BTIA: A Lightweight Trust Score-Based Identity Authentication Scheme for the Internet of Things Using Blockchain

Xiaoyu Du, Chenlin Peng, Yanxiang Zhao* and Song Tao

***Abstract*—The rapid expansion of the Internet of Things (IoT) demands the secure and efficient authentication of a massive number of interconnected devices. To address the challenges of device forgery and unauthorized access, we propose BTIA-a blockchain and edge computing-based trusted distributed authentication scheme. First, BTIA adopts a hierarchical distributed architecture of "trust center-cluster head-cluster," with edge nodes acting as cluster heads to manage the devices within the cluster. Except for initialization and registration, device authentication and communication do not rely on the trust center. Second, we propose a dynamic authentication mechanism based on trust scores, using trust scores as the core authentication parameter. This mechanism combines historical data and real-time feedback of device behaviors to dynamically assess and make authentication decisions. Finally, we design an improved Practical Byzantine Fault Tolerance (PBFT) consensus algorithm in BTIA, optimizing the message-passing mechanism and node collaboration strategies to reduce communication overhead and computational complexity, making it more suitable for resource-constrained IoT environments. Security analysis shows that BTIA can resist common attacks. Performance evaluation demonstrates that BTIA exhibits significant advantages in terms of security, scalability, and resource efficiency.**

***Index Terms*—Internet of Things, blockchain, edge computing, trust management, identity authentication**

I. Introduction

THE Internet of Things (IoT) is revolutionizing communication technologies and enabling a broad spectrum of services across domains such as smart healthcare, smart cities, intelligent transportation, and industrial automation [1]. According to Statista, the number of active IoT devices worldwide increased from 11.7 billion in 2020 to 16.7 billion in 2023, and is expected to exceed 25 billion by 2025. If deployed in untrusted environments, these devices are vulnerable to various security threats, such as data interception, tampering, and unauthorized access to user privacy through message eavesdropping or correlation attacks. IoT environments encompass a diverse array of devices that differ in size, form factor, storage capacity, processing power, functionality, and operational constraints. In such a heterogeneous environment, information security and privacy protection have become two major challenges for IoT systems [2]. Identity authentication serves as the first line of defense in information security. It plays a critical role in ensuring secure data exchange and protecting systems from unauthorized access [3]. Many new protocols and network technologies have been applied to enable secure authentication in IoT, such as Transport Layer Security (TLS) and Secure Sockets Layer (SSL). However, these mechanisms often lack flexibility and scalability. In centralized IoT authentication mechanisms, a unified trusted authority such as an authentication center is typically responsible for authentication and access control. Clearly, this approach suffers from single points of failure and performance bottlenecks [4], making it ill-suited for dynamic and distributed IoT environments.

An increasing number of studies have shown that distributed access mechanisms can overcome the limitations of traditional centralized schemes, offering significant advantages in terms of system robustness, resistance to single-point attacks, and dynamic device management [5]–[7]. Among the available technologies, blockchain is considered an ideal foundation for building distributed access control due to its characteristics of decentralization, immutability, and transparency [8]. Although blockchain is widely regarded as promising, most existing blockchain-based solutions require high computational and storage resources. The redundant computation across network nodes and the irreversible growth of on-chain data have become major resource bottlenecks. In contrast, IoT devices typically operate under severe resource constraints. For example, passive RFID tags have no battery and can only harvest energy from nearby RFID readers or the surrounding environment [9]. With the ongoing miniaturization of hardware, IoT devices are becoming increasingly compact and autonomous, enabling them to perform appropriate actions without explicit human instructions [10]. These factors highlight the necessity of enhancing blockchain performance at the edge. Integrating edge computing into blockchain networks equips edge nodes with additional distributed computing and storage resources, enabling offloading of blockchain-related tasks such as storage and consensus from power-limited IoT devices [11].

To ensure secure interactions among devices, it is essential to establish trustworthy communication mechanisms within the IoT environment. Compared to other network systems, IoT introduces new trust management challenges due to

Manuscript received April 17, 2025; revised July 25, 2025.

This work was supported in part by the Special Project for Key R&D and the Promotion of Science, Technology Department of Henan Province (252102210175, 252102210115, 242102210202, 242102210196), Kaifeng Science and Technology Development Plan (2201010).

Xiaoyu Du is a professor at the School of Computer and Information Engineering, Henan University, Kaifeng 475001, China (e-mail: dxy@henu.edu.cn).

Chenlin Peng is a postgraduate student at the School of Software, Henan University, Kaifeng 475001, China (e-mail: pcl@henu.edu.cn).

Yanxiang Zhao is a lecturer at the School of Software, Henan University, Kaifeng 475001, China (e-mail: zhaoyx@vip.henu.edu.cn).

Song Tao is a graduate student at the School of Computer and Information Engineering, Henan University, Kaifeng 475001, China (e-mail: taosong00110@henu.edu.cn).

its unique characteristics. First, mutual trust among devices is a core issue in IoT trust management. Ensuring secure communication and defending against malicious attacks and unauthorized access are key research concerns. Second, the credibility of data has a direct impact on the accuracy and reliability of decision-making in IoT systems [12]. As IoT data is typically generated in real time and in dynamic contexts, its integrity, availability, and confidentiality must be safeguarded to prevent tampering or forgery [13]. While many existing studies have applied blockchain's distributed ledger to enhance the security and immutability of IoT data [14], there has been limited work on integrating edge computing with blockchain to build a highly efficient and trustworthy environment.

Therefore, leveraging the advantages of blockchain's distributed ledger, this paper proposes a lightweight IoT authentication scheme based on trust scores. The scheme introduces cluster head nodes to manage and register the identities of devices within each cluster, while authentication data and communication records are securely stored on the blockchain. By eliminating reliance on third-party authorities during the authentication process, the proposed approach enhances security. Furthermore, the involvement of cluster head nodes accelerates the authentication procedure and reduces associated costs. The main contributions of this paper are as follows:

1) A distributed trusted authentication system based on blockchain and edge computing is proposed. Considering the constrained computational and storage capabilities of IoT devices, a blockchain-enabled trust management model is developed to support the dynamic evaluation and maintenance of device reputation scores in a decentralized manner.

2) A distributed identity authentication scheme based on elliptic curve cryptography (ECC) is designed. This scheme uses efficient encryption algorithms to ensure identity confidentiality and communication security between cluster head nodes and IoT devices.

3) An optimized Practical Byzantine Fault Tolerance (PBFT) consensus algorithm, referred to as BTCA, is integrated into the blockchain network. Together with the proposed trust management model, it ensures reliable authentication while significantly reducing communication overhead.

The rest of this paper is organized as follows: Section 2 reviews related work. Section 3 introduces the system components of the proposed scheme. Section 4 presents the details of the BTIA scheme, including the authentication process, trust score quantification, and optimized consensus algorithm. Section 5 provides the security analysis. Section 6 evaluates the performance of the proposed scheme through simulation experiments. Finally, Section 7 concludes the paper.

II. Related Work

With the rapid proliferation of IoT devices, authentication as a fundamental mechanism for ensuring network security is encountering increasing challenges. Traditional authentication schemes mainly rely on technologies such as passwords, certificates, or biometrics, and often depend on trusted centralized entities. Sharma et al. [15] designed a scheme that utilizes biometric data for registration and authentication. Their approach combines noisy biometric data with cryptographic functions, ensuring that even if the database is compromised, the original data cannot be reconstructed. While this scheme provides strong security and privacy protection, its high computational overhead and system complexity make it unsuitable for the highly dynamic and heterogeneous nature of IoT devices. In [16], a lightweight authentication protocol based on an improved secure hash algorithm was proposed. This protocol can resist various attacks while reducing computational costs and communication overhead without compromising security. However, its storage requirements remain high, making it less suitable for resource-constrained IoT devices. Moreover, due to the high dynamics and heterogeneity of IoT environments, centralized authentication architectures often become single points of failure and prime targets for attacks.

