Journal of King Saud University - Computer and Information Sciences 36 (2024) 101946

HOSTED BY

Contents lists available at ScienceDirect

Journal of King Saud University - Computer and Information Sciences

journal homepage: www.sciencedirect.com

Hybrid blockchain-based many-to-many cross-domain authentication scheme for smart agriculture IoT networks

Fengting Luo, Ruwei Huang *, Yuqi Xie

School of Computer and Electronic Information, Guangxi University, Nanning 530004, China

ARTICLE INFO

Keywords:
Cross-domain authentication
Blockchain
Batch verification
Smart agriculture IoT networks
Certificateless cryptography (CLC)

ABSTRACT

With the development of smart agricultural Internet of Things (IoT) projects, the need for extensive collaboration among agricultural devices from different domains has surged, necessitating the authentication of device identities for secure communication. Existing centralized architecture-dependent authentication mechanisms encounter issues like a sole point of failure and inefficiencies. Most authentication schemes are unable to support plentiful devices synchronously connecting to numerous data servers in other domains. To address these problems, we propose a many-to-many cross-domain authentication scheme based on the hybrid blockchain architecture for smart agriculture IoT networks. The scheme enables multiple devices simultaneously executing mutual authentication with several data service providers from other agricultural systems. This paper designs a groupable batch verification (GBV) algorithm that dynamically adjusts the batch size by performing group verification for a list of requests, enhancing the flexibility of cross-domain batch authentication. Furthermore, the proposed scheme provides a pseudonym update mechanism to protect the privacy of devices and guards services of different domains from illegal access by tracking malicious devices. The security analysis and performance evaluation demonstrate that the proposed scheme has superior security features and performance.

1. Introduction

The sustainability of humans depends mainly on the prosperity of food and agriculture. The intelligent agricultural IoT project has excellent potential to improve agricultural productivity (Khanal et al., 2017). Multiple servers in the agricultural IoT system provide smart devices with real-time data on soil, crops, weather conditions, and other related services (Dos Santos et al., 2019; Torky and Hassanein, 2020). Cross-domain cooperation between different farms is beneficial to integrate resources to improve agricultural production efficiency. Devices are more likely to span several cooperative farms. However, current IoT systems are highly vulnerable to multiple cyber-attacks by unauthorized parties (Makhdoom et al., 2018). An effective cross-domain authentication mechanism is required to ensure the safe cross-domain connection of devices.

Most current applications select Public Key Infrastructure (PKI) (Housley et al., 1999) to design cross-domain authentication solutions. PKI utilizes the third-party certificate authority (CA) to assign each device a digital certificate. Its disadvantages are the high cost of certificate management and complicated certification paths. Some systems pick inter-domain identity verification approachs designed on Identity Based Cryptography (IBC) (Shamir, 1985), which take the device's valid identity as the public key. A Private Key Generator (PKG) performs the distribution of private keys, which is easy to cause abuses of power. The certificateless cryptography (CLC) mechanism (Liu et al., 2020) is proposed, eliminating the disadvantages of PKI and IBC systems. Therefore, it is worth exploring further to design a solution for authentication across agricultural IoT systems based on CLC.

Most existing authentication schemes (Ali et al., 2018; Chen et al., 2019; Alyahya et al., 2022; Bothe et al., 2019; Rangwani et al., 2021) for smart agriculture adopt centralized network topologies, facing the threat of single point of failure. Luckily, blockchain technology offers a fresh approach to addressing the security issues of IoT (Ferrag et al., 2018). Blockchain is a distributed ledger used to record transactions with timestamps in a peer-to-peer network. Blockchain technology enables users to independently audit and authenticate transactions, making it suitable for trusted data access and recording in distributed environments (Huang et al., 2020).

* Corresponding author.
E-mail address: ruweih@gxu.edu.cn (R. Huang).

https://doi.org/10.1016/j.jksuci.2024.101946
Received 30 August 2023; Received in revised form 24 January 2024; Accepted 25 January 2024
Available online 5 February 2024

1.1. Motivation

In recent years, many scholars (Shen et al., 2020; Wang et al., 2021; Vangala et al., 2022; Yang et al., 2021; Bagga et al., 2021; Xue et al., 2022) have applied blockchain technology to IoT cross-domain identity authentication. Blockchain in these documents is primarily utilized for storing device certificates and sensor data. Agricultural cooperative alliances typically consist of a series of agricultural domains, each hosting IoT smart devices. This means that private data is only accessible to internal personnel, while other data remains unencrypted and open to the public. Hybrid blockchains are considered more suitable for agricultural applications (Vangala et al., 2021; Vangala et al., 2022). In certain sensitive cross-domain collaboration scenarios within smart agriculture, a hybrid blockchain ensures data confidentiality through the implementation of a private chain. Concurrently, utilizing a consortium blockchain provides transparency for agricultural institutions in different domains, ensuring effective public verification among these institutions.

Existing solutions only support cross-domain authentication between single or multiple devices and a single server. In intelligent agricultural systems, multiple devices usually simultaneously request several servers to provide services. For instance, a group of agricultural robots may concurrently seek connections to various servers, including weather information collection stations, greenhouse control systems, insect monitoring systems, and so forth.

Most current authentication schemes operate in a one-to-one or many-to-one mode, resulting in significant overhead between different entities. Applying existing protocols to multi-to-multi scenarios is impractical. To facilitate collaboration and development in multi-domain smart agriculture, it is essential to design a many-to-many cross-domain authentication scheme based on a hybrid blockchain. This approach aims to meet the security requirements of smart agriculture systems while ensuring high computational efficiency. Besides, in traditional batch verification, the batch size cannot be changed. In the case of batch verification failure, all the messages are rejected. In multi-to-multi scenarios, the computational and communication overhead of re-authentication for numerous agricultural devices is substantial. Hence, providing a multi-to-multi authentication mechanism with dynamically adjustable batch sizes can effectively address threats from malicious devices, reducing the re-authentication costs for credible agricultural equipment.

1.2. Research contributions

This article proposes a hybrid blockchain-based many-to-many cross-domain authentication scheme for smart agriculture IoT networks (we call it HBMA). This study will be explored among IoT systems using the CLC mechanism. The primary contributions of this paper are outlined below.

(1) We design a hybrid blockchain model for smart agriculture IoT networks, including a consortium blockchain and several local private blockchains. The private blockchains collaborate with the consortium blockchain to complete cross-domain identity verification for agriculture systems that have varying cryptographic configurations.
(2) We presents a protocol for cross-domain authentication and key agreement that operates in a many-to-many fashion. This protocol supports multiple devices simultaneously conducting mutual authentication with several data service providers (DSPs) from other domains. Furthermore, HBMA provides a groupable batch verification (GBV) algorithm that dynamically adjusts the batch size by performing group verification on a list of requests. When a failure takes place, GBV is initiated to perform verification again, reducing the cost of re-requesting authentication for devices.
(3) To enhance security, the mechanisms of anonymous authentication and device revocation are provided, and a protocol for pseudonym update is designed to prevent malicious attackers from continuously tracking device behavior.

1.3. Outline

The rest of this paper is structured as follows: Section 2 reviews the relevant literature. Section 3 gives an overview of the system model and associated theoretical knowledge. The design details of HBMA are shown in Section 4. Section 5 gives the security analysis of this scheme. Section 6 conductes the performance evaluation, proving that HBMA performs better compared with related solutions. In the end, Section 7 concludes this article.

