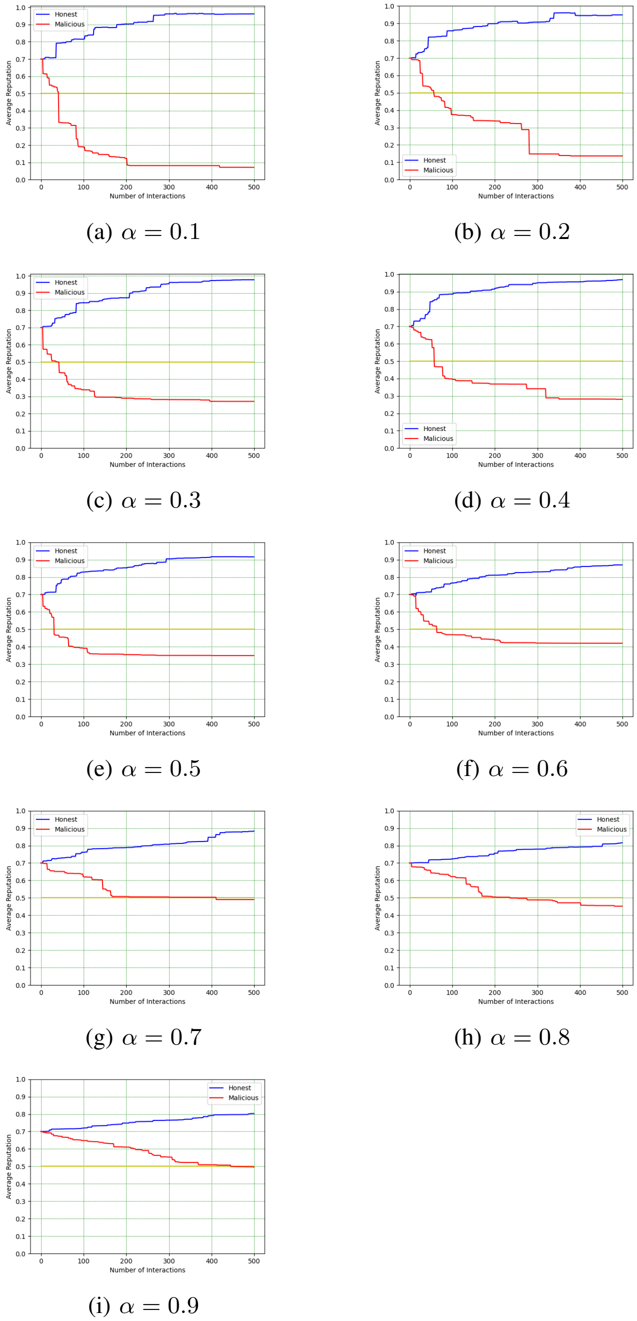
Fig. 5: Evaluation of Optimal Reputation Weight of Smart Object in Probing-Based Method

mode, across a range of reputation weights from $\alpha = 0.1$ to $\alpha = 0.9$. As the weight assigned to past data increases, more transactions between smart objects are needed to identify malicious nodes, since historical behavior is emphasized more heavily. This has significant implications depending on the deployment context. In low-interaction environments—such as smart homes, where devices exchange only a few messages per hour—a lower weight ($\alpha \approx 0.2$–0.4) is advisable to emphasize recent behavior and allow for gradual reputation decay, reducing false positives from isolated malicious actions. Conversely, in high-throughput settings like smart city sensor networks or industrial automation, where hundreds of interactions occur per minute, a higher weight ($\alpha \approx 0.6$–0.8) enables rapid detection and isolation of persistently misbehaving nodes. Notably, lower α values also lead to malicious nodes with fewer interactions maintaining an average reputation below the initial threshold, thereby expediting trust establishment for honest nodes. Thus, tuning α to the network's interaction density strikes a crucial balance between resilience and responsiveness in trust evaluation.

These reputation trajectories illustrate the effect of the weight parameter α on the speed of trust updates. For instance, in a smart city scenario, setting $\alpha = 0.2$ enables rapid detection of misbehaving traffic sensors. This helps mitigate the risk of false congestion reports, which could otherwise cause inefficient route planning or traffic mismanagement.

Figure 7, demonstrates the on-off attack mode, the average count of trusted and malicious nodes per transaction within the organization varies across different reputation weights $\alpha = [0.1, 0.9]$. It's observed that as the reputation weight increases, there is a higher fluctuation in the average reputation of malicious nodes, potentially leading to undetected malicious nodes.

To swiftly identify malicious nodes and also foster trust in honest nodes during subsequent interactions, we propose a reward and penalty mechanism to assign value to the reputation weight of the smart object. As per Equation 17, the reputation weight is determined based on the previous reputation value of the smart object compared to the average value of the new trust. If the previous reputation exceeds the average trust value, indicating a decline in trustworthiness, the reputation weight is set to 0.1. Conversely, if the average trust value surpasses the previous reputation, signifying improved performance in interactions, the reputation weight is set to 0.9.



(a) $\alpha = 0.1$

(b) $\alpha = 0.2$

(c) $\alpha = 0.3$

(d) $\alpha = 0.4$

(e) $\alpha = 0.5$

(f) $\alpha = 0.6$

(g) $\alpha = 0.7$

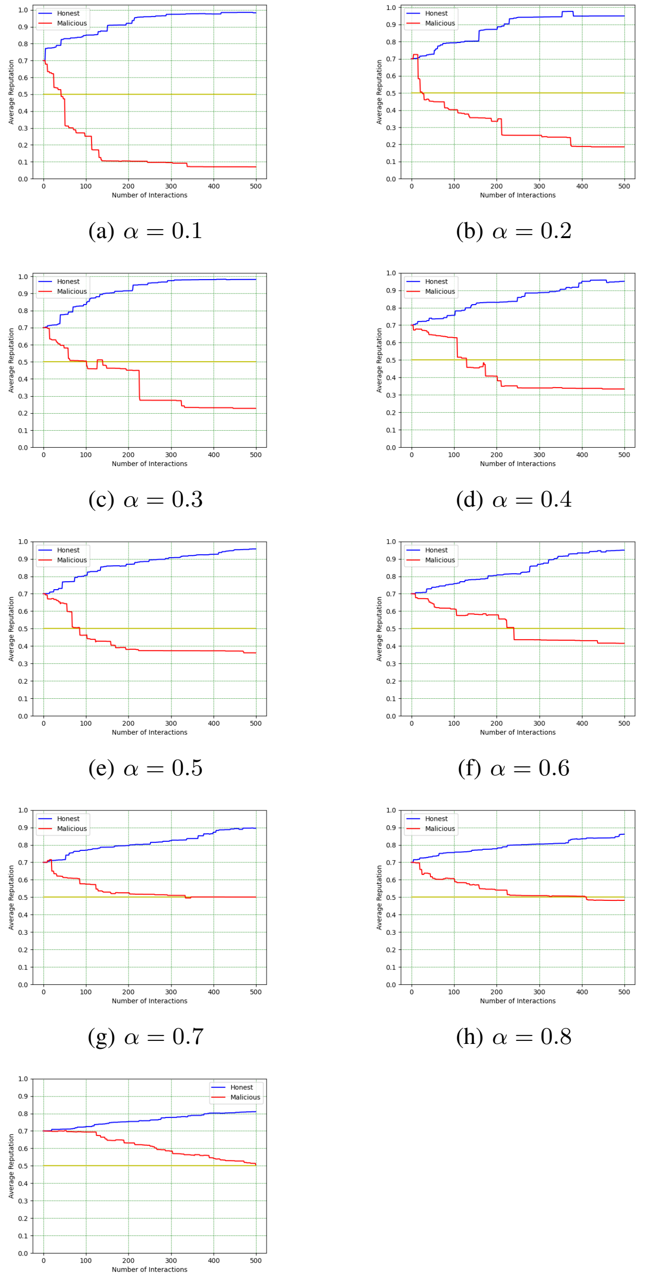
(h) $\alpha = 0.8$

(i) $\alpha = 0.9$

Fig. 6: Examining the Behavior of Persistent Honest and Malicious Nodes with Varying Reputation Weight Parameters in a Probing-Based Method

$$\alpha = \begin{cases} 0.9 & if \qquad R_j^{w-1} < \overline{T_j^w}, \\ 0.1 & otherwise. \end{cases} \tag{17}$$