In recent years, blockchain-based authentication schemes have emerged as a research hotspot. Cui et al. [17] proposed a hybrid blockchain identity authentication scheme for multi-wireless sensor networks. In their design, IoT nodes are classified into base stations, cluster head nodes, and regular nodes based on their capabilities. They also introduced a hybrid blockchain model that combines local chains and a public chain to better adapt to the heterogeneity and complexity of IoT environments. By recording identity information on the blockchain, the scheme achieves decentralization and distributed storage, enhancing both security and reliability. However, the blockchain consensus mechanism used in this approach consumes substantial computational resources and energy. Gao et al. [18] presented a blockchain-based privacy-preserving identity authentication scheme in which users autonomously generate their identity information and complete registration via smart contracts on the blockchain. This approach protects users' real identity information and avoids the storage overhead associated with maintaining large numbers of certificates or keys. Blockchain-based authentication schemes are inherently suitable for distributed scenarios and can effectively prevent the single point of failure problems common in traditional centralized schemes. Nevertheless, most of these approaches still lack robust mechanisms for device trust evaluation and dynamic management, which are essential in large-scale and complex IoT environments.

Trust management has proven to be an effective technique for providing secure services and has shown strong potential in IoT device behavior analysis and anomaly detection. Dehalwar et al. [19] developed a self-sovereign identity-based trust management system using blockchain technology, where blocks are categorized based on the historical behavior and trust index of devices. Blocks are divided into three categories: whitelist, blacklist, and graylist. Only miners from the whitelist—those with the highest level of trust—are allowed to participate in consensus. Once more than 60% of the whitelist blocks reach consensus, the corresponding block is added to the core blockchain. This approach significantly reduces the likelihood of identity-based security vulnerabilities in smart grid environments.

Meanwhile, the rise of edge computing has brought new opportunities for IoT identity authentication and trust management. The integration of blockchain and

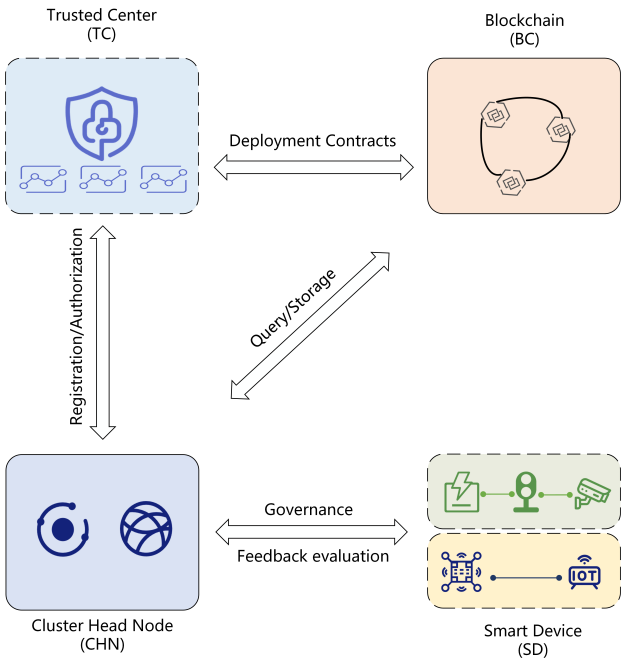

Fig. 1. System Model

edge computing has already demonstrated considerable potential in fields such as transportation, smart grids, and supply chains [20] [21]. Gai et al. [22] proposed a Permissioned Blockchain Edge Model for Smart Grids (PBEM-SGN), in which more than 80% of routine requests are handled by local edge nodes, enabling deep collaboration between blockchain and edge computing. PBEM-SGN shows notable advantages in modern smart grid scenarios involving distributed energy resources. However, due to its authentication mechanism being tightly coupled with the grid's topological features, the model is difficult to extend to other domains such as smart cities or industrial IoT, where cross-domain trust management is required. Kang et al. [23] presented a blockchain-based data sharing protocol for vehicular networks using RSU(road-side unit) clusters as edge nodes. Their scheme utilizes consortium blockchain to ensure data integrity and smart contracts to automate sharing operations. Although effective in vehicle-edge collaborations, its heavy reliance on RSU deployment presents limitations in areas with sparse RSU coverage, where direct peer-to-peer data sharing among vehicles is not supported.

In light of these limitations, designing a lightweight, trustworthy, and secure identity authentication framework that integrates both edge computing and blockchain is crucial for enhancing the security, scalability, and resource efficiency of future IoT networks.

III. Architecture and System Components

In the BTIA scheme, the trust-score-based architecture mainly consists of blockchain, IoT devices, edge nodes, and a trusted center, as shown in Figure 1. The functions of each component are as follows.

1) Trusted Center(TC): The Trusted Center serves as a fully trusted entity. It first performs cluster division and cluster head selection for the devices, then registers and manages the cluster heads, and writes the registration records to the blockchain via smart contracts.

2) Blockchain Network(BC): In the BTIA scheme, a private blockchain is deployed with device identity registration contracts and reputation score calculation contracts. These contracts are used to store the registration records and reputation scores of various nodes.

3) Cluster Head Node(CHN): Devices of the same type within the communication range form a cluster, with the most powerful device in terms of computing and storage capacity serving as the cluster head. The cluster head is responsible for registering devices within the cluster and assisting in authentication between devices.

4) Smart Devices(SD): Smart devices can serve as either communication requesters or receivers. After registering through the cluster head node, smart devices receive a token that stores their reputation score. Devices possessing the token can authenticate each other.

This paper proposes a dynamic, distributed trust model aimed at evaluating the trustworthiness of nodes by calculating the reputation scores of each device in the IoT network, thereby enabling secure communication. The model abstracts each IoT device as a node in the undirected graph $G = \{V, E\}$, with the set of nodes represented as $V = \{V_1, V_2, ..., V_i..., V_n\}$, where n is the total number of IoT devices in the network. Each node has its own set of resources, and accessing different resources may require different reputation scores. In other words, a node may only be allowed to access specific resources of other nodes, rather than all resources. The goal of the model is that, upon receiving a communication request, the message recipient can use a ticket to obtain the trust scores of both the requesting node and the cluster. This enables the recipient to determine whether the requesting node is malicious and decide whether to engage in communication with it.

IV. Trust Score-Based Authentication Scheme

In the BTIA scheme, the trusted center first initializes the system, performs the clustering of IoT devices, and selects the cluster heads. The cluster head nodes must be registered with the trusted center to obtain management authority for the cluster. IoT devices, on the other hand, need to complete their registration at the cluster head of their respective cluster. The cluster head nodes issue tickets to devices that successfully register, which continuously update the trust scores of the devices. These tickets serve as a crucial basis for authentication communication between devices and granting access rights. This section will introduce the BTIA scheme in four stages: initialization, cluster head and device registration, device authentication, and trust score-based maintenance. Table I provides some symbols used in BTIA along with their descriptions.

A. System Initialization

The initialization phase is primarily carried out by the trusted center, which involves clustering IoT devices, selecting cluster heads, and generating public parameters.

1) Clustering: The trusted center groups IoT devices with the same attributes within the communication range into a cluster, forming multiple clusters.

TABLE I
NOTATIONS

Notations	Descriptions
D_i	IoT device
$cluID_k$	Cluster
CID_k	The cluster head node
TC	Trust Center
ID_i	The real identity of D_i
PID_i	The pseudonym of D_i
SK_i	The secret key of D_i
PK_i	The public key of D_i
SK_{clu}	The secret key of CID_k
PK_{clu}	The public key of CID_k
SK_{pub}	The secret key of TC
PK_{pub}	The public key of TC
h_i(i=1,2,3)	Hash functions
T_i(i=1,...,4)	Timestamp
$Ticket_{D_i}$	The ticket of D_i

2) Selection of Cluster Head Nodes: Edge nodes must submit an application to the trusted center's evaluation system, providing qualification information including the node's identity, computational benchmarking report, storage capacity verification, and historical behavior log. The node's identity is established by collecting immutable hardware feature values, which are combined to create a hardware fingerprint. The computational benchmarking report and storage capacity verification provide details of the node's performance in these areas. The historical behavior log reviews the node's reputation, computing power, and storage performance. The trusted center's evaluation system then comprehensively assesses the node's reputation, computing power, and storage performance, selecting the node with the highest value within the cluster as the cluster head node.