2. Related work

2.1. Existing IoT-based authentication scheme

Various authentication schemes (Ali et al., 2018; Chen et al., 2019; Alyahya et al., 2022; Bothe et al., 2019; Rangwani et al., 2021) have recently been suggested to provide security in IoT environments, with the majority relying on digital certificates or usernames and passwords. Liu et al. (2008) designed a authentication solution based on the PKI mechanism that is appropriate for extensive distributed networks, which consumes heavy certificate management overhead. Yuan et al. (2017) introduced a method for achieving mutual authentication between PKI and IBC domains. But this scheme does not support domain systems with different cryptographic parameters. Li et al. (2016) designed a lightweight cross-domain authentication protocol based on the CLC mechanism, which does not need to manage certificates and escrow keys.

The above solutions manage device identity through the trusted third party, which faces potential risks of a single point of failure and security threats caused by deliberate inactions of the third party. Blockchain technology, widely used to support multi-party cooperation, provides a new solution for cross-domain identity authentication of distributed IoT (Shen et al., 2019). Shen et al. (2020) designed a blockchain-assisted authentication scheme to support secure cross-domain collaboration between devices from different domains. Wang et al. (2021) developed a mutual authentication scheme using blockchain technology for a smart grid, which can complete batch authentication of multiple devices. Vangala et al. (2022) designed a blockchain-authenticated key agreement scheme for precision agriculture IoT networks. Yang et al. (2021) proposed a blockchain-based multi-domain vehicle authentication scheme. The plans offered in Shen et al. (2020), Wang et al. (2021), Vangala et al. (2022), and Yang et al. (2021) all face threats of key escrow.

Bagga et al. (2021) proposed a blockchain-based batch authentication scheme without key escrow, which allows a group of vehicles to be verified by the RSU. Xue et al. (2022) designed a cross-domain authentication scheme using two collaborative blockchains for CLC systems that employ various cryptographic settings. However, few studies focus on the application scenario where multiple devices request multiple servers simultaneously. Zhang et al. (2020) proposed a scheme ensuring secure authentication between numerous vehicles and several cloud service providers. Nevertheless, this solution still uses a centralized authentication architecture, which cannot support many-to-many cross-domain authentication.

2.2. Authentication scheme in smart agriculture networks

Ali et al. (2018) designed a remote user authentication scheme using

symmetric cryptography, which is suitable for application in agriculture wireless sensor network monitoring systems. Chae and Cho (2018) developed an authentication scheme to ensure the security of control devices in smart farms, and this scheme enhances authentication efficiency by minimizing encryption and decryption operations. Rangwani et al. (2021) devised a lightweight mutual authentication protocol for secure communication in agricultural IoT networks, protecting privacy through the use of elliptic curve cryptography (ECC) and hash functions. These authentication schemes depend on a central server. However, in the event of a compromise to the central server, the entire smart agriculture system becomes vulnerable to failure, potentially resulting in substantial economic losses for the agriculture sector. Moreover, these centralized authentication schemes cannot be applied to cross-domain collaboration among different smart farms.

Wu and Tsai (2019) developed a smart agriculture identity authentication mechanism based on a private blockchain, eliminating the threat of a single point of failure and aiding in the integrity and source verification of authenticated messages. However, this scheme is susceptible to offline guessing attacks and incurs high costs due to the use of bilinear pairings. Bera et al. (2022) utilized blockchain technology to design an authentication and key management scheme for IoT system in precision agriculture. Vangala et al. (2020) conducted a systematic survey on the application of blockchain in smart agriculture. They also proposed a general architecture based on blockchain that is well-suited for intelligent agriculture environments. Subsequently, Vangala et al. (2021) introduced a new identity authentication scheme for multi-party collaboration in smart agriculture using a hybrid blockchain and edge computing. However, none of the aforementioned works are suitable for smart agriculture tasks where multiple devices simultaneously request service from several data service providers.

In the agricultural domain, each agricultural production enterprise possesses private data and specific security settings. Additionally, certain data must be shared with specific stakeholders or made publicly accessible to ensure transparency and facilitate cross-domain collaboration. Thus, hybrid blockchains are seen as more fitting for use in agriculture. It remains crucial to design an effective many-to-many cross-domain authentication scheme based on a hybrid blockchain for smart agricultural IoT systems with different cryptographic settings.

3. Preliminaries

The symbol definitions involved in this study are shown in Table 1.

3.1. System model

The hybrid blockchain system model designed for distributed smart agriculture IoT networks is depicted in Fig. 1, comprising the device layer, the management layer, and the trust layer.

The device layer is formed by smart agricultural devices (SD_i^N), including cultivators, seeders, and agricultural robots. These intelligent devices are highly mobile.

The management layer consists of a local private blockchain for each domain. Every private chain is jointly maintained by a trusted authority (TA^N), the service manager (SM_k^N) and the data service providers (DSP_j^N). SM_k^N, as a server possessing robust computational and storage capabilities, undertakes device identity verification and supervision. DSP_j^N usually includes meteorological information stations, soil moisture information stations, insect monitoring systems, etc., which are responsible for providing data services. In addition, trusted data service providers with mighty computing power can also act as service managers.

The trust layer is composed of a consortium blockchain collectively managed by TA^N of every domain and the law enforcement agency (LEA). TA^N is responsible for system initialization, entity registration, and device management. LEA is typically an authoritative agricultural regulatory agency responsible for revoking the identity of malicious devices. Agricultural service managers need to access the consortium blockchain to verify the identity of cross-domain agricultural devices. Therefore, a blockchain gateway server (BGS) is deployed to provide access services for SM_k^N, enabling the acquisition of shared agricultural data among collaborating farms. SM_k^N can query the CBC through the BGS to obtain specific identity information of cross-domain devices, excluding the real identity of devices. On the one hand, SM_k^N does not need to allocate additional storage space to synchronize agricultural device information from other farms. On the other hand, this contributes to enhancing the efficiency of the negotiation and consensus process.

Table 1
Symbols used in this study and their definitions.

Symbol	Definition
LEA	Law enforcement authority
TA^N	Trusted authority of *Domain N*
SM_k^N	Service manager marked with k of *Domain N*
DSP_j^N	Data server provider marked with j in *Domain N*
$\left[DSP_j^N\right]$	A cluster of DSP_j^N, where $j = 1, 2, \cdots, m$
SD_i^N	Agricultural smart device marked with i in *Domain N*
$[SD_i^N]$	A cluster of SD_i^N, where $i = 1, 2, \cdots, n$
LBC^N	Local private blockchain of *Domain X*
CBC	Consortium blockchain
RID_i^N	Real identity of the entity i in *Domain N*
PID_i^N	Pseudonym identity of the entity i in *Domain N*
$h_i(msg)$	Hash function

3.2. Implementation of the system model in smart agriculture

The hybrid blockchain model can address various complex scenarios in smart agriculture, and an illustrative example is provided below.

A farm is equipped with various agricultural data service platforms, including meteorological monitoring systems, soil detection systems, insect monitoring systems, and various types of mobile agricultural equipment, such as plows, seeders, and agricultural robots. The facilities are likely to be sourced from different service providers. Implementing a local private blockchain on the farm enables secure and efficient communication among devices and various data service systems. Many crops are significantly affected by seasons and weather conditions, necessitating production and harvesting within specific time frames. For example, staple crops like rice, wheat, and corn are typically planted and harvested in specific seasons. Missing the appropriate time window may lead to reduced yields or an inability to meet market demands. Therefore, the urgency of agricultural production underscores the importance of timely effective management and collaborative efforts to ensure the quality, timeliness, and market competitiveness of agricultural products.