The behavior of honest and malicious nodes under the reward and punishment mechanism is illustrated in Figure 8. In this mode, honest nodes gradually increase their reputation through consistent, trustworthy interactions. Conversely, malicious nodes experience a decline in reputation once their behavior falls below a defined threshold, especially during



(a) $\alpha = 0.1$

(b) $\alpha = 0.2$

(c) $\alpha = 0.3$

(d) $\alpha = 0.4$

(e) $\alpha = 0.5$

(f) $\alpha = 0.6$

(g) $\alpha = 0.7$

(h) $\alpha = 0.8$

(i) $\alpha = 0.9$

Fig. 7: Examining the Behavior of Honest and Malicious Nodes in the On-Off Attack Mode with Varying Reputation Weight Parameters in the Probing-Based Method

limited interactions.

To evaluate the effectiveness of this mechanism, we compared it against a baseline with a static weighting factor $\lambda = 0.5$. Our adaptive mechanism exhibits significantly faster detection of malicious behavior compared to the static model. This enhanced responsiveness is critical for real-time IoT environments such as smart transportation or industrial automation.

The initial reputation value is used as the starting point, and the system dynamically adjusts trust scores based on accumulated positive or negative behavior. As such, this mechanism enhances resilience against both persistent and intermittent malicious behavior patterns.

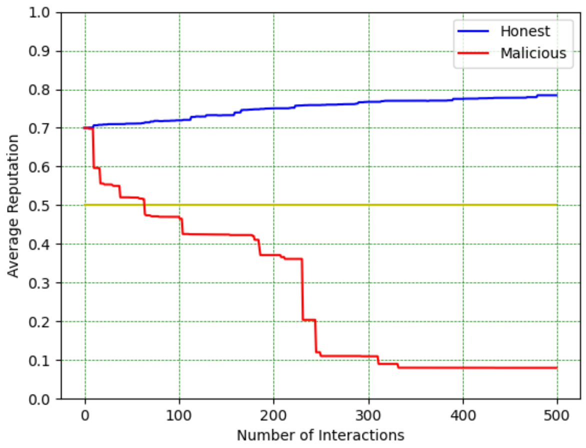


Fig. 8: Examining the Behavior of Honest and Malicious Nodes with Reward and Punishment Mechanism in Probing-Based Method

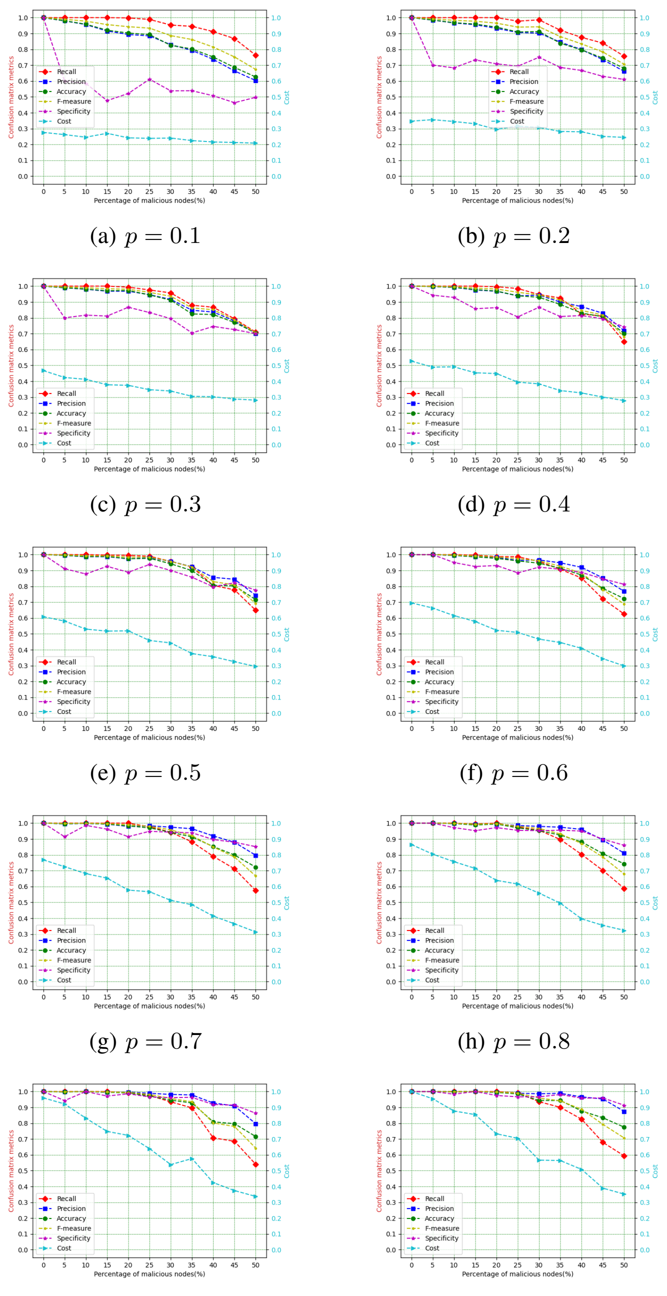
b) Optimal Probing Transaction Probability: In the probing-based method, the reliability of smart objects is assessed through a probing mechanism, typically involving probe queries. However, conducting probing transactions can be costly. To alleviate system overhead and enhance efficiency, probing experiments are randomly conducted with a certain probability. Simulation parameters, as detailed in Table III, are configured to determine the optimal probing transaction probability.

TABLE III: Simulation Parameters for Determining Optimal Probing Transaction Probability in Probing-Based Method

Parameter	Value
Number of Smart Objects	100
Percentage of Malicious Nodes	$[5\%, 50\%]$
Partitions	10
Initial Reputation Value	0.7
Initial Reputation Threshold	0.5
Reputation Weight Value (α)	Equation 17
Local Blockchain Length Threshold	5
Number of Runs	10

In Figure 9, the impact of increasing the number of malicious nodes within the organization on the confusion matrix metrics is illustrated for different probing transaction probabilities $p = [0.1, 1]$. The results indicate that lower probabilities lead to reduced system accuracy and inaccurate detection of malicious nodes. Conversely, higher probabilities enhance system accuracy but also increase system cost and overhead. This cost is quantified by the number of packets sent to perform the probing transaction, calculated as c_k in partition p_{i_j}, as described in Equation 18.

$$Cost_{i_j} = \sum_{k=1}^{\widehat{p_{i_j}}} c_k \tag{18}$$



(a) $p = 0.1$

(b) $p = 0.2$

(c) $p = 0.3$

(d) $p = 0.4$

(e) $p = 0.5$

(f) $p = 0.6$

(g) $p = 0.7$

(h) $p = 0.8$

(i) $p = 0.9$

(j) $p = 1$

Fig. 9: Evaluation of Performance and Cost of Probing-Based Method with Different Probing Transaction Probabilities