3) Parameter Initialization: The trusted center randomly generates a large prime number n, and defines a non-singular elliptic curve $E(n)$. The points on the curve $E(n)$ form an additive group G of order q, with P as the generator. The trusted center then randomly generates a number $SK_{\text{pub}} \in \mathbb{Z}_q^*$, which is used as its private key. The corresponding public key is generated as $PK_{\text{pub}} = PK_{\text{pub}} \cdot P$. Additionally, three hash functions are randomly selected: $h_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$, $h_2 : \{0,1\}^* \rightarrow \{0,1\}^l$, and $h_3 : \{0,1\}^* \rightarrow \{0,1\}^k$, where l is the length of the pseudonym and k is the length of the encryption key. Finally, the trusted center stores its private key and publishes the relevant public parameters: $\{n, q, P, PK_{\text{pub}}, h_1, h_2, h_3\}$.

B. Identity Registration and Authorization

1) Cluster Head Node Registration: The cluster head node needs to register with the trusted center to obtain data access and data publishing authorization. The specific steps are as follows.

1) The cluster head node selects a random number $SK_{\text{clu}} \in \mathbb{Z}_q^*$ as its private key and computes its public key $PK_{\text{clu}} = SK_{\text{clu}} \cdot P$. It also generates a timestamp T_1, and sends the public key and timestamp $\langle PK_{\text{clu}}, T_1 \rangle$ to the trusted center.

2) Upon receiving the information, the trusted center first verifies the validity of the timestamp. If invalid, the registration is terminated. If valid, the trusted center queries the blockchain to check if the node is already registered. If it is already registered, the registration is terminated; Otherwise, the trusted center generates the cluster head node's identity number CID and cluster number cluID = keccak256(CID, $Eads_{TC}$), and writes the information $\langle$CID, cluID$\rangle$ to the blockchain, where $Eads_{TC}$ is the Ether address of the trusted center.

3) The trusted center generates a new timestamp T_2 and sends $\langle$CID, cluID, $T_2\rangle$ to the cluster head node to complete the registration process.

Algorithm 1 Device Registration Algorithm

Input: Cluster ID $cluID$, Device pseudonym PID_i, Cluster head node CID, Cluster's Ethereum address $Eads_{clu}$
Output: bool

```
if msg.sender ≠ CID then
    return false;          ▷ Device registration is invalid
else
    Stored h3(cluID, PID_i) in the blockchain;
    return true;
end if
```

2) IoT Device Registration: The IoT device applies for registration with the cluster head of its respective cluster. The cluster head generates a pseudonym for the smart device and, through the device registration contract on the blockchain, generates a ticket that records the trust score for the smart device, as shown in Figure 2. The specific steps are as follows.

1) The device D_i selects a random integer $m \in \mathbb{Z}_q^*$, then calculates its private key $SK_i = h_1(ID_i \,||\, PK_{\text{pub}} \,||\, m)$ and public key $PK_i = SK_i \cdot P$. The device D_i stores its private key locally, then selects a random number $c_i \in \mathbb{Z}_q^*$ as a private dynamic parameter, and calculates the public dynamic parameter $C_i = c_i \cdot P$ and the public parameter (PK_i, C_i). The device then sends its real identity information ID_i to the cluster head node.

2) The cluster head node computes the pseudonym $PID_i = ID_i \oplus h_2(SK_i \cdot PK_{\text{pub}})$ for the device. The device registration contract deployed on the blockchain, as shown in Algorithm 1, will store the device's cluster number $cluID$ and pseudonym PID_i. It then uses elliptic curve encryption to generate an interaction message that can be shared among nodes in the blockchain network. Formula 1 defines the encryption of the interaction message using the private key of the trusted center SK_{pub}, while Formula 2 describes its decryption using the corresponding public key PK_{pub} .

$$Tra1 = Enc(cluID, PID_i) \tag{1}$$

$$Dec(Tra1) = cluID, PID_i \tag{2}$$

3) As shown in Algorithm 2 , the ticket generation contract generates ticket information for successfully registered devices based on $cluID$, PID_i, $Eads_{clu}$, and $Eads_{TC}$. The ticket is given by $Ticket_{D_i} = (cluID, PID_i, Eads_{TC}, T_3, C_i)$, where the timestamp T_3 is used to ensure the freshness of the ticket. C_i represents the device's reputation score, and the successfully registered device receives an initial C_i equal to the reputation score of its cluster. As defined in Formula 3, the Trust Management Center computes the hash of the issued ticket and encrypts

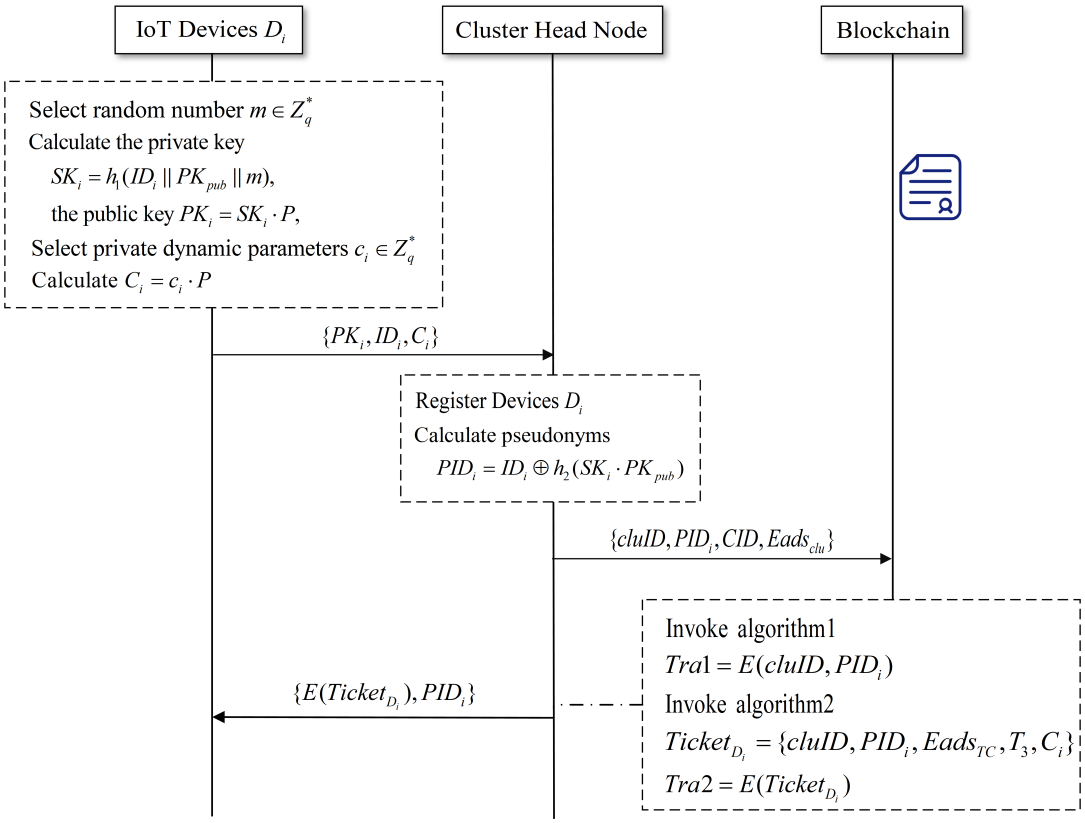

Fig. 2. IoT Device Registration Phase

Algorithm 2 Ticket Generation Algorithm

Input: Cluster ID $cluID$, Device pseudonym PID_i, Cluster's Ethereum address $Eads_{clu}$, TC Ethereum address $Eads_{TC}$

Output: bool

1: **if** msg.sender $\neq$ CID **then**
2: **return false;** ▷ Ticket generation is invalid
3: **else**
4: Generate timestamp T_3; ▷ Ensure freshness of the ticket
5: Calculate device trust score $C_i = E_c$ ▷ Initial trust score is the same as the cluster score
6: Create ticket:
$Ticket_{D_i} = (cluID, PID_i, Eads_{TC}, T_3, C_i)$
7: Store $h_3(Ticket_{D_i})$ in the blockchain;
8: **return true;**
9: **end if**

it using its private key SK_{pub}. This encrypted hash is then written to the blockchain and disseminated to all devices in the associated cluster to ensure integrity and authenticity.

$$Tra2 = Enc(Ticket_{D_i}) \tag{3}$$

C. Identity Authentication Details

This section describes the identity authentication process between registered devices. The sending device D_i sends an authentication request to the cluster head node of the receiving device D_j. After the cluster head node confirms the legitimacy of the device's identity, it checks whether D_i and D_j belong to the same cluster. For devices within the same cluster, the authentication must meet the trust threshold between the devices. If the devices belong to different clusters, the authentication process must also satisfy the trust thresholds defined by each respective cluster, as illustrated in Figure 3. Once the device's identity is confirmed as legitimate and trustworthy, the cluster head will assist D_i and D_j in negotiating a session key for secure communication.