In a region, there are typical numerous farms. Establishing a farm cooperative alliance can enhance the utilization rate of equipment resources across farms, ensuring efficiency and quality in agricultural production. To ensure trustworthy cross-domain collaboration between farms, a consortium blockchain can be implemented. The information about shareable agricultural devices is uploaded to the consortium blockchain, ensuring transparency and resistance to tampering. Additionally, introducing the agricultural regulatory authority in the region into the consortium blockchain as a law enforcement agency node is crucial. It would be responsible for overseeing malicious devices, thereby reinforcing the security and fairness of farm cooperation.

3.3. Cryptosystem and hash function

3.3.1. Elliptic curve cryptosystem

F_p is a finite field defined by p, where p is a huge prime. $E(F_p) : y^2 = x^3 + ax + b \bmod p$ is an elliptic curve over F_p, where $a, b \in F_p$

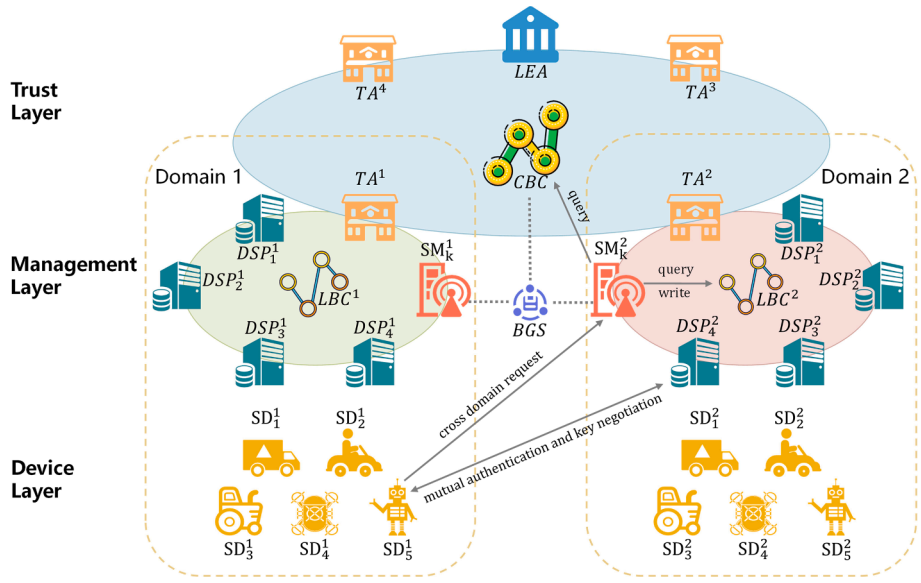

Fig. 1. System model.

and the order of $E(F_p)$ is the big prime q. Suppose P is the generator of $E(F_p)$ and $\mathbb{G}$ is an additive group composed of the points on $E(F_p)$. Given two points P and Q of $\mathbb{G}$, $R = P + Q$ can be obtained. The result of scalar multiplication $k \bullet P$ also can be got, where $k \in Z_q^*$. Additionally, the hardness assumptions for $\mathbb{G}$ are as follows (Menezes and Vanstone, 1993).

(1) Elliptic curve discrete logarithm problem (ECDLP): Let $Q = x \bullet P$, where $P, Q \in \mathbb{G}$. Any probabilistic polynomial time (PPT) adversary $\mathscr{A}$ cannot obtain $x \in Z_q^*$.
(2) Elliptic curve computational Diffie-Hellman (ECCDH) assumption: Let $X = x \bullet P$, $Y = y \bullet P$, where $X, Y, P \in \mathbb{G}$. It is impossible for any PPT adversary $\mathscr{A}$ to calculate $x \bullet y \bullet P$.

3.3.2. Hash function

A hash function consists of a pair of algorithms $Hash = (Gen, H)$ that run in polynomial time (Katz and Lindell, 2014). Let the security parameter 1^λ be the input, algorithm Gen outputs a key $s(\lambda)$. There is a polynomial $\mathscr{L}$ in such a way that for any string $x \in \{0,1\}^*$ and a key $s(\lambda)$, the function H outputs $H^s(x) \in \{0,1\}^{\mathscr{L}(\lambda)}$. A hash function has the following properties.

(1) Easy to compute: Let the input of H be $x \in \{0,1\}^*$, it is easy to output $H^s(x)$.
(2) Hard to invert: Given that the output of H is $H^s(x) \in \{0,1\}^{\mathscr{L}(\lambda)}$, it is difficult for any PPT adversary $\mathscr{A}$ to calculate $x \in \{0,1\}^*$.
(3) Collision-resistance: Let the output of H be $H^s(x) \in \{0,1\}^{\mathscr{L}(\lambda)}$, it is difficult to find $x, x' \in \{0,1\}^*$ to make the equation $H^s(x') = H^s(x)$ hold, where $x' \neq x$.

4. The proposed scheme

This section comprises four stages: initialization, entity registration, cross-domain authentication and key agreement, in addition to pseudonym update and device revocation.

4.1. Initialization

The trusted authority of every IoT domain implements the system initialization. Concretely, TA^X of Domain X selects an additive group $\mathbb{G}$ over an elliptic curve $E(F_{p_x})$ whose generator is P_x and order is q_x. TA^X chooses hash functions $h_0^X : \{0,1\}^* \times \{0,1\}^* \to Z_{q_x}^*$, $h_1^X : \{0,1\}^* \times G^2 \to Z_{q_x}^*$, $h_2^X : \{0,1\}^* \times \{0,1\}^* \times G^2 \to Z_{q_x}^*$, $h_3^X : G^2 \times \{0,1\}^n \to Z_{q_x}^*$, $h_4^X : \{0,1\}^* \times G \times \{0,1\}^n \to Z_{q_x}^*$. Then, TA^X sets a master key pair (s_y, Pub_y), where $s_x \in Z_q^*$ and $Pub_x = s_x * P_x$. TA^X keeps s_x secretly, and publishes the public parameters $\left\{\mathbb{G}, P_x, q_x, Pub_x, h_0^X, h_1^X, h_2^X, h_3^X, h_4^X\right\}$ to CBC and LBC^X.

Likewise, in Domain Y, TA^Y picks an group $\mathbb{G}$ on the elliptic curve $E\left(F_{p_y}\right)$ with the generator P_y and the order q_y. TA^Y defines (s_y, Pub_y) as its key pair and sets hash functions $h_1^Y : \{0,1\}^* \times G^2 \to Z_{q_y}^*$, $h_2^Y : G \times \{0,1\}^n \times \{0,1\}^* \to Z_{q_y}^*$, $h_3^Y : G \times \{0,1\}^n \to Z_{q_y}^*$. The public parameters are issued to CBC and LBC^Y.

4.2. Entity registration

The identity registration of each entity is performed through a dedicated secure channel, and the detailed steps are shown in the following design.

