A Blockchain-Assisted Cross-Domain Mutual Authentication Scheme for the Internet of Things

Xiaoyu Du, Zhenxiang Huo, Ying Du*, Linyu Yang and Zhijie Han

***Abstract*—In the Internet of Things (IoT) environment, mobile nodes are typically distributed across different domains and managed by one or more fog nodes. With the rapid increase in the number of devices and the diversification of application scenarios, the demand for cross-domain access by mobile nodes is continuously growing. Identity authentication is a crucial step to ensure the security of cross-domain access in IoT. However, existing authentication schemes are mostly based on traditional bilinear pairing, which is computationally intensive, and often do not consider the trustworthiness of fog nodes. To address the problem of untrustworthy fog nodes in different domains, a blockchain-assisted cross-domain mutual authentication scheme is proposed to solve the above problems. First, the scheme introduces a trust management mechanism to ensure the trustworthiness of fog nodes. Second, the scheme uses symmetric polynomials instead of traditional bilinear pairing to improve computational efficiency, while utilizing blockchain technology to ensure data security and traceability. The scheme is designed with two phases of pre-authentication and cross-domain authentication to avoid complex duplicate authentication. The final security performance analysis indicates that this scheme offers more comprehensive security features and higher efficiency compared to existing schemes.**

***Index Terms*—Internet of Things, Blockchain, Cross-domain authentication, Trust management, Symmetric polynomial.**

I. Introduction

THE rapid development of the Internet of Things (IoT) [1] is changing the way we live, and IoT applications have become ubiquitous, from smart homes to smart cities [2]. However, the widespread use of IoT [3] also brings unprecedented security and privacy challenges [4]. The IoT environment is highly heterogeneous and distributed, with a large number and variety of devices, including sensors, actuators, and smart devices. These devices interact through different communication protocols and data formats, forming a complex network ecosystem. In addition, IoT devices are often resource-constrained, such as computing power, storage space and battery life, making it necessary to consider their efficiency and low resource consumption when designing security solutions. To address the computational and storage demands in the IoT environment, Fog Computing has emerged. Fog Computing brings computing and storage resources closer to the network edge, offering lower latency and higher bandwidth utilization. In IoT applications, fog nodes [5] can perform data processing and analysis in close proximity to the data source, thus improving the responsiveness and efficiency of the system. The introduction of fog nodes not only eases the load on the cloud computing centers but also provides new opportunities to achieve distributed security management.

In an IoT environment, devices and users are typically distributed across different domains, with each domain managed by one or more fog nodes that oversee the mobile nodes within the domain. In such cases, mobile nodes and fog nodes need to securely access and communicate across different domains. Therefore, ensuring secure communication and reliable authentication between nodes is crucial, making cross-domain access authentication a key component in ensuring the security of IoT systems [6].

Traditional IoT authentication schemes [7], [8], [9] often rely on bilinear pairing to achieve security functions [10]. While bilinear pairing can provide robust encryption capabilities, its high computational complexity makes it unsuitable for resource-constrained IoT devices. Additionally, traditional authentication schemes face multiple challenges in cross-domain access, including the transmission of authentication information, privacy protection, and dependence on centralized trust authorities. Additionally, in a distributed IoT environment, fog nodes serving as domain administrators may become untrustworthy due to certain circumstances, which makes it difficult to ensure the trustworthiness of some fog nodes. Untrustworthy fog nodes may engage in malicious activities, such as stealing sensitive information, tampering with data, or providing denial of service. Therefore, a decentralized cross-domain access authentication scheme is needed to meet the security requirements in an IoT environment [11].

To address the aforementioned issues, this paper introduces symmetric polynomials [12] as a replacement for bilinear pairings. Symmetric polynomials not only provide similar security functionalities but also offer higher computational efficiency, making them more suitable for resource-constrained IoT devices. Blockchain technology [13], [14], [15], as a decentralized distributed ledger technology, features transparency, immutability, and traceability, and has been widely adopted in fields such as finance and supply chain management. In the IoT environment, blockchain can be used to store and manage device identity and authentication information, ensuring data integrity and security. By leveraging blockchain technology,

Manuscript received May 17, 2025; revised August 15, 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).

Zhenxiang Huo is a postgraduate student at the School of Computer and Information Engineering, Henan University, Kaifeng 475001, China (e-mail: hzxcurry@henu.edu.cn).

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

Linyu Yang is a postgraduate student at the School of Computer and Information Engineering, Henan University, Kaifeng 475001, China (e-mail: yanglinyu@henu.edu.cn).

Zhijie Han is a professor at the School of Software, Henan University, Kaifeng 475001, China (e-mail: hanzhijie@126.com).

a secure, reliable, and efficient solution for cross-domain access authentication can be provided. Additionally, to tackle the issue of untrustworthy fog nodes, a trust management mechanism for fog nodes is designed to ensure the reliability of fog nodes within the system.

The main contributions of this paper are as follows:

1. A blockchain-assisted mutual authentication scheme called BAMA is proposed, in which a binary symmetric polynomial is used instead of the traditional bilinear pairing, which significantly reduces the demand for computational resources and improves efficiency. In addition, in BAMA, a blockchain network is introduced as an auxiliary mechanism to store node registration information, authentication information, and trust scores of fog nodes into the blockchain, which ensures the integrity and authenticity of the data.

2. The proposed BAMA authentication scheme is divided into two phases: pre-authentication and cross-domain authentication. In the pre-authentication phase, mutual authentication between cross-domain mobile nodes and local fog nodes is achieved, ensuring the trustworthiness of cross-domain nodes. Upon successful authentication, the local fog node provides the mobile node with a ticket and a cross-domain Authentication Key (IAK), offering flexibility for subsequent cross-domain authentication without the need for repetitive and cumbersome authentication processes.

3. In BAMA, a trust management mechanism is designed to ensure the credibility of fog nodes within each domain. This mechanism employs the BLS aggregate signature algorithm, which significantly reduces the number of signatures while ensuring the accuracy of the trust score calculation for fog nodes.

The remainder of this paper is organized as follows:

Section II discusses the related work, Section III provides a detailed introduction to the proposed BAMA scheme, Section IV presents the theoretical proof and security analysis of BAMA, and Section V compares its performance with other existing schemes. Finally, Section VI concludes the paper.

II. Related Work

Many related literatures on authentication have emerged in recent years. For fog computing environment, Lin et al. [16] proposed a key negotiation and user authentication scheme in fog computing environment, which can establish a secure session between different entities, and users can achieve cross-domain access to other fog servers, and satisfy perfect forward security and anonymity. However, its computational process is overly complex and does not consider the trustworthiness of fog nodes. Guo et al. [17] proposed an anonymous handover authentication scheme called FogHA, which achieves mutual authentication and key agreement between adjacent fog nodes and mobile devices. FogHA uses lightweight cryptographic primitives to eliminate redundant authentication messages with the cooperation of fog nodes. However, it similarly does not consider the trustworthiness of fog nodes.

He et al. [18] proposed a cross-domain authentication scheme for mobile healthcare, which allows mutual authentication and session key generation between patients registered at different medical centers. Zhou et al. [19] proposed a provably secure cross-domain authentication protocol for IoT mobile nodes, which completes the verification of the mobile node's identity legitimacy by the remote domain authentication server through a single round of message interaction. Shashidhara, R. et al. [20] proposed an authentication scheme for mobile environments using lightweight cryptographic primitives, ensuring user anonymity, privacy, and security. Meng et al. [21] proposed a secure anonymous key agreement protocol for cloud computing that addresses poor randomness, binding public keys to entity identities without the need for certificates, thereby solving the certificate management problem. However, these schemes all utilize bilinear pairing, public key encryption, and symmetric encryption, which require substantial computation. Jegadeesan S et al. [22] proposed an anonymous mutual authentication technique for mobile cloud computing in smart cities, allowing mobile users to access services from different service providers using a single private key without relying on a trusted third party. However, this scheme lacks an effective key management mechanism and does not address secure communication between multiple groups of users. Ali R et al. [23] proposed a three-factor authentication scheme for smart agriculture based on fuzzy biometric extraction, smart cards, and password credentials. The scheme supports user device revocation and dynamic node addition but is vulnerable to user impersonation, smart card theft, and denial-of-service (DoS) attacks.

With the rise of blockchain technology, Shen M et al. [24] proposed a blockchain-based cross-domain industrial IoT security authentication and key agreement mechanism. This scheme introduces a consortium blockchain as a trusted platform for sharing specific domain information, supporting identity revocation mechanisms, and protecting entity privacy. However, the need for frequent data exchanges increases communication overhead. Feng C et al. [25] proposed a blockchain-based cross-domain authentication scheme for smart 5G UAV internet that enables reliable communication between entities from different domains by creating multi-signature smart contracts, but the high transaction failure rate of this scheme may cause the device to initiate multiple transactions to complete the authentication, and the use of smart contracts to read and write data on the blockchain introduces delays. Gauhar et al. [26] proposed a blockchain-based IoT permission authorization and access control framework. This framework protects the privacy of external users by allowing them to obtain authentication within their parent IoT domain, with authentication based on authorization policies stored on the blockchain. While the scheme supports cross-domain authentication, it is inefficient and lacks flexibility. Zhang S et al. [27] proposed a group signature scheme aimed at verifying blockchain blocks to address attacks on consensus algorithms, while also providing a mobile device authentication solution. However, if one member of the group crashes or goes offline, it can significantly disrupt the consensus process, leading to a decline in system performance.

In summary, when designing a cross-domain security authentication scheme suitable for the IoT environment, we should focus on the following key aspects: First, it is crucial to ensure that the privacy of device information is effectively protected to prevent the leakage or misuse of sensitive

data. Secondly, the cross-domain authentication scheme should enable effective mutual authentication between the communicating parties, ensuring the legitimacy of identities, thereby enhancing the security of cross-domain communication. In addition, the trustworthiness of the intra-domain authentication server is also crucial, as it directly impacts the overall security defense capability of the system. Our research will comprehensively consider these factors, aiming to design an authentication mechanism that not only meets the complex cross-domain security requirements of the IoT but also enhances the robustness and practicality of the system.

III. Details of BAMA

In this subsection, we first introduce the network model of BAMA and then describe the details of the BAMA scheme. The symbols used in the BAMA scheme are listed in Table I.

TABLE I
Symbols Used in the BAMA Scheme

Symbol	Description
TC	IoT Trust Center
MD	Mobile Device
Fog	Fog Node
FID	Pseudonym of Fog Node and Mobile Device
G	Cyclic additive group
g	Generator of G
q	Large prime numbers
h	Hash Function
T_{fog}	Trust Score of Fog Nodes
pk_{TC}	Public Key of Trust Center
sk_{TC}	Private Key of Trust Center
pk_{MD}	Public Key of Mobile Device MD
sk_{MD}	Private Key of Mobile Device MD
pk_{fog}	Public Key of Fog Node
sk_{fog}	Private Key of Fog Node
ID	Real Identity of Node
$F(x,y)$	Binary t-degree Symmetric Polynomial
TS	Timestamp

A. Network Model

The system model of the BAMA scheme is shown in Figure 1. It includes the following entities: Trust Center (TC), Fog Node (Fog), Mobile Device (MD), and Blockchain (BC).