1) The device D_i initiates an authentication request. D_i calculates $f_i = c_i + SK_i \cdot h_1(PID_i \parallel T_4)$, and then sends $\{PID_i \parallel PID_j \parallel T_4 \parallel f_i \parallel E_{SK_{clu}}(Ticket_{D_i})\}$ to the cluster head node of D_j. Here, $E_{SK_{clu}}(Ticket_{D_i})$ denotes the ticket originally issued by the cluster head of D_i, and encrypted using the cluster's private key SK_{clu}.

2) When the cluster head node receives the message from D_i, it first checks the freshness of the timestamp T_4. Then, according to formula 4, it verifies whether $f_i \cdot P$ is equal to $C_i + h_1(PID_i \parallel T_4)$. If they are not equal, it indicates an error in the information. If they are equal, it proves the authenticity of the identity.

$$\begin{aligned} f \cdot P &= c_i \cdot P + SK_i \cdot h_1(PID_i \parallel T_4) \cdot P \\ &= C_i + PK_i \cdot h_1(PID_i \parallel T_4) \end{aligned} \tag{4}$$

3) Then, the cluster head node decrypts $E_{SK_{clu}}(Ticket_{D_i})$ and verifies the authenticity of the ticket. The cluster head node checks the $cluID$ in the ticket to determine whether D_i belongs to the same cluster. If $cluID_i \neq cluID_j$, it indicates that devices D_i and D_j are in different clusters. In this case, the cluster head will compare the trust scores of the two clusters. If the trust scores satisfy $|E_j - E_i| \leq \sigma E_{\max}, \sigma \in [0, 0.5]$, the cluster head will read the trust score

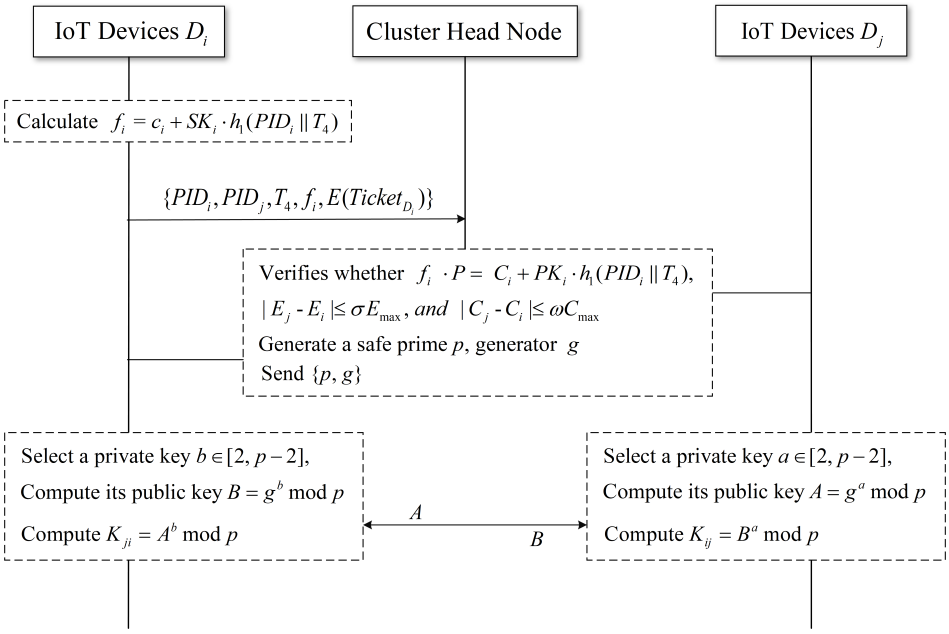

Fig. 3. IoT Device Authentication Phase

C_i from Ticket_{D_i} and check if it satisfies $|C_j - C_i| \leq \omega C_{\max}, \omega \in [0, 0.5]$. Otherwise, if $cluID_i = cluID_j$, meaning the devices are in the same cluster, only the devices' trust scores need to meet the required threshold. Once the trust scores meet the threshold, the cluster head will generate a safe prime p and generator g, and send them to devices D_i and D_j to assist them in negotiating a session key. If the trust score threshold is not met, authentication is denied.

4) Device D_i selects a private key $a \in [2, p-2]$ based on the Diffie-Hellman algorithm and computes its public key $A = g^a \mod p$. Similarly, device D_j selects a private key $b \in [2, p-2]$ and computes its public key $B = g^b \mod p$. After exchanging public keys, D_i calculates the shared key $K_{ij} = B^a \mod p = (g^b)^a \mod p$, while D_j calculates $K_{ji} = A^b \mod p = (g^a)^b \mod p$. Due to the associative property of exponentiation, $K_{ij} = K_{ji}$, and both devices obtain the same shared secret key.

D. Trust Score-Based Ticket Update

Devices will choose devices with higher trust scores for communication within the same range. After the authentication and communication processes are completed, the trust scores of both the cluster head and the devices will be updated.

1) Trust Score Calculation: The trust score of device node D_i for device node D_j is denoted as T_{ij}. The cluster head updates the trust score stored on the blockchain based on the feedback information f from the device node. f_{ij}^t represents the feedback information from the t-th interaction, with a value in the range of $[-1, 1]$. The closer the value is to 1, the better the interaction experience. The closer the value is to -1, the worse the interaction experience. If it is the first interaction between two device nodes, the reputation score of the cluster to which the nodes belong is used as the trust score. The trust score after n interactions is calculated as shown in formula 5.

$$T_{ij}(n) = \alpha \sum_{t=1}^{n} f_{ij}^t \cdot F(t) \tag{5}$$

F(t) is a linear function positively correlated with the number of interactions, used to control the rate of trust score growth. This setup ensures that the more positive interactions occur, the faster the trust-building process, similar to how trust is established in human society. If all interactions are positive, as $n \to \infty$, T_{ij} will reach a maximum threshold, as demonstrated in formula 6.

$$T_{ij}(\infty) = \alpha \sum_{t=1}^{n} f_{ij}^t \cdot F(t) = \delta_{\text{pos}} \tag{6}$$

Conversely, there exists $T_{ij}(\infty) = \delta_{\text{neg}}$, so the trust score T_{ij} is bounded by two extreme cases:

$$\delta_{\text{neg}} \leq T_{ij} \leq \delta_{\text{pos}} \tag{7}$$

2) Calculation of Device Node Reputation: The device's reputation score is determined by the sum of the trust scores from all other device nodes that interact with it, reflecting the device's reliability within the entire network. The calculation formula for the reputation score C_j of device D_i is shown in equation 8:

$$C_i = \frac{\sum_{j=1}^{m} T_{ij} + \beta M_i + \gamma I_i}{1 + \ln(m-1)} \tag{8}$$