(1) **Smart device registration:** SD_i^X selects a private number $r_i \in Z_q^*$, and calculates $R_i = r_i \bullet P_x$. Then, it sends $\{RID_i^X, R_i\}$ to TA^X. TA^X generates a pseudonym $PID_i^X = h_0^X(RID_i^X \| \omega_i)$, where $\omega_i \in Z_q^*$ is a random number. Then, TA^X picks a private key $e_i \in Z_q^*$, and calculates $E_i = e_i \bullet P_x$, $\mu_i = h_1^X(PID_i^X \| R_i \| E_i)$, $\sigma_i = e_i + s_x \bullet \mu_i$. The information $\{PID_i^X, E_i, \sigma_i\}$ is sent to SD_i^X. TA^X executes the registration transaction, and the information $\{PID_i^X, RID_i, R_i, E_i, reg, sta, gid\}$ is written into CBC, and $\{PID_i, R_i, E_i, reg, sta\}$ is saved to LBC^X, where reg indicates the registration time, sta is the credential status (e.g., valid or invalid), and gid indicates the identity of the IoT domain. Receiving the information, if the equation $\sigma_i \bullet P_x = E_i + h_1^X(PID_i^X \| R_i \| E_i) \bullet Pub_x$ holds, SD_i^X retains

$\{RID_i^X, PID_i^X, r_i, \sigma_i, R_i, E_i\}$, otherwise, it ends the registration procedure.

(2) **Data service provider registration:** DSP_j^Y picks key pairs (r_j, R_j), where $r_j \in Z_{q_y}^*$, $R_j = r_j \bullet P_y$. Receiving the information $\{RID_j^Y, R_j\}$, TA^Y calculates $E_j = e_j \bullet P_y$, $\mu_j = h_1^Y(RID_j^Y \| R_j \| E_j)$, $\sigma_j = e_j + s_y \bullet \mu_j$. TA^Y stores $\{RID_j^Y, R_j, E_j, reg, sta, gid\}$ to CBC and writes $\{RID_j^Y, R_j, E_j, reg, sta\}$ into LBC^Y. DSP_j^Y checks the equation $\sigma_j \bullet P_y = E_j + h_1^Y(RID_j^Y \| R_j \| E_j) \bullet Pub_y$. If it holds, DSP_j^Y stores $\{RID_j^Y, r_j, \sigma_j, R_j, E_j\}$.

(3) **Service manager registration:** The registration process of SM_k^Y is similar to that of DSP_j^Y. The public key of SM_k^Y is $R_k = r_k \bullet P_y$, where $r_k \in Z_{q_y}^*$. TA^Y generates partial keys $\{E_k, \sigma_k\}$ for SM_k^Y, where $E_k = e_k \bullet P_y$, $\mu_k = h_1^Y(RID_k^Y \| R_k \| E_k)$, $\sigma_k = e_k + s_y \bullet \mu_k$. The identity information is written into CBC and LBC^Y. SM_k^Y stores $\{RID_k^Y, r_k, \sigma_k, R_k, E_k\}$. Furthermore, SM_k^Y sends the group key GK to $\left[DSP_j^Y\right]$ through a secure channel.

4.3. Cross-domain authentication and key agreement

Assume that a batch of devices $\left[SD_i^X\right]$ from Domain X need to request data services from $\left[DSP_j^Y\right]$ in domain Y, where $i = 1, 2, \cdots, n$, $j = 1, 2, \cdots, m$. With the help of SM_k^Y, $[SD_i^X]$ and $\left[DSP_j^Y\right]$ complete mutual authentication and establish a session key. The process is shown in Fig. 2. Step 1–Step 5 describes the specific design.

Step 1: $[SD_i^X]$ selects a private random number $b_i \in Z_q^*$ and calculates $B_i = b_i \bullet P_y$, $A_i = h_2^Y\left(B_i \| T_i \| \left[RID_j^Y\right]\right)$, $DV_i = r_i \bullet A_i + \sigma_i$. Then, $[SD_i^X]$ sends n authentication request messages $\left\{PID_1^X, B_1, DV_1, T_1, \left[RID_j^Y\right]\right\}$, $\left\{PID_2^X, B_2, DV_2, T_2, \left[RID_j^Y\right]\right\}$, …, $\left\{PID_n^X, B_n, DV_n, T_n, \left[RID_j^Y\right]\right\}$ to SM_k^Y.

Step 2: SM_k^Y checks the freshness of T_i. The requests will be rejected if $|T_i^{'} - T_i| > \Delta t$, where $T_i^{'}$ is the timestamp of receiving the message. Next, SM_k^Y checks LBC^Y to confirm whether $\left[RID_j^Y\right]$ are legitimate entities. It queries CBC to confirm the legitimacy of $[SD_i^X]$ and obtain the authentication material $\{PID_i^X, R_i, E_i, gid\}$ of the devices. Given the request list $\mathscr{PR} = \{req_i = (PID_i^X, R_i, E_i, B_i, DV_i, T_i)\}_{i=1}^n$, batch verification follows equation (1).

$$\left(\sum_{i=1}^{n} DV_i\right) \bullet P_x = \sum_{i=1}^{n} R_i \bullet h_2^Y\left(B_i \| T_i \| \left[RID_j^Y\right]\right) + \sum_{i=1}^{n} E_i + \left(\sum_{i=1}^{n} h_1^X\left(PID_i^X \| R_i \| E_i\right)\right) \bullet Pub_x \tag{1}$$

If equation (1) holds, all the signatures are valid. Otherwise, SM_k^Y calls the GBV algorithm according to the current task requirements, and redoes the batch verification. As shown in Algorithm 1, SM_k^Y picks a verification rate δ and the number a of groups for the list, where $\mathscr{M} = \lceil \delta \bullet n \rceil$ represents the required number of legal requests, and λ is the dynamically adjusted batch size. The algorithm initializes an empty legal request list $\mathscr{LR} = \varnothing$ and an empty illegal request list $\mathscr{TR} = \varnothing$, where *lsum* is the number of the legal requests added to $\mathscr{LR}$ and *fsum* is the number of the illegal requests added to $\mathscr{TR}$. Two conditions can mediate the GBV algorithm. The first is $lsum \geq \mathscr{M}$. The second condition is that there are many invalid signatures in current $\mathscr{PR}$, namely $fsum = = n$, where n is the number of requests of current $\mathscr{PR}$.

Correctness: For the above equations, we can verify that

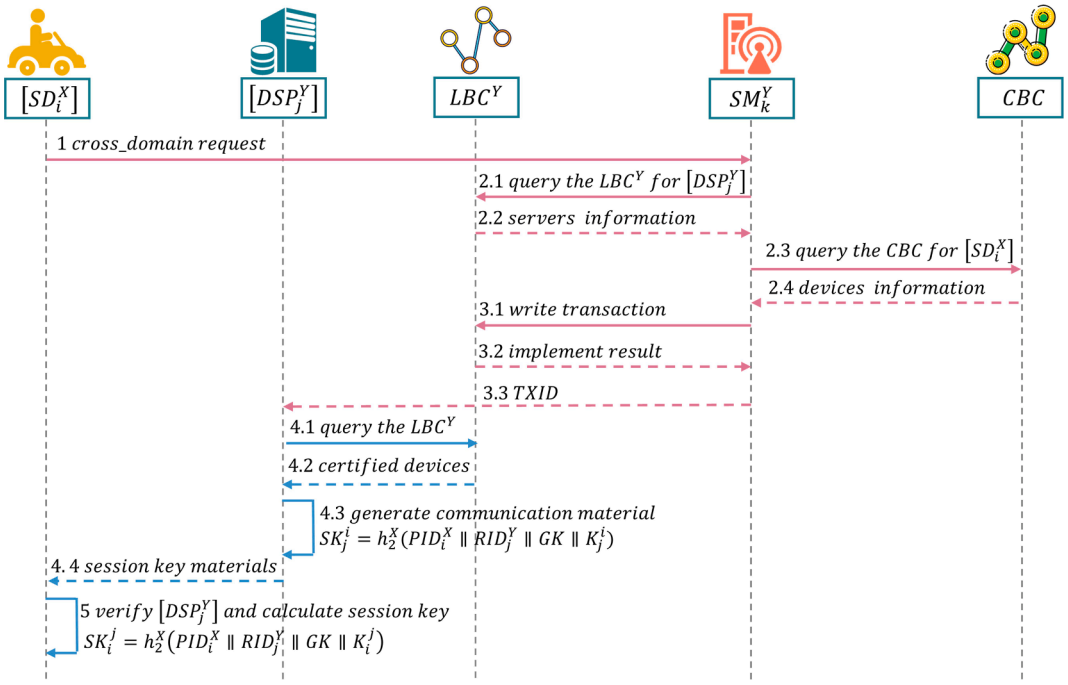

Fig. 2. The process of cross-domain authentication and key agreemnet.