Fog Node: Fog nodes typically possess strong computational and storage capabilities. They are responsible for data processing and storage near end-user locations to reduce latency and provide faster service response. Before joining the system, fog nodes need to register with the TC. They manage mobile devices within their domain and generate tickets for mobile devices for cross-domain authentication and store this ticket on the blockchain.

Mobile Device: Mobile devices include smartphones, cars, drones, etc. Before joining the system, they need to register with the TC to obtain a ticket for pre-authentication. They establish secure communication with trusted fog nodes by querying the trust value of fog nodes in the blockchain. Only after completing pre-authentication can they perform cross-domain authentication. These mobile devices have certain storage and computational units, which can be used to store information such as tickets.

Blockchain: The blockchain provides a decentralized and tamper-proof ledger used to record all authentication-related transactions and data, such as the hash values of tickets used for mobile device authentication, the trust scores of fog nodes, and the identity information of nodes.

Trust Center: As a trusted center, it is responsible for generating pseudonyms for nodes and tickets for mobile nodes used for intra-domain authentication. It also removes fog nodes that fall below the trust threshold from the system and stores the identity information of the nodes on the blockchain to ensure system traceability. In addition, we encrypt the communication between the trusted center and the fog nodes and mobile devices through the TLS protocol in our model, and establish a secure channel through the handshake process to ensure the confidentiality, integrity of the data and the authenticity of the identities of the two parties, which can effectively resist potential attacks and threats.

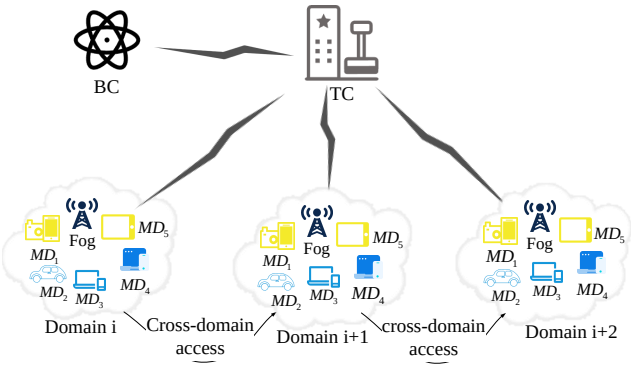

Fig. 1. Network model.

B. System Initialization

The TC selects a cyclic additive group G of order q defined on an elliptic curve E over a finite field Fp, where p is a large prime number, and chooses a generator g.

The TC selects a random number sk_{TC} as its private key and computes the public key $pk_{TC} = sk_{TC} \cdot g$. The TC selects three hash functions $h_1 : \{0,1\}^* \longrightarrow Z_q^*$, $h_2 : \{0,1\}^* \longrightarrow \{0,1\}^l$, $h_3 : \{0,1\}^* \longrightarrow \{0,1\}^k$. Additionally, it randomly selects a binary t-order symmetric polynomial $F(x,y) = (\sum_{m,n=0}^{t} a_{m,n} x^m y^n)$ mod p over the finite field.

The TC then publicly discloses the system parameters $param$ = (p, q, g, G, h), but keeps sk_{TC} and $F(x,y)$ confidential.

C. Fog Node Registration

The registration process of fog nodes is shown in Figure 2 and is detailed as follows:

Firstly the fog node generates a timestamp TS_1 and sends its real identity ID_{fog_j} and TS_1 to the TC over a secure channel.

Upon receipt, the TC first verifies whether $|TS_1 - TS_1^*| \leq \Delta t$ holds to determine the freshness of the message. If it holds, it verifies whether ID_{fog_j} is registered or not in

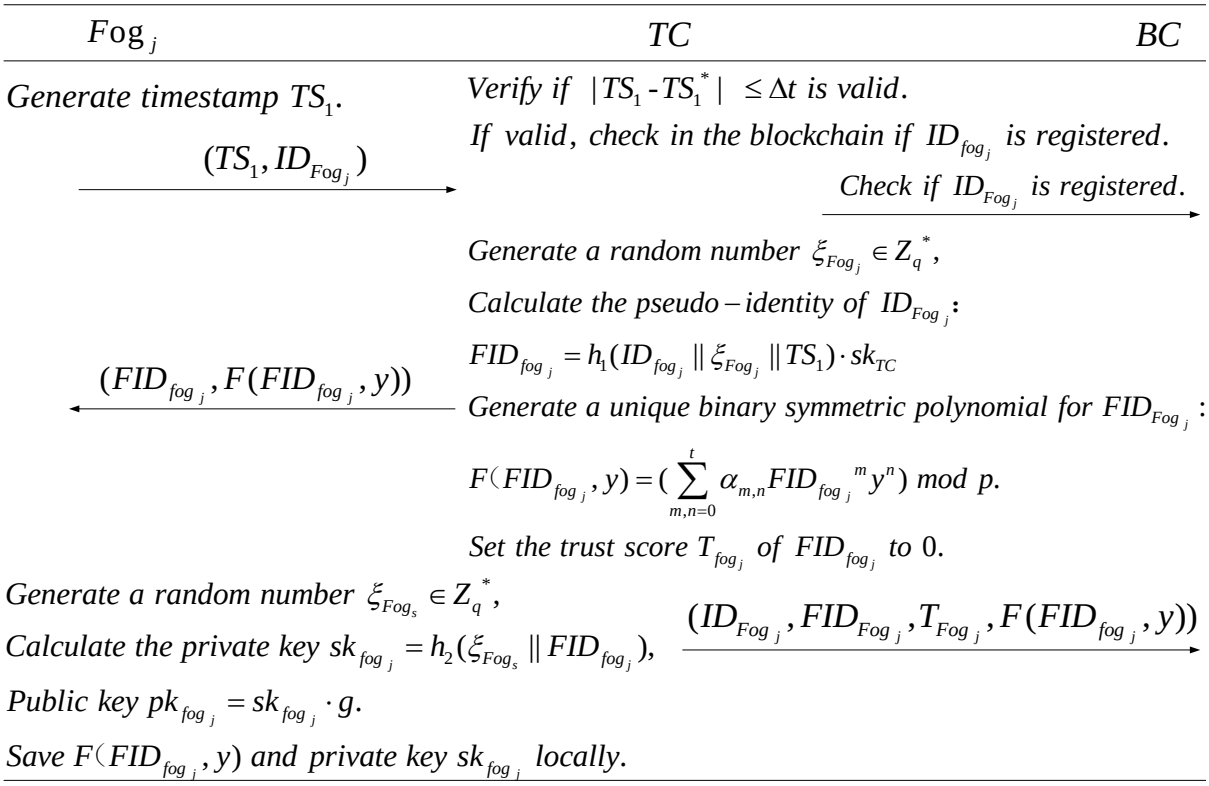

Fig. 2. Fog Node Registration Flowchart.

the blockchain, and if it has been registered, the request is rejected. Otherwise, it will select a random number $\xi_{fog_j} \in Z_q^*$ and compute the pseudonym for ID_{fog_j} as $FID_{fog_j} = h_1(ID_{fog_j} \parallel \xi_{fog_j} \parallel TS_1) \cdot sk_{TC}$, and generate a unique binary symmetric polynomial for the fog node FID_{fog_j} according to $F(FID_{fog_j}, y) = (\sum_{m,n=0}^{t} a_{m,n} FID_{fog_j}^m y^n) \, mod \, p$.

Subsequently the TC sets the trust score of the newly registered fog node FID_{fog_j} to $T_{fog_j} = 0$, when the score of the subsequent FID_{fog_j} is lower than the threshold thr, then FID_{fog_j} is set to be a malicious node, and the communication interaction function cannot be performed in the case that FID_{fog_j} becomes a malicious node. The computation of the trust score of the fog node is updated by the trust management mechanism proposed in subsection 3.7.

The TC returns the pseudonym FID_{fog_j} and the binary symmetric polynomial $F(FID_{fog_j}, y)$ to the fog node. The TC also packages and stores the identity information of the registered FID_{fog_j} in the blockchain, including the fog node's real identity ID_{fog_j}, pseudonym FID_{fog_j}, binary symmetric polynomial $F(FID_{fog_j}, y)$, and the fog node's trust score T_{fog_j}.

Finally, the fog node generates a random number $\xi_{fog_s} \in Z_q^*$ and computes its own private key $sk_{fog_j} = h_2(\xi_{fog_s} \parallel FID_{fog_j})$ and public key $pk_{fog_j} = sk_{fog_j} \cdot g$.

D. Mobile Device Registration

The registration process for mobile devices is shown in Figure 3 and is detailed as follows:

The mobile device MD_i first selects a timestamp TS_2 and sends a registration request to the TC, including TS_2 and its real identity ID_{MD_i}.

Upon receipt, the TC first verifies the freshness of the timestamp TS_2, i.e., verifies whether $|TS_2 - TS_2^*| \leq \Delta t$ is valid, and if it is valid, it checks whether the device has been registered, and if it has been registered, it rejects the request. If the device is not registered, the TC selects a random number $\tau_{MD_i} \in Z_q^*$ and computes the pseudonym for the mobile device as $FID_{MD_i} = h_1(\tau_{MD_i} \parallel ID_{MD_i} \parallel TS_2) \cdot sk_{TC}$. After that the pseudonym FID_{MD_i} of the mobile node is sent to the mobile device MD_i over a secure channel.

After receiving the message, the mobile device MD_i generates a random number $\tau_{MD_s} \in Z_q^*$ and computes its private key $sk_{MD_i} = h_2(\tau_{MD_s} \parallel FID_{MD_i})$ and the public key as $pk_{MD_i} = sk_{MD_i} \cdot g$. It then searches the blockchain for the identity information of the fog node FID_{fog_j} it wants to connect to. If the trust score of FID_{fog_j} exceeds the trust threshold thr, MD_i generates a timestamp TS_3 and sends the fog node's pseudonym FID_{fog_j} along with TS_3 to the TC, indicating its intention to connect to that fog node. If the trust score of FID_{fog_j} is below the threshold, MD_i will search for another fog node whose trust score exceeds the trust threshold to connect to.