Where m represents the set of nodes that have interacted with D_i, and $M_i, I_i \in [-1, 0]$ are the normalized values of the storage capacity and computational ability of D_i, respectively. Considering that higher computational power and storage capacity may increase the risk of malicious behavior, these factors influence the device's reputation score. To mitigate the potential attack motivation of high-capacity nodes, resource penalty factors β and α are introduced, with values randomly selected from the range $[-1, 0]$. In the calculation of the node's reputation score, the

weight of the trust score T_{ij} is much higher than that of storage capacity M_i and computational ability I_i.The node's reputation score is also constrained within a range, as shown in equation 9:

$$\frac{m\delta_{neg}+\beta M_i+\gamma I_i}{1+\ln(m-1)} \le C_i \le \frac{m\delta_{pos}+\beta M_i+\gamma I_i}{1+\ln(m-1)} \quad (9)$$

3) Calculation of Cluster Head Node Trust Score: The selection of cluster head nodes is determined by the trust scores, computing power, and storage performance of edge nodes. The calculation of the cluster head's trust score is as follows: The score of a newly initialized cluster head node is set to $\text{Score}_k^{\text{init}} = \zeta \cdot \text{Score}_{\text{max}}$, where $\zeta \in [0.6, 0.8]$, and $\text{Score}_{\text{max}}$ is the maximum trust score, and ζ is a random number between 0.6 and 0.8, indicating that newly registered devices are assumed to have a certain level of trust. The cluster head node will improve its reputation score by continuously performing beneficial actions on the blockchain network. Let Score_k represent the trust score of the cluster head node, and Cluster represent the cluster managed by the cluster head CID_k. The devices within the cluster will rate the performance and service satisfaction of the cluster head node, denoted by ass_{i-j}. A higher score indicates greater satisfaction with the services provided by the cluster head node.

Additionally, to prevent malicious rating behaviors, the trust score C_i of the device itself must be considered when rating. Devices with higher trust scores will have greater weight in the scoring process. δ_{pos} represents the maximum trust score, and the formula for evaluating the trust score of the cluster head node is shown in equation 10:

$$Score_k = \sum_{i \in Cluster} \frac{C_i}{n\delta_{\text{pos}}} \cdot ass_{i-k} \quad (10)$$

4) Calculation of Cluster Reputation: Communication within a cluster only requires consideration of the nodes' reputation scores. However, if a node wishes to communicate across clusters, the reputation score of the cluster must also be taken into account. For devices within the same cluster, if one device exhibits malicious behavior, the likelihood of other devices in the cluster also engaging in malicious behavior is high. Therefore, communication between device nodes across clusters should also consider the reputation score of the cluster to which the node belongs. The formula for calculating the reputation score of cluster c is shown in equation 11.

$$E_c = \frac{\sigma \sum_{i=1}^{w} C_i}{w} \quad (11)$$

Where w represents the number of devices in cluster c. The cluster head node calculates the reputation score and updates it to the blockchain. Device nodes can query the information on the blockchain through their respective cluster head nodes when needed.

E. Consensus Mechanism

Identity authentication and related transactions between IoT devices will be packaged by consensus nodes and stored on the blockchain. To ensure that all nodes are responsible for maintaining the secure and stable operation of the blockchain to protect their own interests and prevent malicious attacks, this section designs a trust score-based consensus algorithm, BTCA, for IoT environments with low storage and computing capabilities. The specific process of the algorithm is shown in Figure 4. The process of the consensus algorithm can be detailed as follows:

1) Election of the master node. All device nodes are divided into two categories: a consensus node set composed of cluster head nodes and a candidate node set consisting of other device nodes. Cluster head nodes calculate their trust scores based on the trust management mechanism and store them on the blockchain. The trust management center ranks the trust scores of all cluster head nodes by accessing the blockchain and selects the highest-ranking node as the master node. In the case of a tie in rankings, a node is randomly chosen as the master node.

2) Master node block generation phase. The master node packages the data from the buffer pool to form a block, and then sends $\{Block_{id},\ D(M, \text{sk}_m)\}$ to other consensus nodes for verification. $Block_{id}$ is the block number being verified, and $D(M, \text{sk}_m)$ is the block information signed with the master node's private key sk_m.

3) Verification of the generation block phase. Other nodes will provide the verification results to the master node in the form of $\{\{Block_{id}, D(\text{report}, \text{sk}_v)\}$. $D(\text{report}, \text{sk}_v)$ is the verification information signed with the consensus node's private key sk_v, and $report$ takes the value of 1 if the verification is successful, and 0 if the verification fails. Regardless of whether the verification is successful or not, consensus nodes temporarily add the hash of the received block to their buffer pool, but do not add it to their own copies. When the master node receives more verification success messages than failure messages, it proceeds to the next phase.

4) Master node and consensus node block submission phase. The master node sends the block hash $\{h(Block_{id}), \text{sk}_m\}$, signed with its private key, to the consensus nodes. The consensus nodes can decrypt $h(Block_{id})$ and verify whether it matches the block cached in their buffer pool. If they match, the consensus nodes send a response message to the master node. When the master node receives responses from more than two-thirds of the consensus nodes, the transaction is validated and stored on the blockchain. Otherwise, the consensus nodes will remove the block from their buffer pool.

5) The master node broadcasts the new block generation to the entire network. Nodes participating in the consensus algorithm update their reputation scores. For the master node that successfully uploads the block, the reputation score is updated as $\text{Score}_k = \text{Score}_k + \sqrt{t}$, while if block generation fails, the reputation value of the master node is decreased by $\text{Score}_k = \frac{\text{Score}_k}{2}$. Here, Score_k is the trust score of node k, and t is the total number of successful block generations by that node. For nodes that successfully send block verification information, their reputation score is updated as $\text{Score}_k = \text{Score}_k + \frac{\sqrt{q}}{10}$, where q is the total number of successful verification information transmissions. The reputation score of the nodes that did not participate in the block verification message sending is updated according to $\text{Score}_k = \frac{\text{Score}_k}{3}$. Finally, based on the new scores, the

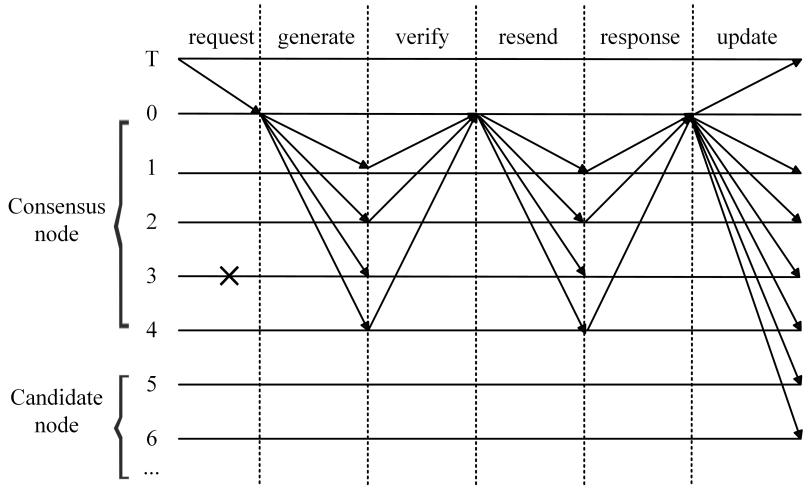

Fig. 4. Trust Score-Based Blockchain Consensus Algorithm

consensus node set and candidate node set are redefined.

V. Security Analysis

The scheme proposed in this paper satisfies the following security:

1) Confidentiality: Unauthorized devices or attackers cannot intercept session keys or sensitive information. The shared key K_{ij} between devices D_i and D_j is computed using Diffie-Hellman, and its security is based on the Discrete Logarithm Problem (DLP). Currently, no efficient algorithm is known to solve DLP within a reasonable time frame. Even if an attacker intercepts the public keys A and B, they cannot compute the session key K_{ij}.

2) Resistance to Replay Attacks: The scheme prevents attackers from intercepting and replaying authentication requests to bypass identity verification. The cluster head node checks the freshness of the timestamp in the received authentication request. If the timestamp is expired, the request is rejected, preventing attackers from using outdated authentication requests to impersonate legitimate devices. Furthermore, the trust score in each device's unique ticket $Ticket_{(D_i)}$ is updated with each authentication, and expired tickets are considered invalid.