$$
\begin{aligned}
(\sum_{i=1}^{n} DV_i) \bullet P_x &= \sum_{i=1}^{n} (r_i \bullet A_i + \sigma_i) \bullet P_x \\
&= \sum_{i=1}^{n} \left(r_i \bullet h_2^Y \left(B_i \| T_i \| \left[RID_j^Y \right] \right) + e_i + s_x \bullet \mu_i \right) \bullet P_x \\
&= \sum_{i}^{n} R_i \\
&\quad \bullet h_2^Y \left(B_i \| T_i \| \left[RID_j^Y \right] \right) + \sum_{i=1}^{n} E_i + \sum_{i=1}^{n} h_1^X \left(PID_i^X \| R_i \| E_i \right) \\
&\quad \bullet Pub_x \\
&= \sum_{i}^{n} R_i \\
&\quad \bullet h_2^Y \left(B_i \| T_i \| \left[RID_j^Y \right] \right) + \sum_{i=1}^{n} E_i + \left(\sum_{i=1}^{n} h_1^X \left(PID_i^X \| R_i \| E_i \right) \right) \\
&\quad \bullet Pub_x
\end{aligned} \tag{2}
$$

Therefore, equation (1) holds.

Algorithm 1 Groupable Batch Verification

Input: $\mathcal{PR} = \{req_i = (PID_i^X, R_i, E_i, B_i, DV_i, T_i)\}_{i=1}^n, \delta, \alpha$
Output: A list of legitimate requests $\mathcal{LR}$
Initialization: $\mathcal{LR} = \emptyset, \mathcal{FR} = \emptyset, lsum = 0;$

```
M = ⌈δ · n⌉;
while n > 1 do
  β = n mod α;
  i = 1;
  fsum = 0;
  for κ = 1,2, ..., α do
    if κ ≤ β then
      λ = ⌈n/α⌉;
    else
      λ = ⌈n/α⌉ − 1;
    end if
    Pick the λ requests req_i, req_{i+1}, ..., req_{i+λ−1} from the list PR;
    Verify the λ requests following Eq.(1);
    if Eq.(1) holds then
      Add the legitimate requests req_i, req_{i+1}, ..., req_{i+λ−1} to LR;
      lsum+= λ;
    else
      Add the failed requests req_i, req_{i+1}, ..., req_{i+λ−1} to FR;
      fsum+= λ;
    end if
    i+= λ;
  end for
  if lsum ≥ M || fsum == n then
    break;
  else
    PR = FR;
    FR = ∅;
    n = fsum;
  end if
end while
return LR;
```

Step 3: SM_k^Y calculates $\mathscr{Z}_i = r_k \bullet B_i$, $\theta_i = GK \oplus h_3^Y(\mathscr{Z}_i \| T_k)$, where T_k is the current timestamp. Then, it writes the authentication information $\left\{ \left[PID_j^Y \right], \left[(PID_i^X, R_i, B_i, \theta_i) \right]_{i=1}^n, gid, T_k \right\}$ of the legal devices into LBC^Y. The transaction code $TXID$ is sent to $\left[DSP_j^Y \right]$.

Step 4: $\left[DSP_j^Y \right]$ query LBC^Y to obtain relevant information. $\left[DSP_j^Y \right]$ calculate $F_j = f_j * P_x$, $\varpi_j = F_j \oplus GK$, where $f_j \in Z_{q_x}^*$ is a private random number. For the device SD_i^X, $\left[DSP_j^Y \right]$ calculate $K_j^i = f_j \bullet R_i + r_j \bullet B_i$, $SK_j^i = h_2^X \left(PID_i^X \| RID_j^Y \| GK \| K_j^i \right)$. Then, $\left[DSP_j^Y \right]$ calculate $PV_j^i = h_3^X \left(K_j^i \| SK_j^i \| T_k \right)$. The information $\left\{ \left(RID_j^Y, \varpi_j, PV_j^i, \theta_i, T_k \right) \right\}_{j=1}^m$ are sent to the device SD_i^X.

Step 5: If $|T_k^{'} - T_k| \le \Delta t$, SD_i^X calculates $\mathscr{Z}_i^{'} = b_i \bullet R_k$, $GK = \theta_i \oplus h_3^Y(\mathscr{Z}_i^{'} \| T_k)$, where $T_k^{'}$ is the timestamp of receiving the message. For the messages from $\left[DSP_j^Y \right]$, SD_i^X calculates $F_j^{'} = \varpi_j \oplus GK$, $K_i^j = r_i \bullet F_j^{'} + b_i \bullet R_j$, $SK_i^j = h_2^X \left(PID_i^X \| RID_j^Y \| GK \| K_i^j \right)$. If $h_3^X \left(K_i^j \| SK_i^j \| T_k \right) = PV_j^i$, SD_i^X will use SK_i^j as the session key communicating with $\left[DSP_j^Y \right]$.

Correctness: From the above equations, we obtain

$$
\begin{aligned}
K_j^i &= f_j \bullet R_i + r_j \bullet B_i = f_j \bullet r_i \bullet P_x + r_j \bullet b_i \bullet P_y = r_i \bullet F_j + b_i \bullet R_j \\
&= r_i \bullet F_j^{'} + b_i \bullet R_j = K_i^j
\end{aligned} \tag{3}
$$

Thus, we obtain $SK_j^i = SK_i^j$.

4.4. Pseudonym update and device revocation

4.4.1. Pseudonym update

For HBMA, devices can periodically update the pseudonym according to their privacy protection needs. The device SD_i^X of Domain X is taken as an example to update the pseudonym. SD_i^X picks a new key pair (r_i^{new}, R_i^{new}), where $r_i^{new} \in Z_q^*$, $R_i^{new} = r_i^{new} \bullet P_x$. SD_i^X calculates $\mathscr{Q} = h_4^X(RID_i^X \| R_i^{new} \| T_i)$. SD_i^X uses the old private key to generate a signature $DV_i^* = r_i^{old} \bullet \mathscr{Q} + \sigma_i^{old}$, and uses the public key of TA^X to generate the encrypted information $\left\{ Enc_{Pub_x}(update, RID_i^X, R_i^{new}, T_i) \right\}$. The signature and encrypted information are sent to TA^X. TA^X decrypts the message with its private key. Then, TA^X queries CBC to obtain the information of RID_i^X. If the equation $DV_i^* \bullet P_x = R_i^{old} \bullet h_4^X(RID_i^X \| R_i^{new} \| T_i) + E_i^{old} + h_1^X \left(PID_i^{X^{old}} \| R_i^{old} \| E_i^{old} \right) \bullet Pub_x$ holds, TA^X generates a new pseudonym $PID_i^{X^{new}} = h_0^X(RID_i^X \| \omega_i^{new})$ and new partial keys $\{E_i^{new}, \sigma_i^{new}\}$ for SD_i^X, where $E_i^{new} = e_i^{new} \bullet P_x$, $\mu_i^{new} = h_1^X(PID_i^{X^{new}} \| R_i^{new} \| E_i^{new})$, $\sigma_i^{new} = e_i^{new} + s_x \bullet \mu_i^{new}$. The identity information of SD_i^X in CBC and LBC^Y are updated, that is, the status of $SD_i^{X^{'}}$s old credential is modified to "invalid" and updated in the blockchain with the current timestamp, and $SD_i^{X^{'}}$s new credential are also written into the blockchain. The encrypted information $Enc_\varphi \left(PID_i^{X^{new}}, E_i^{new}, \sigma_i^{new} \right)$ is sent to SD_i^X, where $\varphi = s_x \bullet R_i^{new}$. SD_i^X uses $\varphi^{'}$ to decrypt the message, where $\varphi^{'} = r_i^{new} \bullet Pub_x$. If the equation $\sigma_i^{new} \bullet P_x = E_i^{new} + h_1^X \left(PID_i^{X^{new}} \| R_i^{new} \| E_i^{new} \right) \bullet Pub_x$ holds, SD_i^X saves the new identity $\left\{ RID_i^X, PID_i^{X^{new}}, r_i^{new}, \sigma_i^{new}, R_i^{new}, E_i^{new} \right\}$. The pseudonym update function for devices is shown in Algorithm 2.