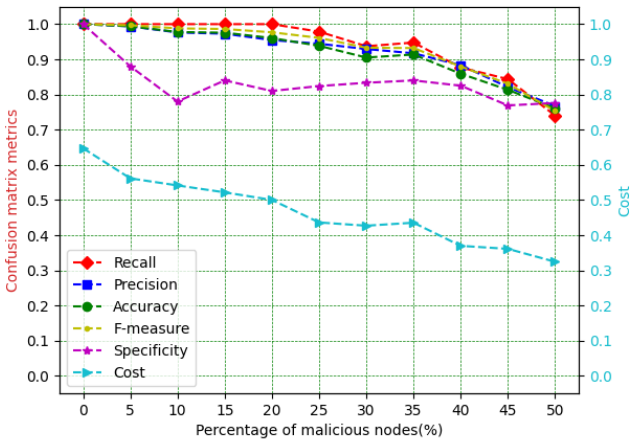


Fig. 10: Evaluation of Efficiency and Cost of the Probing-Based Method with Probing Probability $p = 0.5$ in On-Off Attack Mode

TABLE IV: Simulation Parameters for Determining Optimal Reputation Weight in Evidence-Based Method

Parameters	Values
Number of Smart Objects	100
Malicious Smart Objects	20%
Gateway Nodes	10
Initial Reputation Value	0.7
Initial Reputation Threshold	0.5
Reputation Weight Parameter (λ)	$[0.1, 0.9]$
Local Blockchain Length Threshold	20
Number of Runs	10

To optimally determine the probing transaction probability, we establish threshold values for both the specificity metric and the cost. The specificity metric threshold is set to 0.9, indicating that we aim to correctly detect 90% of malicious nodes. Additionally, the cost criterion threshold is set to 0.6. Based on these threshold definitions, the optimal probing transaction probability occurs at $p = 0.5$. At this probability value, nearly 30% of malicious nodes exhibit specificity values exceeding 0.9, while simultaneously maintaining a cost criterion value lower than 0.6. Therefore, selecting $p = 0.5$ as the probing transaction probability achieves the desired balance between effectively detecting malicious nodes and minimizing the associated cost.

By determining the optimal probing probability as $p = 0.5$, Figure 10 illustrates the impact of increasing malicious nodes in the on-off attack mode on the confusion matrix criteria. Notably, the specificity metric indicates that up to 40% of malicious nodes exhibit values above 0.8.

2) Evidence-based method: The reputation of the smart object is computed in the evidence-based method using Formula 6. To ascertain the optimal reputation weight, simulation parameters are configured based on Table IV. This section investigates the detection of malicious nodes in the on-off attack mode to determine the optimal reputation weight.

In Figure 11, the impact of varying the reputation weight on the metrics of the confusion matrix in the on-off attack mode is illustrated. Generally, as the reputation weight decreases, the system's accuracy increases, leading to a higher rate of correct detection of malicious nodes. The results indicate that for reputation weights up to $\lambda = 0.6$, the specificity metric value

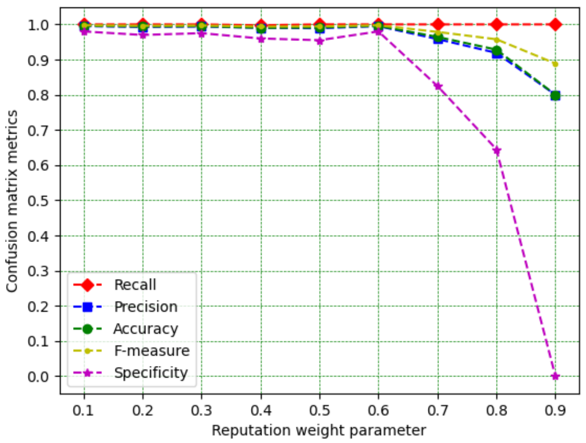


Fig. 11: Investigating Optimal Reputation Weight of Smart Object in Evidence-Based Method

remains consistently above 0.9. This suggests that a lower reputation weight enhances the system's ability to accurately identify malicious nodes, as evidenced by the high specificity metric values observed within this range.

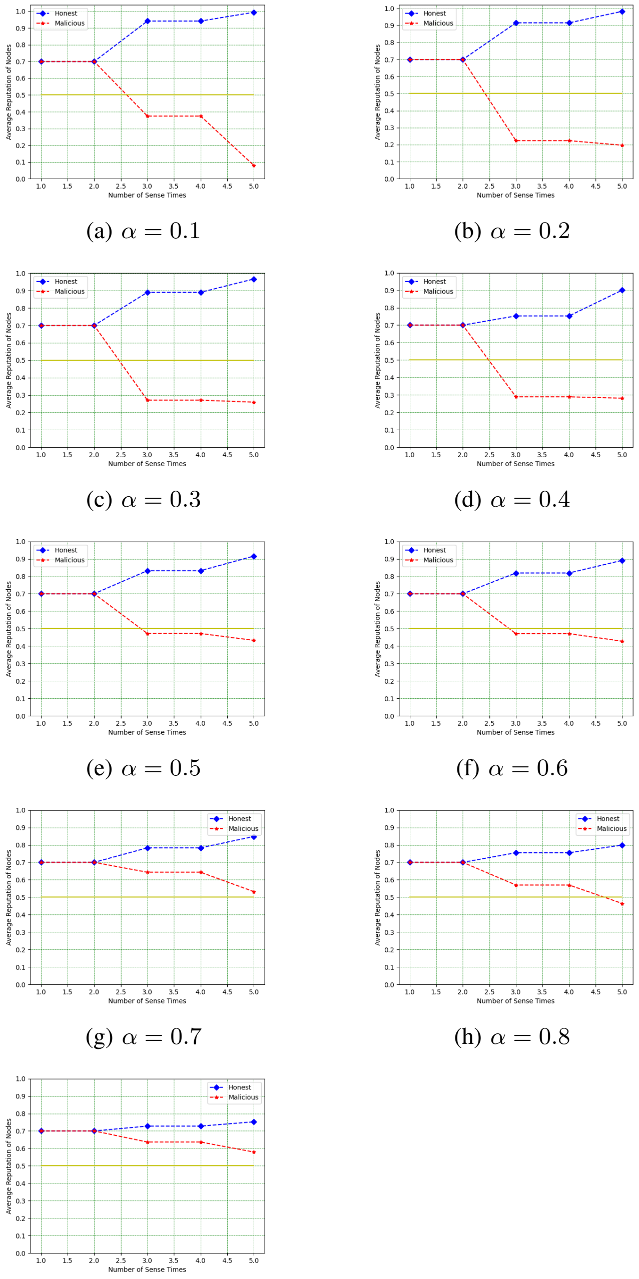
In Figure 12, which pertains to the on-off attack mode, it illustrates the average count of both honest and malicious nodes within the organization each time the gateway node gathers information from smart objects. Generally, as the reputation weight increases, there is a higher likelihood of detecting malicious nodes during data collection while also fostering quicker trust establishment for honest nodes. As depicted in Figure 13, a similar approach to the previous method is employed here, utilizing a reward and punishment mechanism to expedite the identification of malicious nodes and enhance trust in honest nodes during interactions.

In Figure 14, we observe the impact of increasing malicious nodes on the confusion matrix metrics in both permanent malicious mode and on-off attack scenarios, employing a reward and penalty mechanism. Specifically, in Figure 14a, for the permanent malicious mode, the specificity metric reaches one for up to 50% of malicious nodes, indicating accurate detection. However, the recall value exceeds 0.9 for up to 35% of malicious nodes, after which the likelihood of false positives increases, leading to honest nodes being misidentified as malicious. In Figure 14b, corresponding to the on-off attack mode, the recall metric indicates accurate detection for up to 40% of malicious nodes.

B. Evaluation of Results

Based on the results obtained and the specified parameter values, we assess and compare the performance of the two proposed methods.

According to Table V and Figure 15, the performance of the two proposed methods (Evidence and Probing) is evaluated against three baseline approaches [27], [30], and [32], based on confusion matrix metrics—precision, specificity, and recall—under varying proportions of malicious nodes (from 0% to 50%).