3) Resistance to Man-in-the-Middle (MITM) Attacks: The scheme prevents attackers from intercepting and altering device communication while masquerading as legitimate devices. Devices must undergo authentication by the cluster head node to ensure that both communication parties are legitimate devices, preventing attackers from directly accessing the network. After devices D_i and D_j exchange public keys, they each compute the shared key $K_{ij} = B^a \mod p$, $K_{ji} = A^b \mod p$. A MITM attacker cannot modify the public keys A and B without being detected; Otherwise, the key exchange will fail.

4) Trust management and access control: The scheme ensures that communication between devices occurs only between trusted devices, preventing malicious devices from accessing the network. During authentication, the cluster head verifies the trust scores. Authentication within the same cluster requires the device trust score to meet the threshold, while cross-cluster authentication requires both the cluster trust score and the device trust score to meet the threshold. If the device trust score or the cluster trust score does not meet the required threshold, authentication is denied, preventing malicious devices from attacking the network.

5) No online TC: Eliminates the dependence on centralized authority and achieves fully distributed management. The TC no longer needs to perform device registration and pseudonym generation, and the authentication between devices and cluster heads does not depend on the TC. The communicator is able to complete device registration, authentication, and key negotiation without involving the TC.

VI. Implementation and Performance Evaluation

Our experiments were conducted on a laptop equipped with an Intel Core i5-10210U @1.6GHz processor and 32GB of RAM. A private Ethereum blockchain was deployed using Geth version 1.10.8, with smart contracts written in Solidity 0.4.25 to handle the storage and update of trust scores. To evaluate system performance, we measured gas consumption and assessed the trust feedback mechanism and consensus overhead through simulation experiments.

1) Trust Management Evaluation: In the trust establishment process, this section evaluates $F(t)$ for the speed of trust building with different functions. When $F(t) = \sqrt[3]{t^2}$, the initial trust score is 0, the maximum trust score is 1000, and the minimum trust score is -1000.As shown in Figure 5, positive interactions gradually increase the trust score at an accelerating rate, ultimately reaching the maximum value. In contrast, negative interactions progressively damage the trust score, with the rate of decrease accelerating until it eventually reaches the minimum value. When the function is selected as $F(t) = \sqrt{t}$, $F(t) = t$, $F(t) = t^2$, $F(t) = \ln(t)$, as shown in Figure 6, it can be observed that as the number of interactions increases, the trust values of all devices increase.The difference lies in the rate at which trust increases. If the trust score increases too quickly, it will make trust establishment overly easy, which can impact the system's security. As seen in equation 11, a rapid increase in trust scores can cause the trust value of the entire cluster to rise suddenly, which does not reflect real-world scenarios. On the other hand, if trust is established too slowly, it will affect the communication efficiency of the entire system.

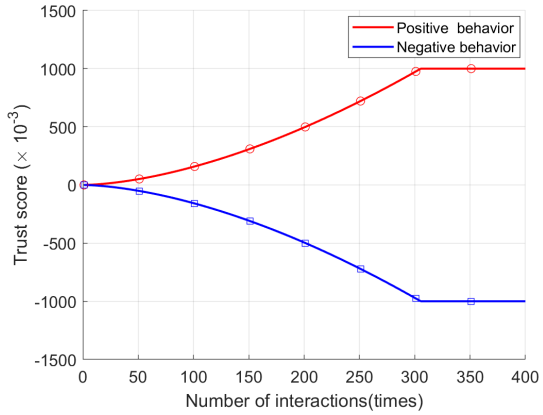

Fig. 5. Trust Establishment Trend

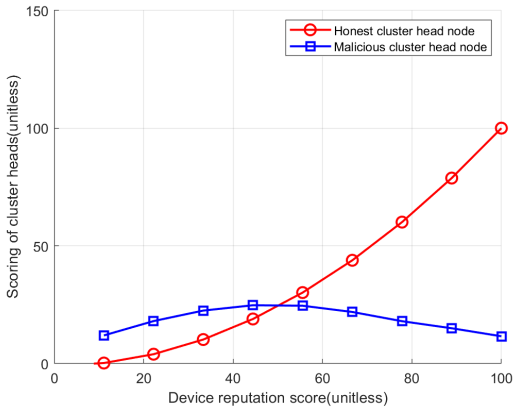

Fig. 8. Impact of Device Scoring on Cluster Head

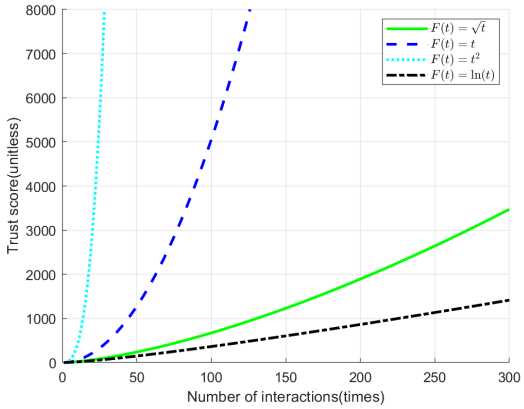

Fig. 6. Trust Establishment Comparison

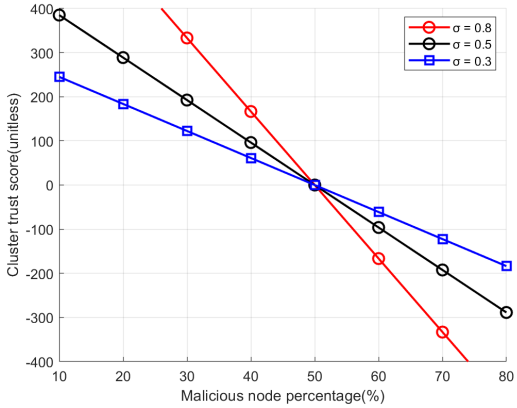

Fig. 7. Cluster Reputation Trend

Many nodes may exhibit friendly behavior but fail to reach the minimum communication threshold for a long period of time, which results in low communication efficiency and negatively impacts the operation of various IoT nodes within the network.

Figure 7 shows the impact of different percentages of malicious nodes on the overall cluster reputation score. The reputation score of malicious nodes is set to -1000, while the reputation score of honest nodes is set to 1000, with a total of 100 nodes. It can be observed that as the number of malicious nodes increases, the cluster's credibility gradually decreases. Additionally, based on equation 11, different values of δ are compared to analyze the rate of decline. As shown in the Figure 7, the larger the value of δ, the greater the impact on the overall cluster credibility, which aligns with the goal of cluster trust assessment. Figure 8 analyzes the impact of devices with different trust scores on the cluster head's rating. Devices with trust scores ranging from 10 to 100 are observed, with lower-scoring devices being more likely to be malicious. Malicious devices may attempt to alter the cluster head's score. The results show that devices with higher trust scores have a greater influence on an honest cluster head's score, ensuring a fairer evaluation. Conversely, for a malicious cluster head, devices with lower trust scores have minimal influence, making it hard to inflate the score. Devices with medium trust scores have the greatest impact, but regardless of the trust score level, the malicious cluster head tends to receive lower ratings.

2) Smart Contract Cost Evaluation: This section tests the Gas cost of the main functions of the contract. Gas is a unit of measurement used to calculate the computational work required for each transaction on the blockchain network. The more complex the transaction, the greater the computational intensity, and thus the higher the Gas cost. Each operation of the smart contract consumes a certain amount of Gas. For the purpose of comparative analysis, the data size uploaded to the smart contract is standardized to 128 bytes, and the Gas consumption for different smart contracts is summarized in Table II. The contracts in the BTIA scheme are deployed on a private blockchain, and the Gas cost is not assigned a monetary value but is used as a measure of consumption.

As shown in Table II, the Gas cost for device registration is the highest, approximately 4 times the Gas cost of updating the device's trust score. However, since the device registration operation is executed only once during the authentication and communication process, this is acceptable. Since the trust score of the cluster head node is influenced by the trust scores of devices within the cluster, the Gas cost for updating the cluster head's trust score is higher than the cost of updating the device's trust score.