Algorithm 2 Pseudonym Update

Input: $RID_i^X, R_i^{new}, T_i, DV_i^*$
Output: $PID_i^{X^{new}}, E_i^{new}, \sigma_i^{new}$

```
SD_i^X: Q = h_4^X(RID_i^X ‖ R_i^new ‖ T_i), DV_i^* = r_i^old · Q + σ_i^old;
SD_i^X to TA^X: {Enc_Pub_x(update, RID_i^X, R_i^new, T_i)}, DV_i^*;
TA^X: if DV_i^* · P_x = R_i^old · h_4^X(RID_i^X ‖ R_i^new ‖ T_i) + E_i^old +
      h_1^X(PID_i^Xold ‖ R_i^old ‖ E_i^old) · Pub_x then
    PID_i^Xnew = h_0^X(RID_i^X ‖ ω_i^new), {E_i^new, σ_i^new};
CBC and LBC^Y: Update the identity information of SD_i^X;
TA^X to SD_i^X: Enc_φ(PID_i^Xnew, E_i^new, σ_i^new);
SD_i^X: if σ_i^new · P_x = E_i^new + h_1^X(PID_i^Xnew ‖ R_i^new ‖ E_i^new) · Pub_x
    then
    New identity {RID_i^X, PID_i^Xnew, r_i^new, σ_i^new, R_i^new, E_i^new};
```

4.4.2. Device revocation

According to predefined behavior rules and identification algorithms, SM_k^N discovers a malicious device SD_i^X and informs the pseudonym PID_i^X to LEA. Subsequently, LEA traces back the real identity RID_i^X of SD_i^X. LEA adds RID_i^X to the blacklist in CBC, and notifies TA^X to revoke SD_i^X. The status of identity materials associated with SD_i^X in CBC and LBC^Y are modified to "invalid" by TA^X. The revocation list update for the local blockchain is done by SM_k^N. Taking Domain Y as an example, if there are relevant authentication records of SD_i^X in LBC^Y, SM_k^Y adds SD_i^X to the revocation list of LBC^Y.

In HBMA, updating the group key is used to block access to the network for the revoked identity. SM_k^Y sends a new group key GK^{new} to $\left[DSP_j^Y\right]$ through the secure channel. $\left[DSP_j^Y\right]$ queries the revocation list of LBC^Y and updates the session key for all legal identities. The encrypted information $Enc_{SK^{old}}(GK^{new})$ is sent to every legal identity SD_{le}^X, where SK^{old} is the old session key. SD_{le}^X calculates the new session key $SK^{new} = h_2^X\left(PID_{le}^X \| RID_j^Y \| GK^{new} \| K_{le}^i\right)$. Revoked devices that have not received the new group key will fail message verification. In other solutions, the servers verify the message authentication requests of devices by auditing the revocation list every time. For the message exchange in HBMA, the data service providers are required to check the revocation list of the local blockchain solely when the group key undergoes an update. Algorithm 3 demonstrates the device revocation functionality.

Algorithm 3 Device Revocation

Input: PID_i^X

Output: Revoke successful/failed

1: $\boldsymbol{SM_k^N}$ **to** $\boldsymbol{LEA}$: PID_i^X;
2: $\boldsymbol{LEA}$: Add RID_i^X to the blacklist in CBC;
3: $\boldsymbol{LEA}$ **to** $\boldsymbol{TA^X}$: PID_i^X;
4: $\boldsymbol{TA^X}$: modify the status of SD_i^X to "invalid";
5: $\boldsymbol{SM_k^N}$: Update the revocation list of LBC^N;
6: $\boldsymbol{SM_k^Y}$ **to** $\boldsymbol{\left[DSP_j^Y\right]}$: GK^{new};
7: $\boldsymbol{\left[DSP_j^Y\right]}$ **to** $\boldsymbol{SD_{le}^X}$: $Enc_{SK^{old}}(GK^{new})$;
8: $\boldsymbol{SD_{le}^X}$: $SK^{new} = h_2^X\left(PID_{le}^X \| RID_j^Y \| GK^{new} \| K_{le}^j\right)$;
9: **return** the result of execution;

5. Security analysis

The security properties of HBMA are analyzed in detail in this section.

(1) **Decentralization:** The HBMA scheme uses the Practical Byzantine Fault Tolerant (PBFT) algorithm as the consensus mechanism. Even if some nodes are attacked, as long as the failure nodes are less than 1/3, the blockchain network can still operate normally. If the adversary $\mathscr{A}$ tries to manipulate the system, it needs to master the computing power of $\frac{C_{\mathscr{A}}}{C_{'}} \geq \frac{2}{3}$, where $C_{\mathscr{A}}$ and $C_{'}$ represent the computing power of $\mathscr{A}$ and the total computing power of the system, respectively. It is difficult for $\mathscr{A}$ to have such a huge computing power to destroy the consensus protocol. Therefore, any identity information written into the blockchain cannot be modified.

(2) **Mutual authentication**: Firstly, $SD_i^X - to - DSP_j^Y$ authentication, suppose that the adversary $\mathscr{A}$ can forge valid authentication messages $\left\{PID_i^X, B_i, DV_i, T_i, RID_j^Y\right\}$, then $DV_i \bullet P_x = R_i \bullet v_1 + E_i + v_2 \bullet Pub_x$ can be obtained, where $v_1 = h_2^Y\left(B_i \| T_i \| RID_j^Y\right)$, $v_2 = h_1^X\left(PID_i^X \| R_i \| E_i\right)$. By invoking the forking lemma (Pointcheval and Stern, 2000), $\mathscr{A}$ performs the above process using the same input randomness but obtains a distinct answer from hash oracle. The defeated device SD_i^X forged $\left\{PID_i^{X^*}, B_i^*, DV_i^*, T_i^*, RID_j^{Y^*}\right\}$ to attain the service of DSP_j^Y. And $DV_i^* \bullet P_x = R_i^* \bullet v_1^* + E_i^* + v_1^* \bullet Pub_x$ can be obtained. Therefore, the following results can be obtained.

$$(DV_i - DV_i^*) \bullet P_x = (r_i v_1 - r_i^* v_1^*) \bullet P_x + (e_i - e_i^*) \bullet P_x + (v_1 - v_1^*) \bullet Pub_x \tag{4}$$