(a) $\alpha = 0.1$ (b) $\alpha = 0.2$

(c) $\alpha = 0.3$ (d) $\alpha = 0.4$

(e) $\alpha = 0.5$ (f) $\alpha = 0.6$

(g) $\alpha = 0.7$ (h) $\alpha = 0.8$

(i) $\alpha = 0.9$

Fig. 12: Investigating the Behavior of Honest and Malicious Nodes in the On-Off Attack Mode with Varying Reputation Weight Parameters in Evidence-Based Method

The Evidence-based method (Figure 15a) consistently achieves perfect precision and specificity (value of 1.0) across all scenarios, indicating strong resistance to false positives and a robust ability to accurately classify trustworthy nodes. However, its recall gradually degrades beyond the 35% threshold, reaching 0.54 at 50% malicious nodes due to an increase in false negatives, likely caused by collusion effects. This trend underscores the importance of maintaining an honest majority, in line with Assumption 1. Additionally, the operational cost of this method declines as malicious nodes are eliminated,

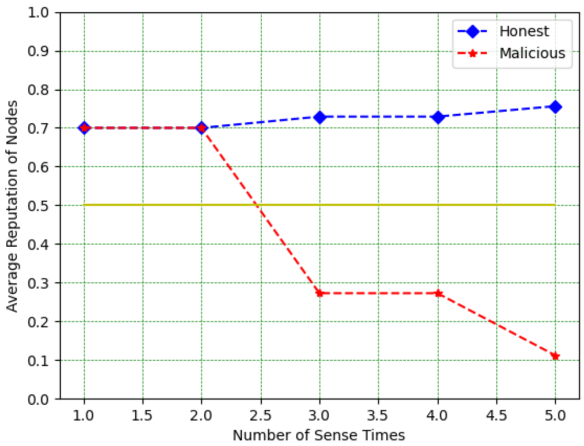
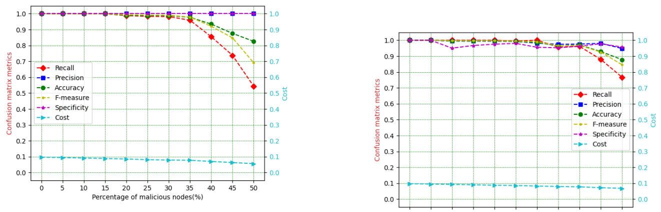


Fig. 13: Investigating the Behavior of Honest and Malicious Nodes with Reward and Punishment Mechanism in Evidence-Based Method



(a) Permanent Malicious Mode: Specificity remains above 1 up to 50% malicious nodes, but Recall drops at 35%.

(b) On-Off Attack Mode: Specificity remains above 0.9 up to 50% malicious nodes, but Recall drops at 45%.

Fig. 14: Detection performance under (a) permanent malicious mode and (b) on-off attack mode. While recall in (b) drops below 0.9 at 35% malicious nodes per round, the overall detection success across attack cycles exceeds 90% due to redundancy and pruning mechanisms.

reducing unnecessary interactions.

The Probing-based method (Figure 15b), with a full probing probability ($p = 1$), also shows strong and balanced performance, maintaining precision and specificity above 0.9 even at 50% malicious nodes. It demonstrates slightly better recall than the Evidence method at high malicious node levels, indicating higher resilience. However, this robustness comes at the cost of increased network overhead due to exhaustive probing. To mitigate this, subsequent experiments apply randomized probing with optimized probability to balance accuracy and efficiency.

In comparison with the baseline methods, the Evidence approach significantly outperforms all in both precision and specificity. While the method in [32] achieves perfect recall, its precision and specificity remain noticeably lower. The other two baselines, particularly [30], show a substantial drop in all metrics as the malicious node ratio increases. For instance, at 50% malicious nodes, its precision and specificity decline to 0.26 and 0.10, respectively, reflecting poor robustness in adversarial settings. As further illustrated in Figure 15c, the method in [27] experiences a drop in specificity below 0.9 even beyond 20% malicious nodes.

In summary, both proposed methods—especially the Evidence-based approach—offer high accuracy and robustness in trust classification, making them suitable for deployment in security-critical IoT environments where minimizing false positives is essential.

These results demonstrate that in smart city scenarios, where thousands of heterogeneous IoT devices (e.g., traffic sensors, streetlights, surveillance units) must collaborate securely, the proposed methods remain robust even as up to 50% of nodes act maliciously. This is particularly vital to prevent cascading failures from compromised edge devices.

The assumption of service redundancy—requiring 2–3 redundant smart objects per partition—is designed to ensure reliable trust evaluation within each service group. In a network of 100 nodes, for instance, with 10 partitions, this results in approximately 20–30 redundant nodes across the system. In a larger network with 1,000 nodes and a proportional increase to 100 partitions, maintaining 2–3 redundant nodes per partition would require 200–300 nodes—preserving the same relative distribution. Therefore, this assumption scales effectively with network size, provided that the number of service partitions increases proportionally. Clustering and partitioning mechanisms help ensure that redundancy is preserved even in highly distributed and large-scale IoT environments.

In a smart home deployment with 100 IoT devices—such as motion sensors, cameras, thermostats, and door locks—a 30% malicious node ratio means 30 compromised devices. Our trust framework maintains 98–100% precision under this condition, ensuring critical security devices are not misclassified or disabled, thus preserving system reliability and occupant safety.

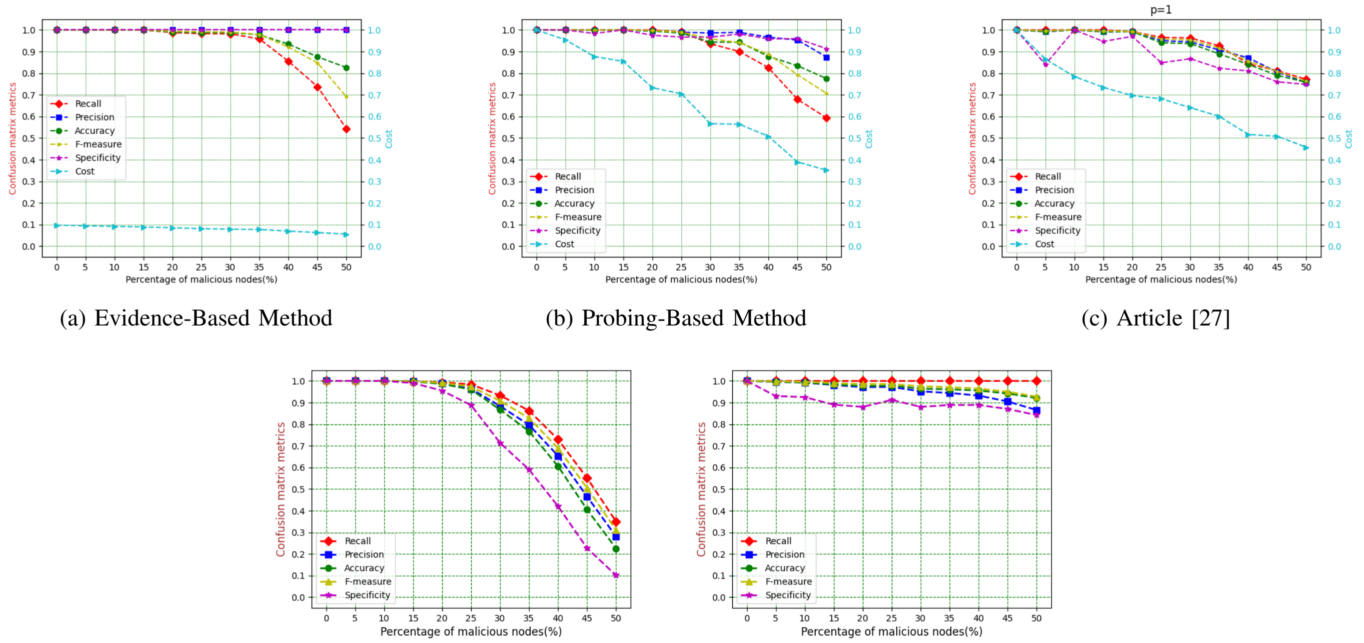
In Figure 16, the trust dynamics of the organization within the inter-organizational layer are explored across various proportions of malicious nodes. Organizations containing more than 35% malicious nodes are deemed malicious, with trust values below the 0.35 threshold warranting classification as blacklisted entities, subject to suspension. Organizations with trust values ranging from 0.35 to 0.7 fall into the gray-listed, while those exceeding the 0.7 threshold are classified as white-listed entities. The Jaccard coefficient undergoes variations with each aggregation within the blockchain when malicious nodes are present in the organization. If organizational changes surpass 50%, the likelihood of collusion increases, potentially leading to honest nodes being misidentified as malicious. Consequently, organizations harboring more than 35% malicious nodes are blacklisted. Conversely, in organizations with less than 35% malicious nodes, the elimination of malicious nodes during subsequent aggregation stages boosts the Jaccard coefficients, ultimately elevating the trust level within the organization.