3) Consensus Cost Evaluation: Figure 9 compares the communication overhead of the BTCA algorithm with the traditional PBFT and improved CPBFT algorithms. As the number of nodes increases, communication overhead grows for all algorithms. However, the BTCA algorithm shows clear

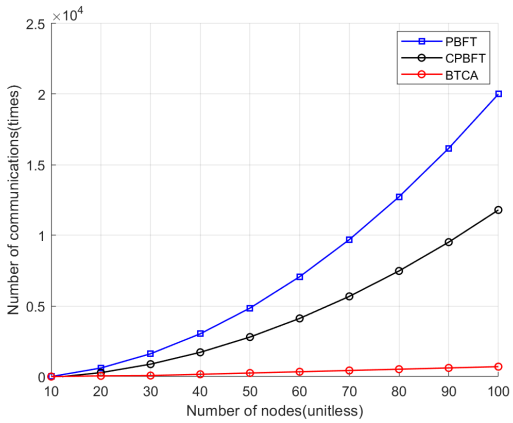

Fig. 9. Comparison of Communication Overhead

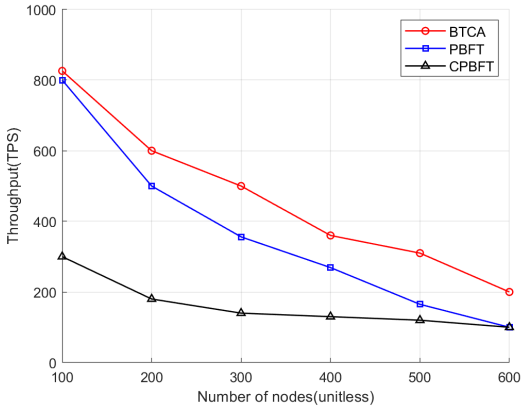

Fig. 10. Throughput Comparison

TABLE II
GAS COST FOR DIFFERENT OPERATIONS

Operation	Gas
Device Registration	107211
Ticket Generation	92961
Device's Trust Score Update	22526
Cluster Head's Trust Score Update	12340

advantages when the node count is large. For 50 nodes, the BTCA overhead is 245, and for 100 nodes, it is 495—both significantly lower than PBFT and CPBFT. This is because PBFT and CPBFT involve communication and verification among all nodes, leading to quadratic growth in overhead. In contrast, the BTCA algorithm reduces overhead by allowing only high-trust nodes to communicate, resulting in a constant increase in communication overhead as nodes grow.

Throughput refers to the number of successful operations completed by the system within a unit of time, typically measured in transactions per second (TPS). It is an important metric for evaluating system performance, directly reflecting the system's processing efficiency and performance level. In this section, consensus messages for 3000 requests were sent by message sender nodes, and transaction throughput was recorded for node counts ranging from 100 to 600, in order to compare the BTCA algorithm with the traditional PBFT and CPBFT algorithms. From the results shown in Figure 10, it can be seen that the BTCA algorithm outperforms

TABLE III
PERFORMANCE OF VARIOUS SYMBOLIC OPERATIONS

Symbol	Description	Alibaba Cloud (ms)
T_e	Modular exponentiation	0.339
T_m	Scalar multiplication	1.97
T_h	Hash function	0.009
T_b	Bilinear pairing	5.275
T_a	Point addition	0.012

TABLE IV
COMPARISON OF COMPUTATIONAL COSTS

Scheme	Computation Costs	Time (ms)
[24]	$T_b + 9T_m + T_e + 10T_h + 3T_a$	23.47
[25]	$7T_m + 16T_h + T_a$	13.946
[26]	$6T_m + 8T_h$	11.892
BTIA	$T_m + 4T_e + T_h + T_a$	5.338

the traditional PBFT and CPBFT algorithms in terms of throughput. Although the throughput of all three algorithms decreases as the number of nodes increases, when the number of nodes reaches 600, the BTCA algorithm still demonstrates a higher throughput, outperforming the PBFT and CPBFT algorithms by 98 and 110 TPS, respectively. This indicates that the BTCA scheme is capable of completing more transactions in a unit of time, exhibiting superior performance and processing efficiency.

4) Comparison of Schemes: In this section, we evaluate the efficiency of the proposed authentication scheme by comparing it with the schemes proposed by Jia et al. [24], Zhang et al. [25], and Wang et al. [26]. Since Jia et al. [24] have already simulated the computational and communication parameters on the Alibaba Cloud platform and assessed the performance of several cryptographic operations, we directly adopt their reported parameters for comparison. Based on the execution times of basic operations listed in Table III, we conduct a comprehensive analysis of several representative authentication schemes. In terms of computational cost, we primarily focus on the authentication phase, as it constitutes the core of the entire identity authentication protocol and is repeatedly executed during actual operation. Cryptographic operations with negligible computational cost, such as comparisons and multiplications, are excluded from the evaluation due to their limited impact on overall performance.

The comparison results are presented in Table IV. In particular, the scheme proposed in [24], which is an anonymous identity authentication method leveraging edge computing, incurs a total computational cost of $T_b + 9T_m + T_e + 10T_h + 3T_a = 23.47$ ms. Scheme [25] proposes a multi-server authentication protocol designed for cloud-edge collaborative IoT environments, with a computational cost of $7T_m + 16T_h + T_a = 13.946$ ms. Scheme [26] introduces an IoT device authentication approach that combines ECC with blockchain, incurring a computational cost of $6T_m + 8T_h = 11.892$ ms. In contrast, the proposed BTIA scheme integrates blockchain and ECC while introducing cluster head nodes to facilitate authentication between devices, thereby reducing the computational cost to $T_m + 4T_e + T_h + T_a = 5.338$ ms. Experimental evaluation of the aforementioned schemes demonstrates that the proposed BTIA scheme significantly

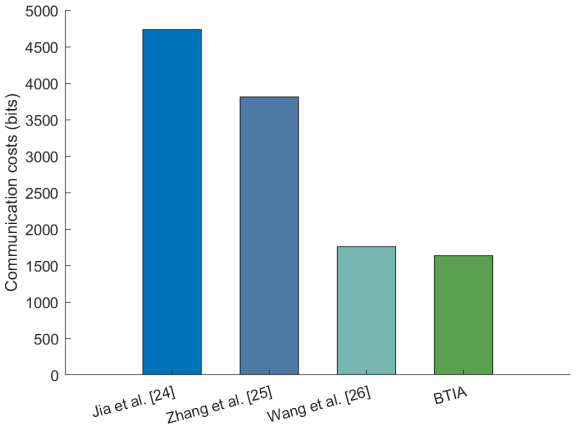

Fig. 11. Communicational costs comparison

TABLE V
COMPARISON OF DIFFERENT SCHEMES

Scheme	[24]	[25]	[26]	BTIA
Decentralization	N	Y	Y	Y
Confidentiality	Y	Y	Y	Y
Lightweight	N	N	Y	Y
Trust management	N	N	N	Y
Identity privacy	Y	Y	Y	Y
Consensus efficiency	N	N	N	Y

outperforms existing approaches in terms of computational complexity and execution time. This superior performance highlights its enhanced applicability in resource-constrained IoT environments. Therefore, BTIA exhibits a distinct advantage in computational efficiency.

The evaluation of communication overhead also omits the registration phase, concentrating on the authentication phase. We assume the element size in the cyclic group $|G|$ is 1024 bits, the element size in the finite field $|Z_q|$ is 160 bits, the output length of the hash function $|H|$ is 256 bits, the user identity $|ID|$ is 256 bits, and the timestamp $|T|$ is 32 bits. Based on these parameters, the communication overhead for the authentication phase of each scheme is calculated. Specifically, the protocol by Jia et al. [24] incurs a communication cost of $4|G|+2|T|+2|Z_q|+|ID| = 4736$ bits. The scheme proposed by Zhang et al. [25] requires $3|G| + 2|T| + |Z_q| + 6|H| = 3808$ bits. Wang et al.'s scheme [26] results in a communication overhead of $|G| + 4|T|+|Z_q|+3|H| = 1760$ bits. In contrast, the BTIA scheme requires only $|G| + |T| + 4|Z_q| + |H| = 1632$ bits. As illustrated in Figure 11, our proposed scheme demonstrates a clear advantage in terms of minimizing communication overhead.