After receiving it, the TC computes an authentication key $IAK_{ij} = F(FID_{MD_i}, FID_{fog_j})$ based on the pseudonym FID_{MD_i} of the MD_i and the pseudonym FID_{fog_j} of the fog node received. It then generates a ticket $ticket_1$ and sends it to FID_{MD_i} for initial authentication with the fog node FID_{fog_j}. This ticket includes the pseudonym of the fog node FID_{fog_j}, the pseudonym of the mobile device FID_{MD_i}, and the public key of the mobile device pk_{MD_i}. The ticket is signed with the TC's private key, and $n_{i,j} = h_3(IAK_{ij} || ticket_1)$ is calculated. Subsequently, the TC sends the authentication key IAK_{ij} and the ticket

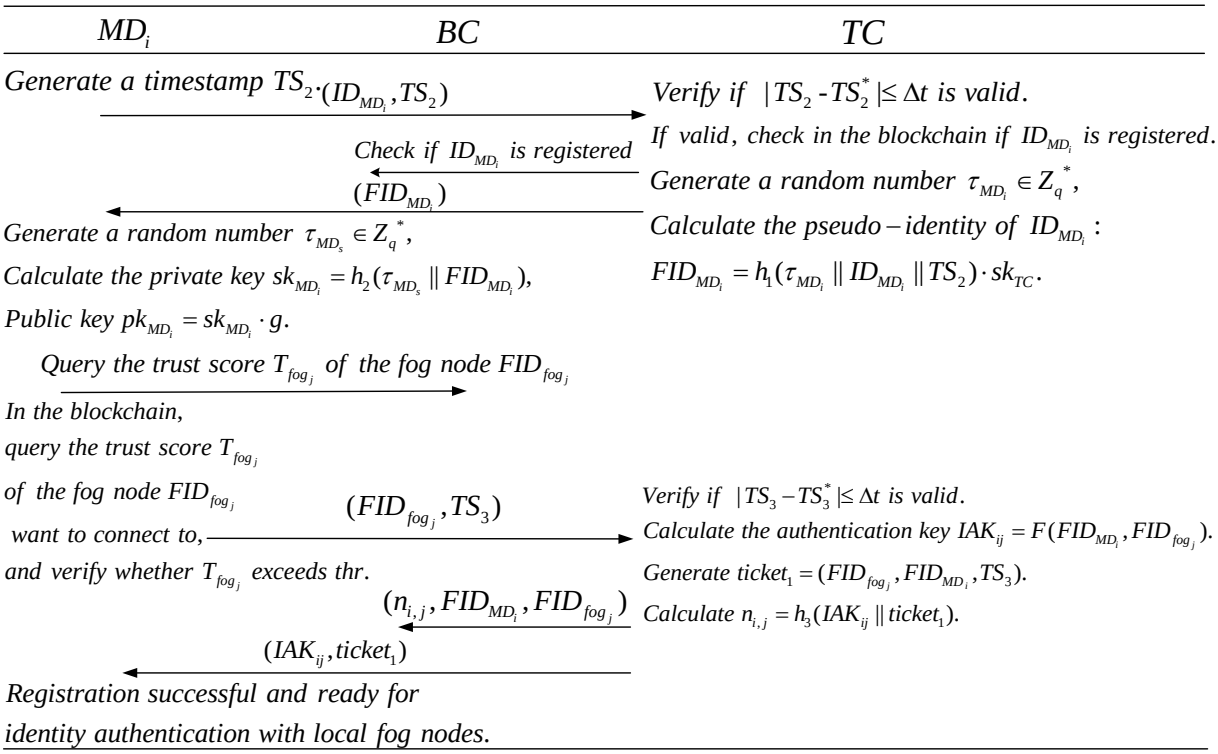

Fig. 3. Mobile Node Registration Flowchart.

$ticket_1$ to the mobile node. It also stores the authentication information $(n_{i,j}, FID_{MD_i}, FID_{fog_j})$ in the blockchain, which will be used for initial verification of the mobile node's legitimacy during authentication with the local fog node. This completes the registration process for the mobile node.

E. Pre-Authentication

This subsection will detail the mutual authentication process between mobile nodes and local fog nodes, which serves as a pre-authentication step before the mobile nodes perform cross-domain authentication. As shown in Figure 4, the specific process is as follows:

The mobile node MD_i first generates two random numbers rn_1 and k_{MD_i}, and a timestamp TS_4. It then computes $m_1 = h_3(rn_1 \parallel IAK_{ij} \parallel TS_3)$, $K_{MD_i} = k_{MD_i} \cdot g$, $m_2 = m_1 \oplus h_3(K_{MD_i} \parallel IAK_{ij})$, $m_3 = k_{MD_i} + h_3(K_{MD_i} \parallel IAK_{ij} \parallel m_1 \parallel m_2) \cdot sk_{MD_i}$. Then, it sends $(K_{MD_i}, m_2, m_3, TS_4)$ along with the ticket $ticket_1$ to the local fog node FID_{fog_j}.

Upon receiving the message, the fog node FID_{fog_j} first verifies the freshness of the message by checking whether $|TS_4 - TS_4^*| \leq \Delta t$ holds. After passing this verification, the fog node FID_{fog_j} searches the blockchain for the authentication information regarding the mobile node $(n_{i,j}, FID_{MD_i}, FID_{fog_j})$ and computes $IAK_{ji} = F(FID_{fog_j}, FID_{MD_i})$ and $n_{i,j}^{'} = h_3(IAK_{ji} \parallel ticket_1)$.

It verifies whether $n_{i,j}^{'}$ is equal to $n_{i,j}$. If they are equal, it indicates that the mobile node is legitimate and that the data has not been tampered with or forged. Then, it continues to compute $m_1^{'} = m_2 \oplus h_3(K_{MD_i} \parallel IAK_{ji})$. It verifies whether $m_3 \cdot g = K_{MD_i} + h_3(K_{MD_i} \parallel IAK_{ji} \parallel m_1^{'} \parallel m_2) \cdot pk_{MD_i}$ holds. If this is true, it indicates that the message has not been tampered with during transmission, thus completing the one-way authentication of the mobile node FID_{MD_i} by the local fog node FID_{fog_j}.

Next, FID_{fog_j} generates two random numbers rn_2 and k_{fog_j}, as well as a timestamp TS_5, and computes $m_5 = h_3(rn_2 \parallel IAK_{ji} \parallel TS_5)$, $K_{fog_j} = k_{fog_j} \cdot g, K_{i-j} = K_{MD_i} \cdot k_{fog_j}$ and $m_6 = h_3(K_{i-j} \parallel IAK_{ji} \parallel K_{fog_j} \parallel m_5 \parallel TS_5) \cdot k_{fog_j}$. Then, it sends $(m_5, K_{fog_j}, m_6, TS_5)$ to the mobile node FID_{MD_i}.

Upon receiving the message, the mobile node verifies the validity of the timestamp TS_5. It then computes $K_{i-j}^{'} = K_{fog_j} \cdot k_{MD_i}$ and verifies whether $m_6 \cdot g = h_3(K_{i-j}^{'} \parallel IAK_{ij} \parallel K_{fog_j} \parallel m_5 \parallel TS_5) \cdot K_{fog_j}$ holds. If this holds true, mutual authentication is successfully completed.

After completing the authentication, the mobile node FID_{MD_i} queries the blockchain for the trust score T_{fog_k} of the fog node FID_{fog_k} it intends to access cross-domain. If the trust score exceeds the threshold thr, it generates a timestamp TS_6 and sends a cross-domain access request to the local fog node FID_{fog_j}, which includes (FID_{fog_k}, TS_6).

Upon receiving the request, the fog node FID_{fog_j} verifies the validity of the message by checking whether $|TS_6 - TS_6^*| \leq \Delta t$ holds. Then, it calculates an authentication key $IAK_{jk} = F(FID_{fog_j}, FID_{fog_k})$ between itself and the receiving domain fog node FID_{fog_k}. A ticket $ticket_2 = (FID_{fog_j}, FID_{MD_i}, Timethr)$ is generated, where Timethr is the validity period of the ticket. The fog node signs the ticket contents using its private key sk_{fog_j} and computes $n_{fog_{j-k}} = h_3(IAK_{jk} \parallel ticket_2)$. Subsequently, it sends the authentication key IAK_{jk} and the ticket $ticket_2$ to the mobile node FID_{MD_i}. The authentication information $(n_{fog_{j-k}}, FID_{MD_i}, FID_{fog_k})$ is stored on the blockchain for preliminary verification by the receiving domain fog node

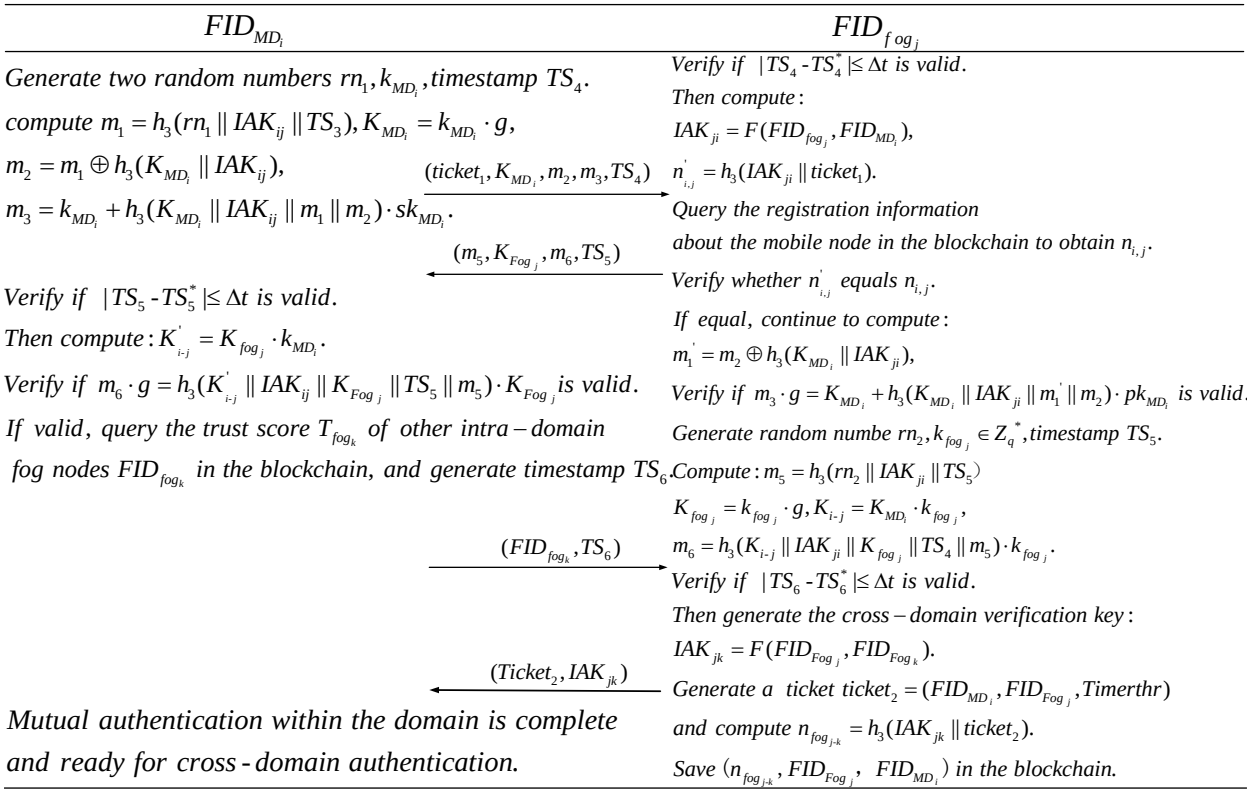

Fig. 4. Flowchart for mutual authentication within domain.

during cross-domain authentication, thus completing mutual authentication between the mobile node and the local fog node.

F. Cross-Domain Identity Authentication for Mobile Nodes

After completing local domain authentication, the mobile node can proceed with cross-domain authentication with a remote domain. As shown in Figure 5, the specific process is as follows:

The mobile node FID_{MD_i} first generates a random number $k_c \in Z_q^*$ and a timestamp TS_7. Then, it computes $K_c = k_c \cdot g$, $s_1 = pk_{fog_k} \cdot k_c$ and $\varphi_1 = h_3(K_c \| IAK_{jk} \| FID_{fog_k} \| FID_{MD_i} \| s_1 \| TS_7)$. Then, it sends the $(ticket_2, \varphi_1, s_1, TS_7)$ to the new regional fog node FID_{fog_k}.