Transforming the equation (4), we can obtain

$$Pub_x = \frac{DV_i - DV_i^* - r_i v_1 + r_i^* v_1^* - e_i + e_i^*}{v_1 - v_1^*} \bullet P_x \tag{5}$$

This situation contradicts the ECDLP we have defined before. Hence it is impossible for PPT adversary $\mathscr{A}$ to break up this authentication protocol. The similar steps for $DSP_j^Y - to - SD_i^X$ authentication. Assuming that $\mathscr{A}$ outputs the valid messages $\left\{RID_j^Y, \varpi_j, PV_j^i, \theta_i, T_k\right\}$ during the interaction process to pass the verification of SD_i^X. $PV_j^i = h_3^X\left(K_j^i \| SK_j^i \| T_k\right)$ is an effective verifier. $\mathscr{A}$ must calculate $K^* = K_j^i = f_j \bullet R_i + r_j \bullet B_i$. Therefore, $\left(f_j \bullet r_i\right) \bullet P_x = K^* - r_j \bullet B_i$ can be obtained as the solution of ECCDH, which violates the ECCDH assumption. In other words, no PPT adversary can successfully forge valid authentication messages.

(3) **Privacy preservation:** The law enforcement agency is introduced to punish devices with malicious behavior. The anonymity authentication protocol ensures that any potential adversary $\mathscr{A}$ cannot obtain the real identity of devices to eavesdrop on the privacy of devices. Devices can periodically call the pseudonym update mechanism to protect their behavior from being completely linked by $\mathscr{A}$.

(4) **Perfect forward secrecy:** To figure out the correct session key, the adversary $\mathscr{A}$ must first obtain $\mathscr{Z}_i^{'} = b_i \bullet R_k$ to calculate the group key GK and $F_j^{'}$, and then further obtain the value of $K_i^j = r_i \bullet F_j^{'} + b_i \bullet R_j$. Even if the public keys R_k and R_j have been exposed to $\mathscr{A}$, due to the fact that the private key r_i and random key b_i are stored secretly by the device, $\mathscr{A}$ can only acquire a legitimate session key by breaking the ECCDH assumption. Assuming that the private keys of SD_i^X and DSP_j^Y are leaking, it is not possible for $\mathscr{A}$ to obtain the previous session key. Because b_i is a one-time random key, it provides guarantees for messages transmitted previously.

(5) **Resistance to common attacks:** The decentralization of blockchain ensures that each peer node has complete information, which can resist Distributed Denial of Service (DDoS) attacks. The freshness of each exchanged message must be check to combat replay attacks. Only devices with legitimate identities and private keys can pass verification. If the adversary $\mathscr{A}$ aims to masquerade as an authenticated device, it must destory the security of mutual authentication described above. Thus, HBMA also opposes impersonation attacks. According to the previous analysis, any authentication parameters forged by $\mathscr{A}$ cannot be verified by any party, which resists man-in-the-middle attacks.

Ulteriorly, some common features of authentication schemes are selected for comparing HBMA with Shen et al. (2020), Bagga et al. (2021), Wang et al. (2021), and Xue et al. (2022). The comparison result is shown in Table 2. The above five plans meet basic security requirements, including resistance to common attacks, mutual authentication, and forward secrecy. The schemes of Shen et al. (2020) and Wang et al. (2021) still have key escrow problems. Once the trusted authority leaks the private key of devices, it will cause serious

Table 2
Scheme features comparison.

Features	Shen et al. (2020)	Bagga et al. (2021)	Wang et al. (2021)	Xue et al. (2022)	HBMA
Resistance to common attacks	✓	✓	✓	✓	✓
Eliminate key escrow	×	✓	×	✓	✓
Mutual authentication	✓	✓	✓	✓	✓
Forward secrecy	✓	✓	✓	✓	✓
Pseudonym update	✓	×	×	×	✓
Groupable batch verification	×	×	×	×	✓
Authentication mode	1–1	m–1	m–1	1–1	m–m

1–1: one-to-one; m–1: many-to-one; m–m: many-to-many.

consequences. The schemes Bagga et al. (2021), Wang et al. (2021), and Xue et al. (2022) do not provide a pseudonym update function, where device behavior is easily linked by attackers, posing a threat to device privacy security. More superior to the other four schemes, HBMA provides a many-to-many authentication service and a flexible groupable batch verification algorithm.

6. Performance evaluation

We analyze and compare the performance of HBMA with Shen et al. (2020), Bagga et al. (2021), Wang et al. (2021), and Xue et al. (2022) from multiple perspectives in this section.

6.1. Computation and communication costs

We assume the parameters set in both domains are the same for better comparative analysis. Generally, system initialization and device registration are executed only once. Therefore, we focus only on the cost of the cross-domain authentication and key agreement phases. Particularly, the complexities of many-to-one and many-to-many communication types are considered from the perspectives of computation and communication costs.

6.1.1. Computation cost

Bagga et al. (2021) used the MIRACL library to evaluate the performance of some cryptographic operations on the server under the Ubuntu system and the Raspberry PI to simulate the computing power of the server and the device. The computational capability of devices from agricultural IoT systems is similar to these types of facilities. Hence, their parameters will be refered for comparison. Table 3 summarizes the experimental results of Bagga et al. (2021). Some lightweight operations (e.g., XOR, concatenation) are disregarded.

(1) **Case I. n devices to a server:** In Case I, we only consider the scenario where cross-domain authentication and key agreement occur between n devices and a single data server. Table 4 outlines the computational costs for each entity, encompassing devices, authentication servers (i.e., service managers), and data service providers. For the proposed solution, the computational expenses for n devices during the authentication phase involve $2n$ elliptic curve scalar multiplication operations, n elliptic curve point addition operations, and n hash function operations. In the key agreement phase, $3n$ elliptic curve scalar multiplication operations, n elliptic curve point addition operations, and $3n$ hash computations are required. Consequently, the total execution time for n devices is $5nT_{ecm} + 2nT_{eca} + 4nT_h = 12.708n(ms)$. The authentication server must complete identity verification and group key encryption tasks for n devices. Identity verification task necessitates $n+2$ elliptic curve multiplication operations, $4n-2$ elliptic curve addition operations, and $2n$ hash computations. Group key encryption consumes n elliptic curve multiplication operations and n hash computations. Thus, the overall cost for the authentication server is $(2n+2)T_{ecm} + (4n-2)T_{eca} + 3nT_h = 1.521n + 1.344(ms)$. Furthermore, a data service provider completing the key agreement process with n devices incurs a computational cost of $(2n+1)T_{ecm} + nT_{eca} + 2nT_h = 1.46n + 0.674(ms)$. In summary, the total computational cost for all entities in the many-to-one communication is $12.708n + 1.521n + 1.344 + 1.46n + 0.674 = 15.689n + 2.018(ms)$. Shen et al. (2020), Bagga et al. (2021), Wang et al. (2021), Xue et al. (2022) can be similarly analyzed, with computational costs for these solutions being $127.284n$, $33.762n + 15.6$, $15.56n + 1.344$, $22.99n$, respectively. Since the computational cost of Shen et al. (2020) is mainly bilinear pairing, the computational cost of this scheme is much higher than the other four schemes. To further analyze the impact of increasing cross-domain devices on computational overhead, HBMA is compared with Bagga et al. (2021), Wang et al. (2021), Xue et al. (2022). As shown in Fig. 3, the computational cost increases as the number of devices increases. The total computation cost growth rate of HBMA is lower than that of the schemes Bagga et al. (2021) and Xue et al. (2022) and is close to that of the scheme Wang et al. (2021).