V. Scalability Analysis

To evaluate the scalability of the proposed model, we first consider the one-layer blockchain configuration. Equa-

TABLE V: Comparison of the performance of the proposed methods (Evidence and Probing) and baseline approaches based on precision, specificity, and recall metrics with increasing percentages of malicious nodes.

	Precision					**Specificity**					**Recall**				
Malicious%	**Evidence**	**Probing**	**[27]**	**[30]**	**[32]**	**Evidence**	**Probing**	**[27]**	**[30]**	**[32]**	**Evidence**	**Probing**	**[27]**	**[30]**	**[32]**
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	0.98	1	1	1	1	0.85	1	0.93	1	1	1	1	1
10	1	0.99	1	1	0.99	1	0.98	1	1	0.92	1	1	1	1	1
15	1	1	0.97	1	0.98	1	1	0.95	0.99	0.89	1	1	1	1	1
20	1	0.98	0.97	0.99	0.97	1	0.97	0.98	0.95	0.88	0.98	1	0.99	0.99	1
25	1	0.97	0.95	0.96	0.97	1	0.96	0.85	0.88	0.91	0.98	0.99	0.97	0.98	1
30	1	0.97	0.94	0.88	0.95	1	0.96	0.87	0.71	0.88	0.97	0.93	0.97	0.93	1
35	1	0.98	0.92	0.79	0.94	1	0.98	0.82	0.59	0.89	0.96	0.90	0.93	0.86	1
40	1	0.96	0.87	0.65	0.93	1	0.96	0.81	0.42	0.89	0.86	0.83	0.85	0.72	1
45	1	0.95	0.81	0.46	0.90	1	0.96	0.76	0.22	0.87	0.73	0.68	0.81	0.55	1
50	1	0.87	0.76	0.26	0.87	1	0.91	0.75	0.10	0.84	0.54	0.59	0.78	0.35	1



(a) Evidence-Based Method

(b) Probing-Based Method

(c) Article [27]

(d) Article [30]

(e) Article [32]

Fig. 15: Precision, recall, and specificity under increasing malicious nodes for (a) evidence-based, (b) probing-based, (c) Corradini et al. [27], (d) Kouicem et al. [30], and Amiri et al. [32] methods.

tion 19 quantifies the number of blocks created in the global blockchain under this configuration:

$$\text{Height}_{\text{One-Layer}}(t) = \sum_{n=1}^{\text{Org}} |G_n| \cdot t \cdot SR_n \tag{19}$$

In Equation 19, SR_n denotes the data collection rate from smart objects by the gateway node, $|G_n|$ represents the number of gateway nodes in each organization, and t is the elapsed time (in minutes). This equation provides insight into the scalability of the system by accounting for the data generation rate and the number of gateway nodes over time, thus highlighting the potential for exponential growth in the blockchain size as the network scales.

To mitigate the rapid increase in transaction volume and reduce the global blockchain size, a hierarchical two-level blockchain architecture is implemented. In this model, transactions are first recorded on a lightweight, temporary local blockchain. Once the local blockchain reaches a predefined threshold length w_n, the aggregated transactions are transferred to the global blockchain. The scalability of this two-layer approach is captured in Equation 20:

$$\text{Height}_{\text{Two-Layer}}(t) = \sum_{n=1}^{\text{Org}} \left\lfloor \frac{|G_n| \cdot t \cdot SR_n}{w_n} \right\rfloor \tag{20}$$

Here, w_n is the threshold length for each organization's local blockchain. By aggregating transactions locally, the model reduces the frequency of updates to the global blockchain, thereby lowering storage and processing overhead. A higher w_n diminishes the global blockchain's growth, enhancing scalability. However, this benefit comes at the cost of delayed updates to smart object reputations, which may affect the accuracy of trust evaluations.

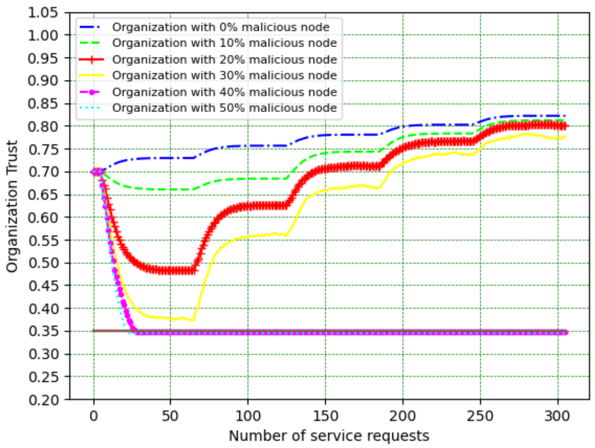


Fig. 16: Investigating Inter-Organizational Interactions with Varying Percentages of Malicious Nodes within Organizations

a) Practical Example:: To ground the above equation in a real-world context, consider a smart city deployment with $|G_n| = 10$ gateway nodes, each collecting trust-related data at a rate of $SR_n = 100$ transactions per minute. For a time interval of $t = 10$ minutes, the total number of transactions is:

$$|G_n| \times SR_n \times t = 10 \times 100 \times 10 = 10{,}000.$$

In a traditional one-layer blockchain setup, all 10,000 transactions would be appended to a single chain. However, using the proposed two-layer model with a local block aggregation threshold of $w_n = 100$, each gateway produces one block every 100 transactions, resulting in:

$$\left\lfloor \frac{10 \times 100 \times 10}{100} \right\rfloor = 100 \text{ global blocks},$$

which constitutes a 100× reduction in global blockchain height. This example demonstrates the effectiveness of our hierarchical design in significantly reducing storage overhead and synchronization frequency while preserving trust granularity.

b) Practical Example:: To ground the above equation in a real-world context, consider a smart city deployment with $|G_n| = 10$ gateway nodes, each collecting trust-related data at a rate of $SR_n = 100$ transactions per minute. For a time interval of $t = 10$ minutes, the total number of transactions is:

$$|G_n| \times SR_n \times t = 10 \times 100 \times 10 = 10{,}000.$$

In a traditional one-layer blockchain setup, all 10,000 transactions would be appended to a single chain. However, using the proposed two-layer model with a local block aggregation threshold of $w_n = 100$, each gateway produces one block every 100 transactions, resulting in:

$$\left\lfloor \frac{10 \times 100 \times 10}{100} \right\rfloor = 100 \text{ global blocks},$$

which constitutes a 100× reduction in global blockchain height. This example demonstrates the effectiveness of our hierarchical design in significantly reducing storage overhead and synchronization frequency while preserving trust granularity.

c) Trade-off Discussion:: While increasing the local blockchain threshold w_n significantly reduces the frequency of global updates—thereby improving scalability and reducing latency—it also introduces a delay in global reputation updates. For example, as illustrated in Figure 18, using $w_n = 100$ achieves approximately a 3× reduction in transaction latency. However, this also implies that reputation values are updated only after 100 local trust transactions have been collected. In rapidly evolving IoT environments, such as vehicular networks or industrial automation, this delay may hinder timely detection of malicious nodes or abnormal behavior. Consequently, selecting an appropriate w_n requires careful balancing between performance optimization and trust responsiveness, depending on the time-sensitivity and risk profile of the deployment context.