Upon receiving the message the fog node FID_{fog_k} first verifies the freshness of the message by checking if $|TS_7 - TS_7^*| \le \Delta t$ holds. If the verification is successful, it further verifies the validity period timethr of the ticket to ensure that it remains fresh. Then, it computes $IAK_{kj} = F(FID_{fog_k}, FID_{fog_j})$ and $n_{fog_{j-k}}^{'} = h_3(IAK_{kj} \| ticket_2)$. After the computation, it queries the blockchain for the authentication information of the mobile node $(n_{fog_{j-k}}, FID_{MD_i}, FID_{fog_k})$ and verifies whether $n_{fog_{j-k}}^{'}$ is equal to $n_{fog_{j-k}}$. If the verification is successful, it indicates that the mobile node has completed authentication with the local fog node and has received authorization from the local fog node. Then, it computes $K_c^{'} = s_1 \cdot sk_{fog_k}^{-1}$, $\varphi_1^{'} = h_3(K_c^{'} \| IAK_{kj} \| FID_{fog_k} \| FID_{MD_i} \| s_1 \| TS_7)$. Verifies whether φ_1 is equal to $\varphi_1^{'}$. If the verification is successful, it indicates that the message is secure during transmission, thereby completing one-way authentication.

Subsequently, the fog node FID_{fog_k} generates a random number $k_d \in Z_q^*$ and a timestamp TS_8, and then computes $K_d = k_d \cdot g$, $s_2 = pk_{MD_i} \cdot k_d$, $K_e = h_3(K_d \| IAK_{kj} \| K_c^{'})$, generates the session key $SK_{ki} = h_3(K_d \| K_c^{'} \| K_e \| IAK_{kj})$, and then computes $\varphi_2 = h_3(SK_{ki} \| IAK_{kj} \| TS_8)$. Afterwards, it sends $(s_2, R_e, \varphi_2, TS_8)$ to the mobile node FID_{MD_i}.

Upon receiving the message, the mobile node FID_{MD_i} first verifies the validity of the message by checking if $|TS_8 - TS_8^*| \le \Delta t$ holds. If valid, it then computes $K_d^{'} = s_2 \cdot sk_{MD_i}^{-1}$, $K_e^{'} = h_3(K_d^{'} \| IAK_{jk} \| K_c)$ and computes the session key $SK_{ik} = h_3(K_d^{'} \| K_c \| K_e^{'} \| IAK_{jk})$ and $\varphi_2^{'} = h_3(SK_{ik} \| IAK_{jk} \| TS_8)$, and verifies whether $\varphi_2^{'}$ is equal to φ_2. If they are equal, it indicates that mutual authentication is successful and the correct session key has been generated, enabling secure cross-domain access.

G. Trust Management Mechanism for Fog Nodes

Since not all fog nodes in the network are trustworthy, we propose a trust management mechanism in this subsection to calculate and evaluate the trust scores of fog nodes within each domain. This ensures that the fog nodes in the system are secure and reliable. As shown in Figure 6, the details are as follows:

If the fog node is not registered, the TC will set the initial trust score of the fog node to 0 after the new registration of the fog node. Subsequently, based on the historical trust score of the fog node as well as the recommendation scores of the mobile node and the nearby fog nodes, the trust score of the fog node is computed and updated at regular intervals. When the trust score is below the threshold value thr, the fog node is defined as a malicious node and is removed from the

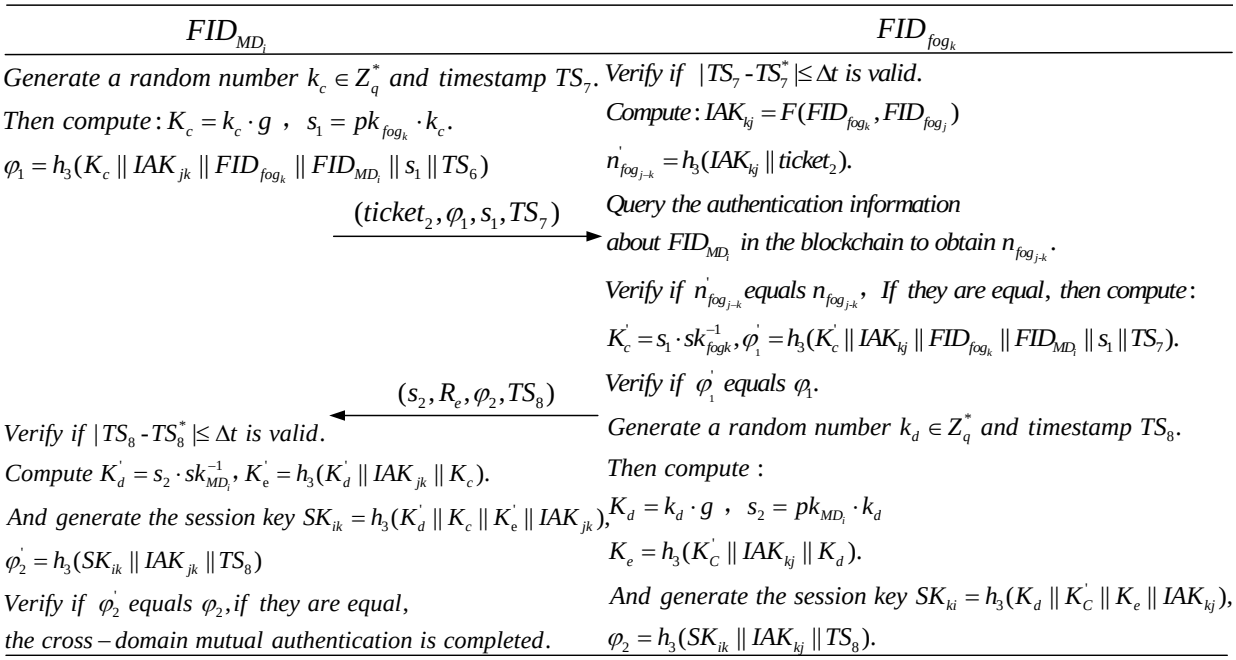

Fig. 5. Cross Domain Authentication Flowchart.

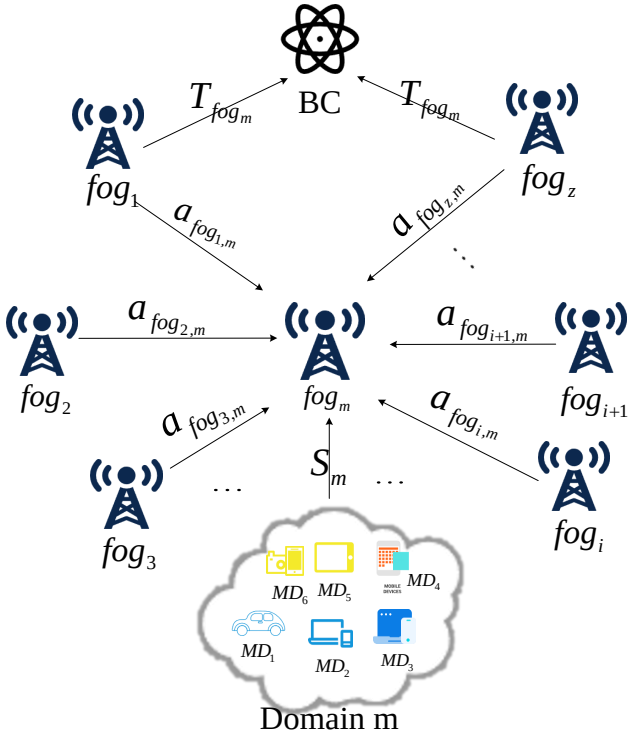

Fig. 6. Trust Management Model.

system.

For already registered fog nodes, the process to calculate and update the trust score is as follows:

Assuming that there are t mobile nodes in the same domain as fog node m, the recommendation score of mobile node i to fog node m is denoted as $s_{MD_i,m}$. After generating the recommendation score for fog node m, each mobile node will sign it with its private key sk_{MD_i}, denoted as $sig_{MD_i}(s_{MD_i,m})$, to ensure that the recommendation score cannot be tampered with by malicious nodes. Finally, the mobile device sends the signed recommendation score $sig_{MD_i}(s_{MD_i,m})$ to the blockchain network, and all the fog nodes can access this data.

All fog nodes in the blockchain network collect the recommendation scores $sig_{MD_i}(s_{MD_i,m})$ from all mobile nodes within the same domain as fog node m. The fog nodes then verify the signatures collected from the mobile nodes. To significantly reduce the number of signatures, the BLS aggregate signature algorithm is used to aggregate multiple signatures into a single signature:

$$\begin{aligned} sig^m_{MD_1,MD_2,\ldots,MD_t} = & f(sig_{MD_1}(s_{MD_1,m}), \\ & sig_{MD_2}(s_{MD_2,m}), \\ & \ldots, sig_{MD_i}(s_{MD_i,m}), \\ & \ldots, sig_{MD_t}(s_{MD_t,m})) \end{aligned}$$

At the same time, the public key PK_m used for verification is also aggregated:

$$PK_m = f(pk_{MD_1}, pk_{MD_2}, \ldots, pk_{MD_i}, \ldots, pk_{MD_t})$$

The BLS signature algorithm allows multiple signatures to be aggregated into a single signature, thereby greatly reducing the storage space and transmission bandwidth required for signatures. This is suitable for large-scale distributed systems and can improve the scalability and performance of the system.

If the fog node successfully verifies the signatures, it calculates the final recommendation score S_m of the mobile nodes for fog node m according to the following formula:

$$S_m = \frac{\sum_{i=1}^{t} \omega_i \cdot s_{MD_i,m}}{\sum_{i=1}^{t} \omega_i}$$

Where $s_{MD_i,m}$ is the recommendation score of the i-th mobile node for fog node m, ω_i is the weight of the i-th mobile node. Additionally, t is the total number of mobile nodes within the domain of fog node m.

Similarly, we assume that all fog nodes can obtain the final recommendation score A_m for fog node m from z other

nearby fog nodes:

$$A_m = \frac{\sum_{j=1}^{z} \vartheta_i \cdot a_{fog_j,m}}{\sum_{j=1}^{z} \vartheta_j}$$

Where $a_{fog_j,m}$ is the recommendation score of the j-th nearby fog node for fog node m, ϑ_j is the weight of the j-th fog node, and z is the total number of nearby fog nodes.

Subsequently, the fog node queries the historical trust score of fog node m from the blockchain and calculates the new trust score for fog node m using the following formula:

$$T_{fog_m} = \alpha \cdot T_{fog_m}^{old} + \beta \cdot S_m + \gamma \cdot A_m$$

Where T_{fog_m} represents the new trust score of the fog node, α, β, and γ are the weights, and $\alpha + \beta + \gamma = 1$. These weights are dynamically adjusted based on the system's operational status and historical data to more accurately reflect the current network environment. Additionally, $T_{fog_m}^{old}$ represents the historical trust score of fog node m stored on the blockchain; S_m is the final recommendation score from all mobile devices managed by fog node m; and A_m is the final recommendation score from other nearby fog nodes.