The comparative analysis between the proposed BTIA scheme and existing authentication protocols highlights its data security and practical applicability. In Table V, "Y" and "N" denote whether a specific functional feature is supported. It can be seen that BTIA offers significant advantages in authentication. Compared with the identity authentication scheme designed for mobile edge computing environments [24], BTIA provides a decentralized data storage solution through blockchain technology, thereby eliminating the reliance on centralized servers for data storage. Zhang et al. [25] integrated Physical Unclonable Functions (PUFs) with blockchain to facilitate the secure sharing of physical identities. However, their scheme does not incorporate any improvements to the underlying consensus algorithm. The authentication scheme proposed by Wang et al. [26] ensures the authenticity of user identities and the integrity of transmitted data. Nevertheless, it lacks the support of trust management techniques and does not utilize edge node assistance, which results in relatively high computational overhead. BTIA not only preserves identity privacy and ensures data security, but also incorporates a trust management mechanism to govern cluster operations, thereby mitigating the threats posed by malicious nodes. Moreover, BTIA integrates an optimized consensus algorithm to improve the efficiency of reaching agreement among distributed nodes.

VII. CONCLUSION AND FUTURE WORK

To address the challenges of identity authentication and data sharing in the Internet of Things (IoT), we propose a novel identity authentication scheme, BTIA, which integrates blockchain technology, cryptographic methods, and trust management mechanisms. In BTIA, hashed pseudonyms are employed to protect the identity privacy of IoT devices. A comprehensive trust evaluation system is established to calculate and update the trust scores of both cluster head nodes and device nodes. Furthermore, the entire authentication and data sharing process is recorded on the blockchain to ensure data authenticity and traceability. Theoretical analysis demonstrates that BTIA ensures the confidentiality, integrity, availability, and privacy of IoT device interactions. Experimental results indicate that the Gas consumption, computational cost, and communication overhead of BTIA remain within acceptable limits.

REFERENCES

[1] O. Aouedi, T.-H. Vu, A. Sacco, D. C. Nguyen, K. Piamrat, G. Marchetto, and Q.-V. Pham, "A survey on intelligent internet of things: Applications, security, privacy, and future directions," 2024. [Online]. Available: https://arxiv.org/abs/2406.03820

[2] C. Saadouni, S. E. Jaouhari, N. Tamani, S. Ziti, L. Mroueh, and K. E. Bouchti, "Identification techniques in the internet of things: Survey, taxonomy and research frontier," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2025.

[3] I. Cetintav and M. Tahir Sandikkaya, "A review of lightweight iot authentication protocols from the perspective of security requirements, computation, communication, and hardware costs," *IEEE Access*, vol. 13, pp. 37 703–37 723, 2025.

[4] M. K. Hasan, Z. Weichen, N. Safie, F. R. A. Ahmed, and T. M. Ghazal, "A survey on key agreement and authentication protocol for internet of things application," *IEEE access*, 2024.

[5] S. Szymoniak, "Key distribution and authentication protocols in wireless sensor networks: A survey," *ACM Computing Surveys*, vol. 56, no. 6, pp. 1–31, 2024.

[6] K. Wang, Y. Hong, Y. Li, R. Yan, and J. Feng, "A distributed zero-trust scheme for airborne wireless sensor networks using dynamic identity authentication," *Scientific Reports*, vol. 15, no. 1, p. 8036, 2025.

[7] M. Muhammad and G. A. Safdar, "V2x application server and vehicle centric distribution of commitments for v2v message authentication," *Ad Hoc Networks*, vol. 167, p. 103701, 2025.

[8] A. Deep, A. Perrusquía, L. Aljaburi, S. Al-Rubaye, and W. Guo, "A novel distributed authentication of blockchain technology integration in iot services," *IEEE Access*, vol. 12, pp. 9550–9562, 2024.

[9] X. Lu, D. Niyato, H. Jiang, D. I. Kim, Y. Xiao, and Z. Han, "Ambient backscatter assisted wireless powered communications," *IEEE Wireless Communications*, vol. 25, no. 2, pp. 170–177, 2018.
[10] S. Pattar, R. Buyya, K. R. Venugopal, S. Iyengar, and L. Patnaik, "Searching for the iot resources: Fundamentals, requirements, comprehensive review, and future directions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2101–2132, 2018.
[11] R. Yang, F. R. Yu, P. Si, Z. Yang, and Y. Zhang, "Integrated blockchain and edge computing systems: A survey, some research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1508–1532, 2019.
[12] R. Yadav and G. Baranwal, "An efficient trust management using feedback credibility evaluation method in fog computing," *Simulation Modelling Practice and Theory*, vol. 120, p. 102610, 2022.
[13] B. Kaur, S. Dadkhah, F. Shoeleh, E. C. P. Neto, P. Xiong, S. Iqbal, P. Lamontagne, S. Ray, and A. A. Ghorbani, "Internet of things (iot) security dataset evolution: Challenges and future directions," *Internet of Things*, vol. 22, p. 100780, 2023.
[14] N. T. Y. Huan and Z. A. Zukarnain, "A survey on addressing iot security issues by embedding blockchain technology solutions: Review, attacks, current trends, and applications," *IEEE Access*, vol. 12, pp. 69 765–69 782, 2024.
[15] S. B. Sharma, I. Dhall, S. R. Nayak, and P. Chatterjee, "Reliable biometric authentication with privacy protection," in *Advances in Communication, Devices and Networking: Proceedings of ICCDN 2021*. Springer, 2022, pp. 233–249.
[16] S. U. Jan, F. Qayum, and H. U. Khan, "Design and analysis of lightweight authentication protocol for securing iod," *Ieee access*, vol. 9, pp. 69 287–69 306, 2021.
[17] Z. Cui, X. Fei, S. Zhang, X. Cai, Y. Cao, W. Zhang, and J. Chen, "A hybrid blockchain-based identity authentication scheme for multi-wsn," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 241–251, 2020.
[18] S. Gao, Q. Su, R. Zhang, J. Zhu, Z. Sui, and J. Wang, "A privacy-preserving identity authentication scheme based on the blockchain," *Security and Communication Networks*, vol. 2021, no. 1, p. 9992353, 2021.
[19] V. Dehalwar, M. L. Kolhe, S. Deoli, and M. K. Jhariya, "Blockchain-based trust management and authentication of devices in smart grid," *Cleaner Engineering and Technology*, vol. 8, p. 100481, 2022.
[20] W. Dai, C. Dai, K.-K. R. Choo, C. Cui, D. Zou, and H. Jin, "Sdte: A secure blockchain-based data trading ecosystem," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 725–737, 2019.
[21] M. Zhaofeng, W. Xiaochang, D. K. Jain, H. Khan, G. Hongmin, and W. Zhen, "A blockchain-based trusted data management scheme in edge computing," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2013–2021, 2019.
[22] K. Gai, Y. Wu, L. Zhu, L. Xu, and Y. Zhang, "Permissioned blockchain and edge computing empowered privacy-preserving smart grid networks," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7992–8004, 2019.
[23] J. Kang, R. Yu, X. Huang, M. Wu, S. Maharjan, S. Xie, and Y. Zhang, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE internet of things journal*, vol. 6, no. 3, pp. 4660–4670, 2018.
[24] X. Jia, D. He, N. Kumar, and K.-K. R. Choo, "A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing," *IEEE Systems Journal*, vol. 14, no. 1, pp. 560–571, 2019.
[25] Y. Zhang, B. Li, B. Liu, Y. Hu, and H. Zheng, "A privacy-aware pufs-based multiserver authentication protocol in cloud-edge iot systems using blockchain," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 13 958–13 974, 2021.
[26] W. Wang, B. Yan, B. Chai, R. Shen, A. Dong, and J. Yu, "EBIAS: ECC-enabled blockchain-based identity authentication scheme for iot device," *High-Confidence Computing*, vol. 5, no. 1, p. 100240, 2025.