To illustrate the practical scalability benefits, consider a smart city with $|G_n| = 10$ gateways, each producing 100 transactions per minute. Over 10 minutes, the one-layer model yields 10,000 blocks, while the two-layer model (with $w_n = 100$) reduces this to just 100 global blocks—a 100× reduction. As shown in Figure 18, latency reduction saturates at $w_n \approx 100$, beyond which local trust aggregation delays become significant. This highlights the trade-off between scalability and responsiveness: larger w_n improves throughput and storage efficiency but may delay reputation updates in time-sensitive networks. The use of PoA for local chains ensures scalability without requiring expensive global consensus at every step.

A. *Storage Efficiency Improvement*

The reduction factor in storage growth can be expressed as:

$$\text{Reduction Factor} = \frac{\text{Height}_{\text{One-Layer}}(t)}{\text{Height}_{\text{Two-Layer}}(t)} \tag{21}$$

Substituting the expressions from Equations 19 and 20:

$$\text{Reduction Factor} = \frac{\sum |G_n| \cdot t \cdot SR_n}{\sum \left\lfloor \frac{|G_n| \cdot t \cdot SR_n}{w_n} \right\rfloor} \tag{22}$$

For large values of w_n, the reduction factor approaches approximately w_n. Thus, storage requirements are reduced by roughly a factor of w_n. For example:

- If $w_n = 10$, the global blockchain storage is reduced by approximately 10x.
- If $w_n = 100$, the reduction is about 100x.

This confirms that the two-layer blockchain scales better since the global blockchain grows much slower than in a traditional one-layer blockchain.

Figure 17 demonstrates that employing a two-layer architecture with a predefined threshold results in a significant reduction in the growth of the blockchain height, thereby decreasing overall storage requirements. This approach contrasts

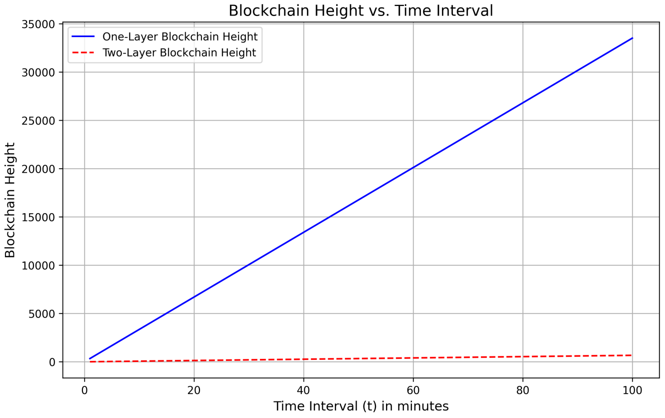


Fig. 17: Blockchain height growth comparison: one-layer vs. two-layer models over time.

with the model presented in [30], which relies on a single-layer blockchain. In BC-trust [30], the blockchain grows linearly with increasing data rates and the number of organizations, leading to rapidly escalating storage demands. In our two-layer model, the introduction of a threshold effectively limits blockchain growth, enhancing scalability and making it better suited for environments with a high number of organizations and elevated data generation rates.

B. Latency Analysis

In a one-layer blockchain, every transaction undergoes immediate global consensus, leading to higher confirmation delays. The total transaction latency, $L_{\text{One-Layer}}$, comprises:

$$L_{\text{One-Layer}} = L_{\text{prop}} + L_{\text{cons}} + L_{\text{block}} \tag{23}$$

where:

- L_{prop} is the transaction propagation delay.
- L_{cons} is the consensus delay.
- L_{block} is the block inclusion delay.

The growth rate of latency is $O(n)$, i.e., it increases linearly with the number of nodes and transactions.

In the two-layer model, transactions are initially recorded locally and are aggregated when the threshold w_n is reached. The total transaction latency is:

$$L_{\text{Two-Layer}} = L_{\text{local-prop}} + L_{\text{local-cons}} + \frac{L_{\text{One-Layer}}}{w_n} \tag{24}$$

where:

- $L_{\text{local-prop}}$ is the local transaction propagation delay (which is faster than global propagation).
- $L_{\text{local-cons}}$ is the local consensus delay (with fewer nodes and lower complexity).

This formulation shows that as w_n increases, the global blockchain latency component $\frac{L_{\text{One-Layer}}}{w_n}$ decreases significantly, leading to a lower overall latency. The latency growth rate in the two-layer model is $O(n/w)$, which is lower than $O(n)$.

1) Latency Reduction Factor: The latency reduction factor is defined as:

$$\text{Reduction Factor} = \frac{L_{\text{One-Layer}}}{L_{\text{Two-Layer}}} \tag{25}$$

Substituting the latency components:

$$\text{Reduction Factor} = \frac{L_{\text{prop}} + L_{\text{cons}} + L_{\text{block}}}{L_{\text{local-prop}} + L_{\text{local-cons}} + \frac{L_{\text{One-Layer}}}{w_n}} \tag{26}$$

For large w_n and assuming relatively low local latency, the reduction factor approximates to w_n. For instance:

- If $w_n = 10$, the latency is reduced by approximately 10x compared to the one-layer blockchain.
- If $w_n = 50$, the latency reduction factor is around 50x.

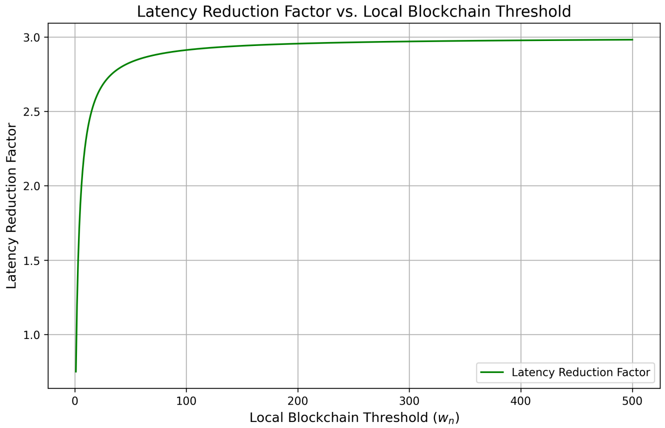


Fig. 18: Latency Reduction Factor vs. Local Blockchain Threshold (w_n) for a 100-node network, showing saturation at $w_n = 100$ with a 3x latency reduction

In Figure 18, the relationship between the Latency Reduction Factor and the Local Blockchain Threshold (w_n) is illustrated. The horizontal axis represents the value of w_n, while the vertical axis indicates the Latency Reduction Factor. The results demonstrate the impact of increasing w_n on reducing overall system latency. Notably, when w_n exceeds 100, the Latency Reduction Factor asymptotically approaches a constant value of approximately 3. This saturation behavior suggests that beyond a certain threshold, further increases in w_n do not contribute to additional latency reduction, indicating the presence of a performance plateau in the system.