To avoid relying on a single node to store the new trust score, the PBFT consensus algorithm is employed to determine the final trust score and write it to the blockchain. The PBFT algorithm can tolerate a certain number of malicious or faulty nodes in a distributed network and ensures that all fog nodes reach a consensus through its consensus mechanism.

In the proposal phase, at the beginning of each consensus round, the system selects the fog node with the highest trust score as the leader (primary node), while other fog nodes act as ordinary nodes. The leader is responsible for initiating the proposal for the new trust score. The leader calculates the new trust score T_{fog_m}, signs it using its private key, and then sends this signed proposal to all ordinary nodes.

In the preparation phase, after all ordinary nodes receive the proposal from the leader, they first verify the validity of the proposal and the leader's signature. If the verification is successful, each node signs the proposal with its own private key and broadcasts a "prepare" message to all nodes, indicating its approval of the proposal.

In the commit phase, once a node receives a sufficient number of prepare messages (usually $2f + 1$, where f is the maximum number of malicious nodes the system can tolerate), it enters the commit phase. The node then broadcasts a commit message to the network.

In the final completion phase, once a node receives a sufficient number of commit messages, consensus is considered achieved. The trust score T_{fog_m} is then written to the blockchain, and all nodes confirm and record this trust score.

Subsequently, mobile nodes use these trust scores to select trustworthy fog nodes for connection, thereby enhancing the overall reliability and security of the system.

IV. Security Analysis

In this subsection, we conduct a comprehensive formal and informal security analysis of the proposed BAMA scheme.

A. Formal Security Using ROR Model

To verify the security of the session key SK_{ik} generated between the mobile device MD_i and the remote fog node fog_k during the cross-domain authentication phase in our proposed BAMA authentication scheme, we adopt the widely used Real-or-Random (RoR) security model[29]. This model evaluates the semantic security of the session key from a theoretical perspective by formulating a game-based framework. The adversary's capabilities are based on the standard Dolev-Yao (DY) attack model[30], which assumes full control over the communication channel, including eavesdropping, modification, forgery, and replay of messages. In the RoR model, the hash function h is modeled as a random oracle, accessible to all parties including the adversary. The queries that the adversary $\mathcal{A}$ can perform, following the Dolev-Yao model, are summarized in Table 2.

Entities: mobile node FID_{MD_i}, and remote domain fog node FID_{fog_k} as participants. Each entity is considered as a separate instance in each round of protocol run. For example, $\Pi_{MD_i}^t$ denotes the instance of mobile node FID_{MD_i} executing the protocol at the t-th time; $\Pi_{\text{fog}_k}^t$ denotes the instance of remote fog node FID_{fog_k} executing the protocol at the t-th time. These instances interact with each other for information and try to accomplish two-way authentication and session key negotiation.

Accepted State: When an instance receives the final message of the protocol, it is considered to be in the "Accepted State".

Partnering: There are three conditions for two instances to be partners with each other: 1) Both are in the accepted state. 2) Authenticate each other and share the same session key. 3) Be the communication counterpart of each other.

Freshness: an instance is fresh for Test queries if the adversary has not called Reveal or Corrupt queries on it or its partners.

Semantic security of session keys: in the RoR model, we define the semantic security of session keys through a challenge game. adversary $\mathcal{A}$ distinguishes the real session key from the pseudo-random number by multiple Test queries. Eventually, for the guess value c' returned by the Test queries, if $c' = c$, it means $\mathcal{A}$ wins the game, otherwise $\mathcal{A}$ loses the game. Thus, the advantage of the adversary in breaking the semantic security of the session key in polynomial time t is defined as:

$$Adv_{\mathcal{A}}^{BAMA}(t) = |2 \cdot Pr[Succ] - 1| = |2 \cdot Pr[c' = c] - 1|$$

where $Pr[Succ]$ denotes the success probability that $\mathcal{A}$ wins the game. $Pr[c' = c]$ denotes the probability that $\mathcal{A}$ guesses c correctly. If the value is sufficiently small, it means that the protocol satisfies semantic security, i.e., the adversary cannot effectively distinguish the real key from the pseudo-random number.

Random Oracle: in the RoR model, we assume that the hash function $h()$ is a publicly available randomized prediction machine, which can be accessed by all participants (including the adversary $\mathcal{A}$). When the adversary or any participant queries $h(x)$, if x has not been queried before, the random oracle generates a random output value k and stores the pair (x, k) in a hash table. For any subsequent query of the same input x, the oracle returns the same output k, thereby ensuring consistency in the query results.

TABLE II
QUERIES AND THEIR PURPOSES

Query	Purpose
Execute(Π^{t1}, Π^{t2})	Eavesdrop on the communication between two protocol instances
Send(Π^t)	Send a forged message to a protocol instance and receive the response
Reveal(Π^t)	Query the session key established by a protocol instance
CorruptFog(Π^t)	Simulate the compromise of a fog node and extract stored sensitive data.
Test(Π^t)	Test whether the returned session key is the real key or a random value: return the real session key if $c = 1$ and a random key if $c = 0$, enabling the adversary to distinguish between the two.

Theorem 1: Suppose that an adversary $\mathcal{A}$ attempts to break the semantic security of the proposed BAMA protocol within polynomial time t. Then, the following equation holds:

$$Adv_{\mathcal{A}}^{BAMA}(t) \leq \frac{q_h^2 + q_{\text{send}}^2}{|Hash|} + 2 \cdot Adv_H^{\text{coll}}(t)$$

where q_h denotes the number of times the adversary executes a hash query, $|Hash|$ denotes the size of the output space of the hash function, q_{send} denotes the number of times the call sends a query, and $Adv_H^{coll}(t)$ denotes the adversary's ability to find a hash collision in polynomial time t.

Proof: We construct a sequence of games G_n to analyze the advantage of the adversary $\mathcal{A}$, where $n \in [0, 3]$. Let $Succ$ denote the event that $\mathcal{A}$ successfully guesses the correct bit c and wins the game G_n. The details of each game are described as follows.

Game 0: At the beginning of the game, the adversary $\mathcal{A}$ is restricted to passively observing the protocol execution. It is allowed to perform the query $Test(\Pi_{MD_i}^i)$ in an attempt to guess the system's random bit c and determine whether the session key is real or a random value. The adversary's advantage in this game is equivalent to the semantic security of the session key:

$$Adv_{\mathcal{A}}^{BAMA}(t) = |2 \cdot Pr[Succ_0] - 1|$$

Game 1: In this game, the Execute query is introduced, allowing the adversary $\mathcal{A}$ to passively eavesdrop on communication transcripts. By analyzing the session key SK_{ik}, which is derived as $SK_{ik} = h(K_d^{'} \parallel K_c \parallel K_e^{'} \parallel IAK_{jk})$, we note that it is determined by two ephemeral secrets k_c and k_d, where $K_c = k_c \cdot g$, $K_d = k_d \cdot g$, and $K_e = h_3(K_d \parallel IAK_{jk} \parallel K_c^{'})$. To compromise the session key SK_{ik}, the adversary must solve the ECDHP and obtain the authentication key IAK_{jk}, both of which are infeasible within polynomial time. However, under a passive attack model, $\mathcal{A}$ can only access messages such as $(ticket_2, \varphi_1, s_1, TS_7)$ and $(s_2, R_e, \varphi_2, TS_8)$, which are insufficient for key recovery. Therefore, the adversary's advantage in Game 1 remains unchanged compared to Game 0:

$$|Adv_{\mathcal{A}.G_1}^{BAMA} - Adv_{\mathcal{A}.G_0}^{BAMA}| = Pr[Succ_1] - Pr[Succ_0] = 0$$

Game 2: The Send and Hash queries are introduced in this game, allowing the adversary to actively forge messages to deceive participants. After intercepting the messages $(ticket_2, \varphi_1, s_1, TS_7)$ and $(s_2, R_e, \varphi_2, TS_8)$, the adversary $\mathcal{A}$ attempts to modify some of the messages in order to forge a legitimate message that can pass verification. However, the adversary must know the authentication key IAK_{jk}, which is protected by a collision-resistant one-way hash function and verified between FID_{MD_i} and FID_{fog_k}. Therefore, to find a collision in the message digest, the adversary $\mathcal{A}$ can only attempt multiple hash queries. The hash collision probability is analyzed based on the birthday paradox:

$$\begin{aligned} |Adv_{\mathcal{A}.G_2}^{BAMA} - Adv_{\mathcal{A}.G_1}^{BAMA}| &= Pr[Succ_2] - Pr[Succ_1] \\ &=\leq \frac{q_h^2 + q_{\text{send}}^2}{2 \cdot |Hash|} \end{aligned}$$

Game 3: The CorruptFog attack is introduced in this game, where the adversary performs power analysis attacks to extract keys, pseudo-identities, and other sensitive information. If $\mathcal{A}$ attempts to compromise the session key $SK_{ik} = h(K_d^{'} \parallel K_c \parallel K_e^{'} \parallel IAK_{jk})$ of an uncompromised fog node, it must break the hash function $h(\cdot)$ by finding a collision, solve the ECDHP Problem , and obtain the authentication key IAK_{jk}, all within polynomial time. Therefore, we assume that the advantage of a hash collision attack $Adv_H^{coll}(t)$ is negligible, and thus we calculate the advantage difference between Game 3 and Game 2 as follows:

$$\begin{aligned} |Adv_{\mathcal{A}.G_3}^{BAMA} - Adv_{\mathcal{A}.G_2}^{BAMA}| &= Pr[Succ_3] - Pr[Succ_2] \\ &= Adv_H^{\text{coll}}(t) \end{aligned}$$

The final Game 3 simulates all the query operations available to the adversary. If the adversary still fails to break the protocol, the only remaining option is to guess the return value of the Test(Π^t) query, i.e., to determine whether it is a real session key or a random value. In this case, the success probability of the adversary equals the probability of correctly guessing the random bit c. Therefore:

$$Pr[Succ_3] = Pr[c = c'] = \frac{1}{2}$$

By combining the advantage differences across all games, we obtain:

$$\begin{aligned} Adv_{\mathcal{A}}^{BAMA}(t) &= |2 \cdot Pr[Succ_0] - 1| \leq |2 \cdot Pr[Succ_1] - 1| \\ &\leq \left| 2 \cdot \left(\frac{q_h^2 + q_{\text{send}}^2}{2 \cdot |Hash|} + Pr[Succ_2] \right) - 1 \right| \\ &\leq \left| 2 \cdot \left(\frac{q_h^2 + q_{\text{send}}^2}{2 \cdot |Hash|} + Adv_H^{coll}(t) + Pr[Succ_3] \right) - 1 \right| \\ &\leq \left| 2 \cdot \left(\frac{q_h^2 + q_{\text{send}}^2}{2 \cdot |Hash|} + Adv_H^{coll}(t) + \frac{1}{2} \right) - 1 \right| \\ &\leq \frac{q_h^2 + q_{\text{send}}^2}{|Hash|} + 2 \cdot Adv_H^{coll}(t) \end{aligned}$$

Based on the above derivation, under the assumption that the adversary makes at most q_h hash queries and $q_s end$ active message forgeries, and the hash function is collision-resistant, the resulting advantage function is negligible. Therefore, we conclude that the proposed cross-domain authentication protocol BAMA achieves semantic security of the session key under the RoR model and is secure.

B. Informal Security Analysis

1.Mutual Identity Authentication: In BAMA, there are two phases: pre-authentication and cross-domain authentication.

During the pre-authentication phase, the mobile node FID_{MD_i} completes authentication with the local fog node FID_{fog_j}. The mobile node FID_{MD_i} sends the ticket $ticket_1$ and authentication information $(K_{MD_i}, m_2, m_3, TS_4)$ obtained during the registration phase to the local fog node. The local fog node FID_{fog_j} performs the following computation:

$$\begin{aligned} IAK_{ji} &= F(FID_{fog_j}, FID_{MD_i}), n_{i,j}^{'} \\ &= h_3(IAK_{ji} \parallel ticket_1) \end{aligned}$$

And queries $n_{i,j}$ from the blockchain to sequentially verify the authenticity of $ticket_1$ and the accuracy of the authentication information, ensuring that the mobile node is a legitimate node to complete one-way authentication. Subsequently, it generates the authentication information $(m_5, K_{fog_j}, m_6, TS_5)$ and the ticket $ticket_2$, which are sent back to the mobile node. The ticket $ticket_2$ is an important credential for the mobile node to complete cross-domain authentication. The mobile node verifies whether $m_6 \cdot g = h_3(K_{i-j}^{'} \parallel IAK_{ij} \parallel K_{fog_j} \parallel m_5 \parallel TS_5) \cdot K_{fog_j}$ holds. If it holds, mutual authentication within the domain is completed.

During the cross-domain authentication phase, the mobile node sends the ticket $ticket_2$ and authentication information to FID_{fog_k}. FID_{fog_k} verifies the authenticity of the ticket and the accuracy of the information by computing and querying related information on the blockchain. This ensures that the mobile node has obtained authorization for cross-domain access within its local domain and that the information has not been tampered with during transmission. After completing one-way authentication, FID_{fog_k} generates some authentication information and a session key and returns them to the mobile node FID_{MD_i}. The mobile node then verifies the authentication information, computes the session key, and verifies the accuracy of the session key, thereby completing cross-domain mutual authentication.

Therefore, BAMA achieves both intra-domain mutual authentication and cross-domain mutual authentication. In BAMA, the authentication keys: $IAK_{ij} = F(FID_{MD_i}, FID_{fog_j}), IAK_{jk} = F(FID_{fog_j}, FID_{fog_k})$ are critical parts of mutual authentication. They are based on a binary t-degree polynomial: $F(x, y) = (\sum_{m,n=0}^{t} a_{m,n} x^m y^n) \bmod p$. Its security is based on the number of data points required to reconstruct a binary t-degree polynomial and the randomness of the coefficients. It requires $t+1$ values to reconstruct the t-degree polynomial. Since the number of nodes within the domain is only t, even if an attacker captures all the nodes in the domain, they still cannot reconstruct the polynomial.

2.Resistance to Malicious Fog Nodes: To prevent some malicious fog nodes FID_{fog} from infiltrating the system, BAMA proposes a trust management mechanism to address this issue. The trust score of each fog node is updated based on its historical trust score T_{fog}^{old}, the recommendation values S_m from all mobile nodes within the domain, and the recommendation values A_m from nearby fog nodes. The updated trust scores are then stored on the blockchain. A threshold value thr is set, and when a fog node's trust score T_{fog} is lower than the threshold, the TC designates the fog node as malicious, disallowing mobile nodes from interacting with it. Additionally, before connecting to a fog node or performing cross-domain authentication access, mobile nodes can query the fog node's trust score on the blockchain to decide whether to connect to or access the fog node.

3.Resistance to Man-in-the-Middle Attacks: Suppose an attacker attempts to impersonate a mobile node MD or tamper with the message sent to the fog node $(ticket_2, \varphi_1, s_1, TS_7)$ during the cross-domain authentication phase. In that case, they need to obtain the correct $ticket_2$. However, the ticket is signed by the Trusted Center's private key, making it impossible for the attacker to forge the ticket. Additionally, since $\varphi_1 = h_3(K_c \parallel IAK_{jk} \parallel FID_{fog_k} \parallel FID_{MD_i} \parallel s_1 \parallel TS_7)$, the attacker must know the authentication key IAK_{jk} or IAK_{kj} to generate a legitimate φ_1. Since the authentication key IAK_{jk} or IAK_{kj} is obtained through a binary t-degree symmetric polynomial, BAMA can effectively resist man-in-the-middle attacks.

4.Session Key Protocol: In BAMA, when the mobile node FID_{MD_i} and the fog node FID_{fog_k} complete mutual authentication, a session key is generated simultaneously: $SK_{ki} = h_3(K_d \parallel K_c^{'} \parallel K_e \parallel IAK_{kj}) = h_3(K_d^{'} \parallel K_c \parallel K_e^{'} \parallel IAK_{jk}) = SK_{ik}$, Additionally, $K_c = k_c \cdot g$, $K_d = k_d \cdot g$, $K_e = h_3(K_d \parallel IAK_{jk} \parallel K_c^{'})$. The session key is primarily determined by two random numbers k_c and k_d. Cracking the session key would require the attacker to solve the Elliptic Curve Computational Diffie-Hellman Problem (ECCDHP) in polynomial time, which is impossible. Moreover, the session key includes the authentication key as a parameter. Since the authentication key IAK_{kj} is obtained through a binary t-degree symmetric polynomial, the attacker cannot reconstruct the polynomial. Therefore, BAMA ensures the security of the session key.

5.Anonymity: In the BAMA scheme, attackers cannot obtain the real identities of nodes from the pseudonyms FID of fog nodes and mobile nodes. Taking the mobile device MD_i as an example, the pseudonym: $FID_{MD_i} = h_1(\tau_{MD_i} \parallel ID_{MD_i} \parallel TS_2) \cdot sk_{TC}$ is generated by the Trusted Center by introducing a random number τ_{MD_i} to increase unpredictability and using the Trusted Center's private key sk_{TC} to ensure the security of pseudonym generation and verification. h_1 is an encryption hash function whose output is unpredictable and unique. This ensures that the input random number τ_{MD_i}, real identity ID_{MD_i}, and timestamp TS_2 are irreversible after hashing. Since the TC is completely trustworthy, attackers cannot obtain the real identities of mobile nodes. Therefore, the BAMA scheme

ensures the anonymity of the nodes.

6.Traceability: Since the TC packages and stores the identity information of nodes on the blockchain after completing their registration, if it becomes necessary to trace a node due to certain events, the TC can retrieve the real identity information of the node from the blockchain based on the pseudonym. Therefore, BAMA achieves the traceability of nodes.

7.Forward/Backward Security: In BAMA, session keys are generated using random numbers k_c and k_d, which have strong freshness properties, ensuring the independence between session keys. The secure encryption hash function h_3 is used to ensure that the session key generation process is irreversible. Therefore, even if an attacker manages to obtain the key of a particular session, they cannot infer the keys of previous or subsequent sessions. Thus, the session keys in BAMA possess forward and backward security.

8.Resistance to Replay Attacks: In the BAMA scheme, timestamps (TS) and random numbers are used during both the registration and authentication phases. Therefore, BAMA can resist replay attacks.

9.Scalability: The BAMA scheme enables the simultaneous authentication and interaction of a large number of mobile devices with fog nodes in other domains. It also supports the dynamic addition of new devices and the removal of malfunctioning devices, ensuring the system's flexibility and scalability.

V. Performance Evaluation

In this subsection, the proposed BAMA scheme will be compared with four existing representative schemes in the IoT environment in terms of security features and efficiency. The efficiency comparison will focus on computational overhead and communication overhead. All the simulation experiments in this subsection were conducted on a computer with hardware configuration of AMD Ryzen 7 6800H processor, 16GB RAM and RTX 3060 graphics card using MATLAB 2022b software for computational overhead and communication overhead.

TABLE III
Comparison of Security Features

Feature	[9]	[18]	[20]	[28]	BAMA
Cross-domain mutual authentication	✓	✓	✓	✓	✓
Intra-domain mutual authentication	×	×	×	×	✓
Session key protocol	✓	✓	✓	✓	✓
Forward/Backward security	✓	✓	✓	✓	✓
Resistance to man-in-the-middle attacks	×	✓	✓	✓	✓
Resistance to replay attacks	✓	×	✓	×	✓
Resistance to malicious authentication servers	×	×	×	×	✓
Anonymity	✓	✓	✓	✓	✓
Traceability	×	✓	×	×	✓

A. Security Feature Evaluation

In Table 3, we compare the security features of the proposed BAMA scheme with those of four other schemes. As shown in the table, BAMA offers more comprehensive security features than the other schemes. Specifically, BAMA excels in resisting malicious authentication servers (domain administrators), which is crucial for preventing insider attacks. Moreover, none of the other four schemes have implemented mutual authentication within the domain. Compared to [9], [20], and [28], BAMA meets the requirement for node traceability and additionally provides protection against replay attacks, which is a common security threat, unlike [18] and [28]. Furthermore, in BAMA, node identity information, registration information, and some authentication information are stored on the blockchain, ensuring the immutability of the data.

B. Computational Overhead Evaluation

For convenience, we define some basic cryptographic operation symbols and provide their corresponding execution

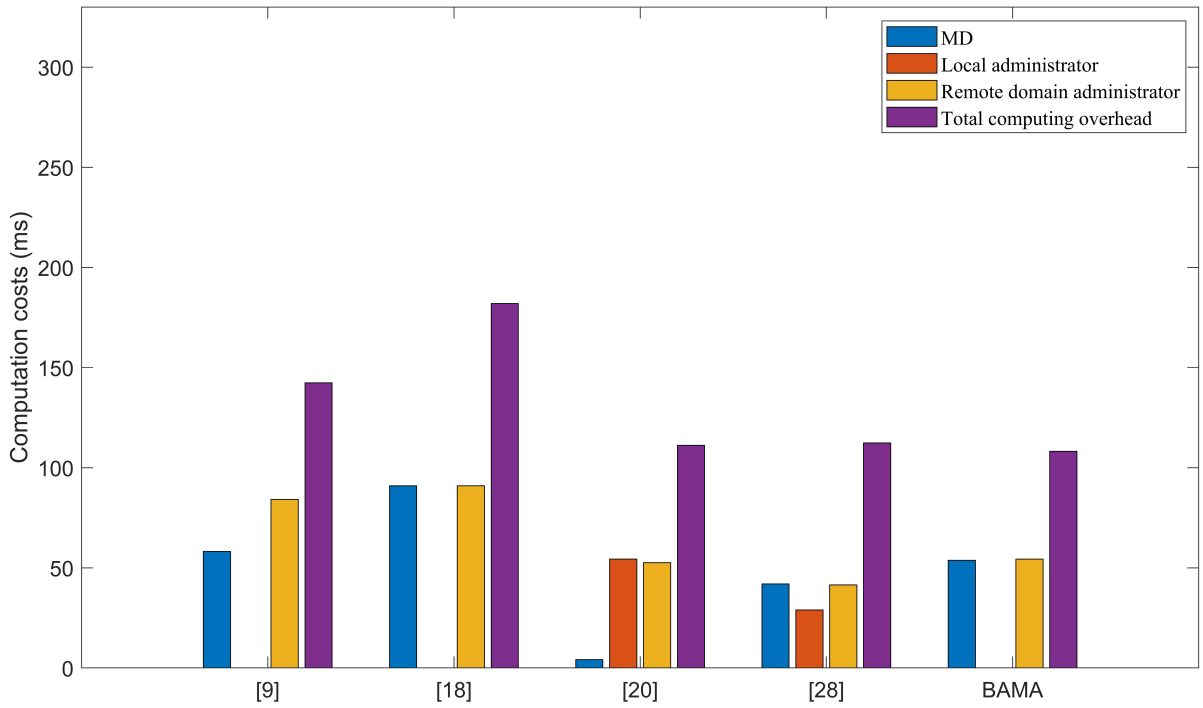

Fig. 7. Comparison of computational overhead.