Initially, as w_n increases, the Latency Reduction Factor rises rapidly, highlighting the significant effect of transaction aggregation in minimizing processing and consensus delays.

As w_n continues to increase, the rate of growth in the Latency Reduction Factor gradually declines. This observation suggests that beyond a certain threshold of w_n, the latency reduction reaches an optimal limit, and further increasing w_n has little additional impact on reducing latency.

While a larger w_n yields lower latency and smaller blockchain growth, it may delay the reputation update cycle. In dynamic IoT networks such as intelligent transportation systems, tuning w_n requires balancing between efficient aggregation and timely trust evaluation to ensure safety and responsiveness.

In a 1000-node smart city, dividing nodes into 100 partitions with 2–3 redundant devices per partition ensures sufficient honest majority and fault tolerance. In delay-sensitive IoT applications like autonomous vehicle fleets or industrial automation, high trust latency can result in operational delays or safety risks. The proposed two-layer model reduces latency by up to 3x for local trust updates, enabling more timely decisions in these dynamic environments.

VI. Security Evaluation

In this section, we evaluate the resilience of the proposed methods against the considered security attacks. A set of initial hypotheses are as follows

1) An adequate number of smart objects is assumed to be accessible to deliver the service effectively.
2) A maximum of t colluding malicious smart objects is assumed in each local group (e.g., partitions or gateway clusters), where $t < |p_{ij}|$ and $t < |N_{ij}|$. This constraint ensures that malicious nodes cannot dominate trust evaluations during aggregation or probing. However, as illustrated in figure 15, in all proposed methods, when the proportion of malicious nodes approaches 50%, the recall metric significantly drops. This indicates a critical decline in detection performance, where a growing number of honest nodes are incorrectly classified as malicious, undermining the trust framework's reliability under high-adversity conditions.
3) The size of all pruned partitions ($\widehat{p_{i_j}}$) and the number of pruned objects per gateway node ($\widehat{N_{i_j}}$) are stipulated to be greater than t.

The following attacks are taken into account:

1) Self-promoting attacks: This attack involves an intelligent object manipulating its reputation to increase it, which can be executed by either an individual attacker or a coordinated group of nodes. The trust level in the proposed methods is determined through probing and evidence mechanisms, stored locally in the blockchain, and used to calculate reputation. Therefore, a smart object cannot autonomously alter its trust value. Colluding nodes attempting to boost each other's reputation are prevented by assumptions 2 and 3. For instance, in a smart agriculture system, a compromised irrigation controller may artificially inflate its own trust value to continuously gain access to water resources, even while malfunctioning.
2) Bad-mouthing attack: In this scenario, the attacker aims to manipulate the reputation of other smart objects by reporting false data, either individually or as a group. Smart objects participating in the probing mechanism are unaware when responding to tests or queries. If an object behaves maliciously by reporting incorrect data, it will suffer reputational damage and eventually be expelled from the organization. In a smart home setting, a malicious thermostat may falsely report its neighbor's temperature sensor as faulty, leading to misclassification and the exclusion of honest nodes.
3) Ballot-stuffing attacks: Here, the attacker aims to artificially boost the reputation of malicious objects and provide positive recommendations for them. Due to blockchain technology, no intelligent object can manipulate trust or reputation in the proposed methods. Assumptions 2 and 3 rule out the possibility of collaborative node attacks. In smart manufacturing, an attacker could inject positive feedback loops through colluding robotic nodes to elevate a malicious sensor's reputation and mask its anomalies.
4) On-off attack: This type of attack involves a malicious smart object intermittently providing both good and bad services. As shown in Figure 10, in the proposed methods, the probing-based approach at $p = 0.5$ is capable of detecting up to 90% of malicious nodes when the maximum percentage of malicious nodes in the organization is 35%. Similarly, as illustrated in Figure 14b, the evidence-based method can accurately identify 90% of malicious nodes when the maximum percentage of malicious nodes in the organization reaches 50%. In vehicular networks, a compromised vehicle might behave normally for a period and then switch to malicious routing behavior intermittently, attempting to bypass trust filters through temporal evasion.
5) Whitewashing attack: This attack occurs when a malicious smart object with a tarnished reputation deliberately behaves in a manner that rapidly diminishes its reputation to the point of being expelled from the system. Subsequently, it attempts to rejoin the system and resume malicious activities with its original reputation intact. This form of attack is prevented because reputation values are permanently stored in the global blockchain. Additionally, smart objects are registered via unique physical addresses (MAC addresses) using smart contracts. Consequently, even after expulsion, the blockchain retains a record of the malicious node's behavior. In a smart campus network, a malicious node may leave the network and rejoin with a new identity to reset its reputation, effectively bypassing historical penalties. Our system mitigates this using node authentication and identity linkage.

It is important to distinguish between per-round recall and overall multi-cycle detection rates. While Figure 14b shows that recall declines at high malicious node ratios within a single evaluation round, the cumulative detection performance across time—particularly under on-off behavior—is higher, exceeding 90% in long-running simulations with pruning enabled.

In a smart grid deployment, where timing is critical for fault isolation and load balancing, trust misclassification due to 35% malicious nodes may result in detection delays of 10–15 minutes. This delay can propagate across subsystems, potentially leading to energy misallocation or regional power instability. Similarly, in a smart transportation network, if 30 out of 100 roadside sensors are compromised and falsely marked as trustworthy, traffic routing systems may generate incorrect detours, leading to congestion and average commute delays of up to 20 minutes during peak hours. These examples underscore the importance of rapid and accurate trust assessment in dynamic, safety-critical IoT environments.

VII. CONCLUSION

This paper introduces a novel hierarchical trust management mechanism leveraging blockchain technology. The proposed mechanism adopts a two-layer blockchain architecture to address the inherent limitations of smart objects while simultaneously mitigating transaction volume and blockchain length. This is achieved by employing a lightweight and temporary blockchain within the intra-organizational layer, thereby enhancing scalability. In both the probing-based and evidence-based methods, smart object trust is accurately computed and stored using local blockchains. The enterprise layer employs a local blockchain to periodically update smart object reputations to the global blockchain. Nodes failing to meet the initial reputation threshold are automatically removed from their organizations via a smart contract. The inter-organizational layer focuses on maintaining organizational trust and smart object reputations within the global blockchain. The proposed approach significantly enhances privacy and scalability, improves reliable data collection, and effectively identifies and eliminates malicious nodes within organizations. Evaluation results underscore the susceptibility of organizations with over 35% malicious nodes to collusion, potentially leading to the misclassification of honest nodes as malicious, thus warranting their classification as blacklist organizations.

REFERENCES

[1] G. Fortino, L. Fotia, F. Messina, D. Rosaci, and G. M. Sarné, “Trust and reputation in the internet of things: State-of-the-art and research challenges,” *IEEE Access*, vol. 8, pp. 60 117–60 125, 2020.

[2] Y. Liu, J. Wang, Z. Yan, Z. Wan, and R. Jäntti, “A survey on blockchain-based trust management for internet of things,” *IEEE Internet of Things Journal*, vol. 10, no. 7, pp. 5898–5922, 2023.

[3] D. Stefanescu, L. Montalvillo, P. Galán-García, J. Unzilla, and A. Urbieta, “A systematic literature review of lightweight blockchain for iot,” *IEEE Access*, 2022.

[4] Y. Liu, Z. Liu, Q. Zhang, J. Su, Z. Cai, and X. Li, “Blockchain and trusted reputation assessment-based incentive mechanism for healthcare services,” *Future Generation Computer Systems*, vol. 154, pp. 59–71, 2024.

[5] Z. Shen, Y. Wang, H. Wang, P. Liu, K. Liu, and J. Zhang, “Trust mechanism privacy protection scheme combining blockchain and multi-party evaluation,” *IEEE Transactions on Intelligent Vehicles*, 2024.