TABLE IV
EXECUTION TIME OF BASIC OPERATIONS (MS)

Symbol	Description	Execution Time
T_{ecm}	Execution time of elliptic curve point multiplication	13
T_{eca}	Execution time of elliptic curve point addition	5
T_{sed}	Execution time of symmetric encryption decryption	6
T_{pair}	Execution time of bilinear pairing	25
T_h	Execution time of hash operation	0.6
T_{exp}	Execution time of modular exponentiation	16
T_{pkd}	Execution time of public key encryption	43
T_{ske}	Execution time of private key decryption	9
T_{g-sig}	Execution time of identity-based signature generation	57
T_{u-sig}	Execution time of identity-based signature verification	7

TABLE V
COMPUTATIONAL OVERHEAD COMPARISON(MS)

Scheme	Mobile Device	Local Domain Administrator	Remote Domain Administrator	Total Computational Cost
[9]	$4T_{ecm} + T_{eca} + 2T_h$	-	$6T_{ecm} + T_eca + 2T_h$	$10T_{ecm} + 2T_{eca} + 4T_h$
[18]	$5T_h + 6T_{ecm} + 2T_{eca}$	-	$5T_h + 6T_{ecm} + 2T_{eca}$	$10T_h + 12T_{ecm} + 4T_{eca}$
[20]	$7T_h$	$4T_h + T_pkd + T_ske$	$T_h + T_{pkd} + T_{ske}$	$12T_h + 2T_{pkd} + 2T_{ske}$
[28]	$5T_h + 3T_{ecm}$	$5T_h + 2T_ecm$	$4T_h + 3T_{ecm}$	$14T_h + 8T_{ecm}$
BAMA	$3T_h + 4T_{ecm}$	-	$4T_h + 4T_{ecm}$	$7T_h + 8T_{ecm}$

TABLE VI
COMMUNICATION OVERHEAD COMPARISON(BITS)

Scheme	Registration Phase	Cross-Domain Authentication Phase	Total Communication Overhead
[9]	2208	1184	3392
[18]	992	960	1952
[20]	480	3008	3488
[28]	1120	2240	3360
BAMA	1504	1024	2528

times according to [16], as shown in Table 4. Table 5 summarizes the computational overhead comparison between the proposed BAMA scheme and four other schemes, with computational entities divided into mobile nodes, local domain administrators, and remote domain administrators. As seen in Table 5, the computational cost of the BAMA scheme is the lowest. Additionally, Figure 7 compares the computational overhead of each entity in different schemes and the total computational overhead of the different schemes. From Figure 7, it can be seen that BAMA has the lowest computational overhead. This is because BAMA uses only a small number of elliptic curve point multiplications and hash operations, and therefore does not require more complex cryptographic operations such as bilinear pairing and public key encryption and decryption.

C. Communication Overhead Evaluation

We assume that the length of identity information is 160 bits, the length of random numbers is 160 bits, the length of timestamps is 32 bits, and the length of hash digests is 160 bits. The symmetric encryption and decryption algorithm is the Advanced Encryption Standard (AES) with a key length of 256 bits (maximum security). The asymmetric encryption and decryption algorithm uses the 1024-bit RSA (Rivest-Shamir-Adleman) algorithm, and the private key signature length is 1024 bits. Furthermore, we assume that the security of a 160-bit ECC (Elliptic Curve Cryptography) is equivalent to that of a 1024-bit RSA, and the prime number on the elliptic curve is 160 bits.

Table 6 summarizes the communication overhead of each scheme during the registration phase, cross-domain authentication phase, and the total communication overhead. In the registration phase, the data transmission length between the mobile node MD and the TC in the BAMA scheme is $544 + 960 = 1504$ bits. In the cross-domain authentication phase, the data transmission length is $512 + 512 = 1024$ bits. Therefore, the total communication overhead of BAMA is 2528 bits, while the total communication overheads of [9], [28], and [20] are 3392 bits, 3360 bits, and 3488 bits, respectively, all higher than that of BAMA. For better visualization, we present the results from Table 6 in Figure 8. As we can

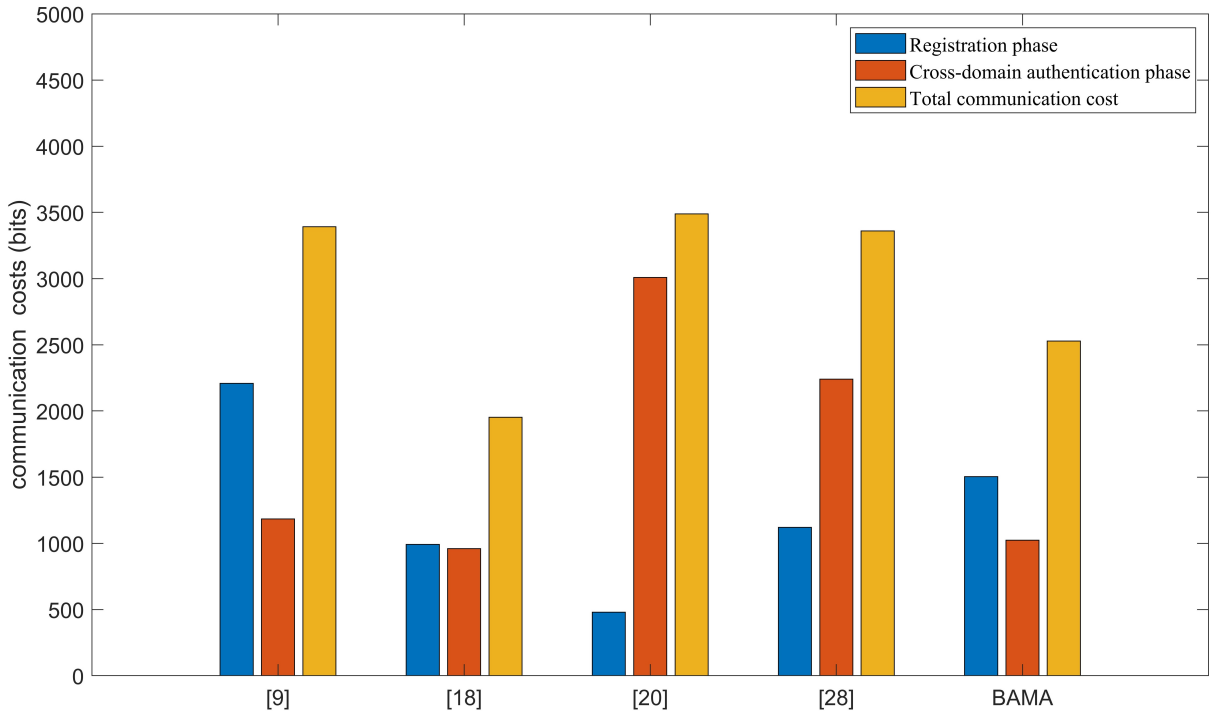

Fig. 8. Comparison of communication overhead.

see, the communication overhead of the BAMA scheme is the second lowest, slightly higher than the 1952 bits of the [18] scheme. However, BAMA provides more security features than [18]. Specifically, [18] does not offer intra-domain mutual authentication, cannot resist replay attacks, and cannot identify malicious authentication servers within the domain.

D. Blockchain Simulation Experiment

In this subsection, we conducted a simulation experiment on the blockchain application within the BAMA scheme, utilizing Hyperledger Fabric 1.4.0 as the development platform. The aim of the experiment was to store the identity information of mobile nodes and fog nodes, along with the trust scores of fog nodes, on the blockchain. The blockchain's features were leveraged to enable identity information querying and verification. To validate the feasibility of this scheme, we performed tests using drones as mobile nodes.

Firstly, during the registration phase, the fog node Fog_j sends a registration request to the TC. The TC queries the blockchain to check whether identity information for Fog_j already exists. If no corresponding record is found, it confirms that the node has not yet been registered, as shown in Figure 9. Subsequently, the TC generates a pseudonym FID_{fog_j} and a unique binary symmetric polynomial $F(FID_{fog_j}, y)$ for Fog_j, while also assigning an initial trust score T_{fog_j}. This trust score will be periodically updated and re-evaluated in the future. Finally, the pseudonym FID_{fog_j}, the binary symmetric polynomial $F(FID_{fog_j}, y)$, and the trust score T_{fog_j} are stored on the blockchain, as illustrated in Figure 10. Additionally, during the registration phase, the mobile node MD_i sends a registration request to the TC, which includes a timestamp and its real identity ID_{MD_i}. The ID_{MD_i} can be the product identification code of the drone, serving as its unique identifier, which typically contains the manufacturer's name code, product model code, and serial number. After receiving the request, the TC generates a pseudonym FID_{MD_i} for ID_{MD_i}. Subsequently, MD_i can query the identity information of the fog node FID_{fog_k} it wishes to communicate with on the blockchain and verify whether the trust score of the fog node exceeds the threshold, as shown in Figure 11. Finally, the TC computes the authentication information $n_{i,k}$ between the drone MD_i and the fog node FID_{fog_k}. The $n_{i,k}$, along with FID_{MD_i} and

```
root@liu-machine:~/go/src/github.com/hyperledger/fabric/scripts/fabric-samples/test-netwo
rk# peer chaincode query -C mychannel -n basic -c '{"Args":["ReadAssetByPseudonym","FID_(
fog_j)"]}'
Error: endorsement failure during query. response: status:500 message:"asset with pseudon
ym FID_(fog_j) does not exist"
root@liu-machine:~/go/src/github.com/hyperledger/fabric/scripts/fabric-samples/test-netwo
rk#root@liu-machine:~/go/src/github.com/hyperledger/fabric/scripts/fabric-samples/test-ne
```

Fig. 9. Failed to find the identity information for FID_{fog_j}.

```
root@liu-machine:~/go/src/github.com/hyperledger/fabric/scripts/fabric-samples/test-netwo
rk# peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.
com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/ordere
r.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" \
-C mychannel -n basic --peerAddresses localhost:7051 --tlsRootCertFiles "${PWD}/organizat
ions/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" \
--peerAddresses localhost:9051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations
/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" \
-c '{"function":"CreateAsset","Args":["Fog node_j", "FID_(fog_j)", "F(FID_(fog_j),y)", "0
", "TS_3"]}'
2024-10-13 17:25:56.345 CST 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode
invoke successful. result: status:200
```