[6] A. Heshmati, M. Bayat, M. Doostari, and S. M. Pournaghi, “Blockchain based authentication and access verfication scheme in smart home,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 3, pp. 2525–2547, 2023.

[7] K. Gai, Y. She, L. Zhu, K.-K. R. Choo, and Z. Wan, “A blockchain-based access control scheme for zero trust cross-organizational data sharing,” *ACM Trans. Internet Technol.*, vol. 23, no. 3, 2023. [Online]. Available: https://doi.org/10.1145/3511899

[8] C. Esposito, M. Ficco, and B. B. Gupta, “Blockchain-based authentication and authorization for smart city applications,” *Information Processing & Management*, vol. 58, no. 2, p. 102468, 2021.

[9] R. Chai, W. Jiang, and X. Yang, “Hierarchical blockchain-based resource access control architecture and scheme for iot devices,” in *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*. IEEE, 2022, pp. 1–5.

[10] J. Ye, X. Kang, Y.-C. Liang, and S. Sun, “A trust-centric privacy-preserving blockchain for dynamic spectrum management in iot networks,” *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 13 263–13 278, 2022.

[11] X. Wu and J. Liang, “A blockchain-based trust management method for internet of things,” *Pervasive and Mobile Computing*, vol. 72, p. 101330, 2021.

[12] K. A. Awan, I. U. Din, M. Zareei, M. Talha, M. Guizani, and S. U. Jadoon, “Holitrust-a holistic cross-domain trust management mechanism for service-centric internet of things,” *Ieee Access*, vol. 7, pp. 52 191–52 201, 2019.

[13] K. A. Awan, I. U. Din, A. Almogren, M. Guizani, A. Altameem, and S. U. Jadoon, “Robusttrust–a pro-privacy robust distributed trust management mechanism for internet of things,” *IEEE Access*, vol. 7, pp. 62 095–62 106, 2019.

[14] J. Chen, Z. Tian, X. Cui, L. Yin, and X. Wang, “Trust architecture and reputation evaluation for internet of things,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, pp. 3099–3107, 2019.

[15] G. Zheng, B. Gong, and Y. Zhang, “Dynamic network security mechanism based on trust management in wireless sensor networks,” *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–10, 2021.

[16] M. Ebrahimi, M. S. Haghighi, A. Jolfaei, N. Shamaeian, and M. H. Tadayon, “A secure and decentralized trust management scheme for smart health systems,” *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 5, pp. 1961–1968, 2021.

[17] Y. Alghofaili and M. A. Rassam, “A trust management model for iot devices and services based on the multi-criteria decision-making approach and deep long short-term memory technique,” *Sensors*, vol. 22, no. 2, p. 634, 2022.

[18] G. D. Putra, C. Kang, S. S. Kanhere, and J. W.-K. Hong, “Detrm: Decentralised trust and reputation management for blockchain-based supply chains,” in *2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, 2022, pp. 1–5.

[19] G. Savithri and N. R. Sai, “Blockchain base solution for trust management challenges internet of things application,” in *2022 OITS International Conference on Information Technology (OCIT)*. IEEE, 2022, pp. 620–624.

[20] A. Pathak, I. Al-Anbagi, and H. J. Hamilton, “Tabi: Trust-based abac mechanism for edge-iot using blockchain technology,” *IEEE Access*, 2023.

[21] G. D. Putrat, S. Malik, V. Dedeoglu, S. S. Kanhere, and R. Jurdak, “Trust and reputation management for blockchain-enabled iot,” in *2023 15th International Conference on COMmunication Systems & NETworkS (COMSNETS)*. IEEE, 2023, pp. 529–536.

[22] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, “Lsb: A lightweight scalable blockchain for iot security and anonymity,” *Journal of Parallel and Distributed Computing*, vol. 134, pp. 180–197, 2019.

[23] A. R. Shahid, N. Pissinou, C. Staier, and R. Kwan, “Sensor-chain: A lightweight scalable blockchain framework for internet of things,” in *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2019, pp. 1154–1161.

[24] T. Dasu, Y. Kanza, and D. Srivastava, “Unchain your blockchain,” in *Proc. Symposium on Foundations and Applications of Blockchain*, vol. 1, 2018, pp. 16–23.

[25] M. Conti, G. Kumar, P. Nerurkar, R. Saha, and L. Vigneri, “A survey on security challenges and solutions in the iota,” *Journal of Network and Computer Applications*, vol. 203, p. 103383, 2022.

[26] H. Honar Pajooh, M. Rashid, F. Alam, and S. Demidenko, “Multi-layer blockchain-based security architecture for internet of things,” *Sensors*, vol. 21, no. 3, p. 772, 2021.

[27] E. Corradini, S. Nicolazzo, A. Nocera, D. Ursino, and L. Virgili, “A two-tier blockchain framework to increase protection and autonomy of smart objects in the iot,” *Computer Communications*, vol. 181, pp. 338–356, 2022.

[28] W. Ruan, J. Liu, Y. Chen, S. M. Islam, and M. Alam, “A double-layer blockchain based trust management model for secure internet of vehicles,” *Sensors*, vol. 23, no. 10, p. 4699, 2023.

[29] X. Wang, C. Zhang, and X. Chang, “Trusted management infrastructure with blockchain for edge device in smart city,” in *2022 IEEE 5th International Conference on Computer and Communication Engineering Technology (CCET)*. IEEE, 2022, pp. 207–213.

[30] D. E. Kouicem, Y. Imine, A. Bouabdallah, and H. Lakhlef, “Decentralized blockchain-based trust management protocol for the internet of things,” *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 2, pp. 1292–1306, 2020.

[31] V. Dedeoglu, R. Jurdak, G. D. Putra, A. Dorri, and S. S. Kanhere, “A trust architecture for blockchain in iot,” in *Proceedings of the 16th EAI international conference on mobile and ubiquitous systems: computing, networking and services*, 2019, pp. 190–199.

[32] M. Amiri-Zarandi, R. A. Dara, and E. Fraser, “Lbtm: A lightweight blockchain-based trust management system for social internet of things,” *The Journal of Supercomputing*, pp. 1–19, 2022.

[33] X. Wu, Y. Liu, J. Tian, and Y. Li, “Privacy-preserving trust management method based on blockchain for cross-domain industrial iot,” *Knowledge-Based Systems*, vol. 283, p. 111166, 2024.

[34] L. Fischer, F. Dötzer, and P. Magiera, "Vars: A vehicle ad-hoc network reputation system," in *Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks(WOWMOM)*, vol. 01, 06 2005, pp. 454–456.

[35] Z. Huang, S. Ruj, M. A. Cavenaghi, M. Stojmenovic, and A. Nayak, "A social network approach to trust management in vanets," *Peer-to-Peer Networking and Applications*, vol. 7, no. 3, pp. 229–242, 2014.

[36] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet of Things Journal*, pp. 1–1, 2018.

[37] M. Raya, P. Papadimitratos, V. D. Gligor, and J. P. Hubaux, "On data-centric trust establishment in ephemeral ad hoc networks," in *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, April 2008.

[38] M. A. Azad, S. Bag, S. Parkinson, and F. Hao, "Trustvote: Privacy-preserving node ranking in vehicular networks," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 5878–5891, 2019.

[39] H. Mahmoud, M. A. Azad, J. Arshad, and A. Aneiba, "A framework for decentralized, real-time reputation aggregation in iov," *IEEE Internet of Things Magazine*, vol. 6, no. 2, pp. 44–48, 2023.

[40] S. Awan, N. Javaid, S. Ullah, A. U. Khan, A. M. Qamar, and J.-G. Choi, "Blockchain based secure routing and trust management in wireless sensor networks," *Sensors*, vol. 22, no. 2, p. 411, 2022.