Fig. 10. Storing FID_{fog_j} to the Blockchain.

```
root@liu-machine:~/go/src/github.com/hyperledger/fabric/scripts/fabric-samples/test-netwo
rk# peer chaincode query -C mychannel -n basic -c '{"Args":["ReadAssetByPseudonym","FID_(
fog_k)"]}'
{"Identity":"Fog node_k","Pseudonym":"FID_(fog_k)","SymmetricPolynomial":"F(FID_(fog_k),y
)","TrustScore":78,"Timestamp":"TS_4"}
root@liu-machine:~/go/src/github.com/hyperledger/fabric/scripts/fabric-samples/test-netwo
rk#
```

Fig. 11. Successfully queried the identity information for FID_{fog_k}.

```
root@liu-machine:~/go/src/github.com/hyperledger/fabric/scripts/fabric-samples/test-netwo
rk# peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.
com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/ordere
r.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" \
-C mychannel -n basic --peerAddresses localhost:7051 --tlsRootCertFiles "${PWD}/organizat
ions/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" \
--peerAddresses localhost:9051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations
/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" \
-c '{"function":"CreateAsset","Args":["Mobile node_i", "FID_(MD_i)", "n_(i,k)", "FID_(fog
_k)", "TS_5"]}'
2024-10-13 18:18:42.613 CST 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode
invoke successful. result: status:200
```

Fig. 12. Storing FID_{MD_i} to the Blockchain.

```
root@liu-machine:~/go/src/github.com/hyperledger/fabric/scripts/fabric-samples/test-netwo
rk# peer chaincode query -C mychannel -n basic -c '{"Args":["ReadAssetByPseudonym","FID_(
MD_i)"]}'
{"Identity":"Mobile node_i","Pseudonym":"FID_(MD_i)","AuthenticationInfo":"n_(i,k)","FogN
odePseudonym":"FID_(fog_k)","Timestamp":"TS_5"}
```

Fig. 13. Successfully queried the authentication information for FID_{MD_i}.

```
root@liu-machine:~/go/src/github.com/hyperledger/fabric/scripts/fabric-samples/test-netwo
rk# peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.
com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/ordere
r.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" \
-C mychannel -n basic --peerAddresses localhost:7051 --tlsRootCertFiles "${PWD}/organizat
ions/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" \
--peerAddresses localhost:9051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations
/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" \
-c '{"function":"CreateAsset","Args":["Fog node_k", "FID_(fog_k)", "n_(fog_(k-h))", "FID_
(MD_i)", "TS_6"]}'
2024-10-13 18:23:23.302 CST 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode
invoke successful. result: status:200
```

Fig. 14. Store the cross-domain authentication information on the blockchain.

FID_{fog_k}, is stored on the blockchain, as shown in Figure 12.

In the pre-authentication phase, the local fog node FID_{fog_k} verifies the authenticity and legitimacy of the mobile node MD_i by retrieving the authentication information from the blockchain and performing the necessary calculations, as shown in Figure 13. After successful authentication, the drone MD_i will further query the blockchain for the identity information of the fog node it intends to access across domains, ensuring that the trust score of the fog node meets the required threshold. The query results are the same as shown in Figure 11. Subsequently, the local fog node generates authentication information $n_{fog_{k-h}}$ and stores $n_{fog_{k-h}}$, FID_{MD_i}, and FID_{fog_h} on the blockchain, as shown in Figure 14. In the cross-domain authentication phase, the remote fog node FID_{fog_h} can perform preliminary authentication with the mobile node MD_i by querying the authentication information stored on the blockchain. The query results are similar to those shown in Figure 13.

In summary, this experiment not only validated the effectiveness of blockchain technology within the BAMA scheme but also demonstrated its excellent performance in device identity management, information querying, and verification. The immutability and traceability of blockchain provided reliable security guarantees for the entire system, particularly in dynamic and complex network environments. By utilizing blockchain to store identity information and trust scores, the system can better prevent malicious attacks and identity spoofing issues, thereby enhancing the overall security and credibility of the system.

VI. Conclusion

In this paper, we propose a secure cross-domain authentication scheme named BAMA to address the cross-domain access requirements in mobile IoT. Firstly, BAMA not only achieves intra-domain mutual authentication but also realizes mutual authentication between mobile nodes and other domain administrators, generating session keys during the cross-domain authentication process. BAMA replaces traditional bilinear pairing with symmetric polynomials, which simplifies the computation process and improves system efficiency. Additionally, by storing nodes' registration information and some authentication information on the blockchain, BAMA enhances the system's traceability and security.

BAMA also introduces a trust management method specifically for evaluating and managing the trustworthiness of fog nodes. This method involves all fog nodes in the network performing calculations and updates, reaching consensus through the PBFT consensus algorithm, and storing the latest trust scores of the fog nodes in the blockchain. Through this approach, mobile nodes can query the trust scores of fog nodes on the blockchain before connecting to them, ensuring the trustworthiness of each domain administrator (fog node). Finally, through security and performance analysis, BAMA provides more comprehensive security features and better performance.

In the future, the scheme could be extended to broader application domains to comprehensively meet the complex demands of the Internet of Things (IoT). Additionally, attention could be given to the development of emerging technologies, such as quantum secure communication, to further enhance the security of the scheme.

References

[1] Nguyen D C, Ding M, Pathirana P N, et al. 6G Internet of Things: A comprehensive survey[J]. IEEE Internet of Things Journal, 2021, 9(1): 359-383.

[2] Haque A K M B, Bhushan B, Dhiman G. Conceptualizing smart city applications: Requirements, architecture, security issues, and emerging trends[J]. Expert Systems, 2022, 39(5): e12753.

[3] Stoyanova M, Nikoloudakis Y, Panagiotakis S, et al. A survey on the internet of things (IoT) forensics: challenges, approaches, and open issues[J]. IEEE Communications Surveys & Tutorials, 2020, 22(2): 1191-1221.

[4] Ismagilova E, Hughes L, Rana N P, et al. Security, privacy and risks within smart cities: Literature review and development of a smart city interaction framework[J]. Information Systems Frontiers, 2022: 1-22.

[5] Iorga M, Feldman L, Barton R, et al. Fog computing conceptual model[J]. 2018.

[6] Gong B, Zheng G, Waqas M, et al. LCDMA: Lightweight cross-domain mutual identity authentication scheme for Internet of Things[J]. IEEE Internet of Things Journal, 2023, 10(14): 12590-12602.

[7] Wang F, Wang J, Yang W. Efficient incremental authentication for the updated data in fog computing[J]. Future Generation Computer Systems, 2021, 114: 130-137.

[8] Ali Z, Chaudhry S A, Mahmood K, et al. A clogging resistant secure authentication scheme for fog computing services[J]. Computer Networks, 2021, 185: 107731.
[9] Wang W, Huang H, Zhang L, et al. Secure and efficient mutual authentication protocol for smart grid under blockchain[J]. Peer-to-Peer Networking and Applications, 2021, 14(5): 2681-2693.
[10] Shuai M, Yu N, Wang H, et al. Anonymous authentication scheme for smart home environment with provable security[J]. Computers & Security, 2019, 86: 132-146.
[11] Zhang H, Chen X, Lan X, et al. BTCAS: A blockchain-based thoroughly cross-domain authentication scheme[J]. Journal of Information Security and Applications, 2020, 55: 102538.
[12] Ding J, Ke P, Lin C, et al. Bivariate polynomial-based secret sharing schemes with secure secret reconstruction[J]. Information Sciences, 2022, 593: 398-414.
[13] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system[J]. 2008.
[14] Dai H N, Zheng Z, Zhang Y. Blockchain for Internet of Things: A survey[J]. IEEE internet of things journal, 2019, 6(5): 8076-8094.
[15] Zhaofeng M, Lingyun W, Xiaochang W, et al. Blockchain-enabled decentralized trust management and secure usage control of IoT big data[J]. IEEE Internet of Things Journal, 2019, 7(5): 4000-4015.
[16] Lin Y, Wang X, Gan Q, et al. A secure cross-domain authentication scheme with perfect forward security and complete anonymity in fog computing[J]. Journal of Information Security and Applications, 2021, 63: 103022.
[17] Guo Y, Guo Y. FogHA: An efficient handover authentication for mobile devices in fog computing[J]. Computers & Security, 2021, 108: 102358.
[18] He D, Kumar N, Wang H, et al. A provably-secure cross-domain handshake scheme with symptoms-matching for mobile healthcare social network[J]. IEEE Transactions on Dependable and Secure Computing, 2016, 15(4): 633-645.
[19] Zhou Y W, Yang B. Provable secure authentication protocol with direct anonymity for mobile nodes roaming service in Internet of things[J]. Journal of Software, 2015, 26(9): 2436-2450.
[20] Shashidhara R, Lajuvanthi M, Akhila S. A secure and privacy-preserving mutual authentication system for global roaming in mobile networks[J]. Arabian Journal for Science and Engineering, 2022, 47(2): 1435-1446.
[21] Meng X, Zhang L, Kang B. Fast secure and anonymous key agreement against bad randomness for cloud computing[J]. IEEE Transactions on Cloud Computing, 2020, 10(3): 1819-1830.
[22] Jegadeesan S, Azees M, Kumar P M, et al. An efficient anonymous mutual authentication technique for providing secure communication in mobile cloud computing for smart city applications[J]. Sustainable Cities and Society, 2019, 49: 101522.
[23] Ali R, Pal A K, Kumari S, et al. A secure user authentication and key-agreement scheme using wireless sensor networks for agriculture monitoring[J]. Future Generation Computer Systems, 2018, 84: 200-215.
[24] Shen M, Liu H, Zhu L, et al. Blockchain-assisted secure device authentication for cross-domain industrial IoT[J]. IEEE Journal on Selected Areas in Communications, 2020, 38(5): 942-954.
[25] Feng C, Liu B, Guo Z, et al. Blockchain-based cross-domain authentication for intelligent 5G-enabled internet of drones[J]. IEEE Internet of Things Journal, 2021, 9(8): 6224-6238.
[26] Ali G, Ahmad N, Cao Y, et al. xDBAuth: Blockchain based cross domain authentication and authorization framework for Internet of Things[J]. IEEE access, 2020, 8: 58800-58816.
[27] Zhang S, Lee J H. A group signature and authentication scheme for blockchain-based mobile-edge computing[J]. IEEE Internet of Things Journal, 2019, 7(5): 4557-4565.
[28] Roy P K, Bhattacharya A. Secure and authentic anonymous roaming service[J]. Wireless Personal Communications, 2022, 125(1): 819-839.
[29] Abdalla M, Fouque P A, Pointcheval D. Password-based authenticated key exchange in the three-party setting[C]//International workshop on public key cryptography. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005: 65-84.
[30] Dolev D, Yao A. On the security of public key protocols[J]. IEEE Transactions on information theory, 1983, 29(2): 198-208.