(2) **Case II. n devices to m servers:** For Case II, the process of cross-domain authentication and key agreement between n devices and m data servers is considered. The computation results are summarized in Table 5. For the devices in HBMA, the cost of authentication in the initial phase of multi-to-multi communication is consistent with the cost in the multi-to-one scenario, denoted as $2nT_{ecm} + nT_{eca} + nT_h$. The cost for devices in the key negotiation phase is also $(2mn+n)T_{ecm} + mnT_{eca} + (2mn+n)T_h$. As a result, the total execution time for n devices is $(2mn+3n)T_{ecm} + (mn+n)T_{eca} + (2mn+2n)T_h = 5.21mn + 7.498n$. The computational cost required by the authentication server is $(2n+2)T_{ecm} + (4n-2)T_{eca} + 3nT_h = 1.521n + 1.344$, which is not affected by an increase in the number of data service providers. The computational expense incurred by m data service providers for key negotiation with n devices is $(2mn+m)T_{ecm} + mnT_{eca} + 2mnT_h = 1.46mn + 0.674m$. Therefore, the overall computational cost for all entities in the multi-to-multi scenario is $5.21mn + 7.498n + 1.521n + 1.344 + 1.46mn + 0.674m = 6.67mn + 9.019n + 0.674m + 1.344(ms)$. Similarly, the computational cost for Shen et al. (2020), Bagga et al. (2021), Wang et al. (2021), Xue et al. (2022) are $127.284mn$, $33.762mn + 15.6m$, $15.56mn + 1.344m$, and $22.99mn$, respectively. To analyze the effect of the number of data servers on computation cost, we compare Bagga et al. (2021), Wang et al. (2021), Xue et al. (2022) with HBMA. Suppose that the number of devices is n = 1000. The number of data servers takes values 5, 10, 15, 20, and 25 in sequence. The comparison results are demonstrated in Fig. 4. The computational overhead increases with the data servers increase. The total computation cost growth rate for HBMA is significantly lower than that of the other schemes.

Table 3
Execultion time of cryptographic operations.

Operation	Description	Server (ms)	Raspberry PI(ms)
T_h	General hash function	0.055	0.309
T_e	Modular exponentiation	0.072	0.228
T_{bp}	Bilinear pairing operation	4.716	32.084
T_{ecm}	Elliptic curve scalar multiplication	0.674	2.288
T_{eca}	Elliptic curve point addition	0.002	0.016
T_{enc}	ECC encryption	1.350	4.592
T_{dec}	ECC decryption	0.676	2.304

Table 4
Computation cost of various schemes in Case I.

Scheme	Device(ms)	Authentication(ms)	Data server(ms)	Total time(ms)
Shen et al. (2020)	$3nT_{bp} + 2nT_e + 5nT_{ecm} + nT_{eca} + 4nT_h = 109.4n$	$2nT_{bp} + nT_e + 2nT_{ecm} + nT_{eca} + 2nT_h = 10.964n$	$nT_{bp} + nT_e + 3nT_{ecm} + 2nT_h = 6.92n$	$127.284n$
Bagga et al. (2021)	$10nT_{ecm} + 7nT_{eca} + 5nT_h + nT_{dec} = 26.841n$	$3T_{bp} + (7n+1)T_{ecm} + (6n-3)T_{eca} + (2n+1)T_h = 4.84n + 14.871$	$(n+1)T_{ecm} + nT_{eca} + (n+1)T_h + nT_{enc} = 2.081n + 0.729$	$33.762n + 15.6$
Wang et al. (2021)	$5nT_{ecm} + nT_{eca} + 2nT_h = 12.074n$	$(2n+2)T_{ecm} + (3n-2)T_{eca} + nT_h = 1.409n + 1.344$	$3nT_{ecm} + nT_h = 2.077n$	$15.56n + 1.344$
Xue et al. (2022)	$7nT_{ecm} + 4nT_{eca} + 6nT_h = 17.934n$	$3nT_{ecm} + 2nT_{eca} + 3nT_h = 2.191n$	$4nT_{ecm} + 2nT_{eca} + 3nT_h = 2.865n$	$22.99n$
HBMA	$5nT_{ecm} + 2nT_{eca} + 4nT_h = 12.708n$	$(2n+2)T_{ecm} + (4n-2)T_{eca} + 3nT_h = 1.521n + 1.344$	$(2n+1)T_{ecm} + nT_{eca} + 2nT_h = 1.46n + 0.674$	$15.689n + 2.018$

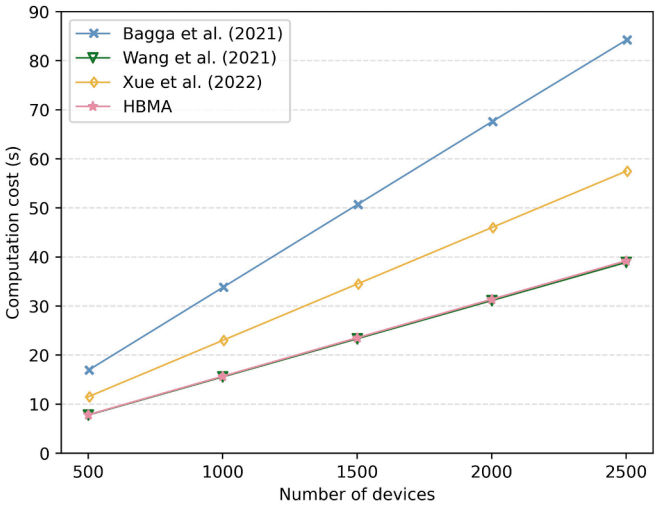

Fig. 3. Computation cost comparison in Case I.

6.1.2. *Communication cost*

Regarding the contrast of communication cost, we presume that the length of real identity, timestamp, and other messages is 4 bytes; the length of the elements in $\mathbb{G}$ is 64 bytes; the length of both the hash result and the blockchain transaction code is 32 bytes. This section summarizes the communication cost between different entities, where D_i denotes the device, DSP_j denotes the data service provider, and AS_j denotes the server for verification and authorization.

(1) **Case I. n devices to a server:** Table 6 summarizes the communication cost of various schemes for Case I. For the case where n devices request services from 1 data service provider, the proposed solution requires completing 5 message exchanges, including $D_i toAS_j : M1 = \{PID_i^X, B_i, DV_i, T_i, RID_j^Y\}$, $AS_j toCBC : M2 = \{[SD_i^X]_{i=1}^n\}$, $CBC\, toAS_j : M3 = \{PID_i^X, R_i, E_i, gid\}$, $AS_j toDSP_j : M4 = \{TXID\}$ and $DSP_j toD_i : M5 = \left\{\left(RID_j^Y, \varpi_j, PV_j^i, \theta_i, T_k\right)\right\}_{j=1}^m$. The HBMA scheme incurs the following communication costs for these 5 messages: $|M1| = 32n + 64n + 64n + 4n + 4n = 168n(bytes)$, $|M2| = 32n(bytes)$, $|M3| = 32n + 64n + 64n + 4n = 164n(bytes)$, $|M4| = 32(bytes)$, $|M5| = 4n + 64n + 32n + 32n + 4n = 136n(bytes)$. Therefore, for Case I, the total communication overhead is $|M1| + |M2| + |M3| + |M4| + |M5| = 168n + 32n + 164n + 32 + 136n = 500n + 32$. Following the same

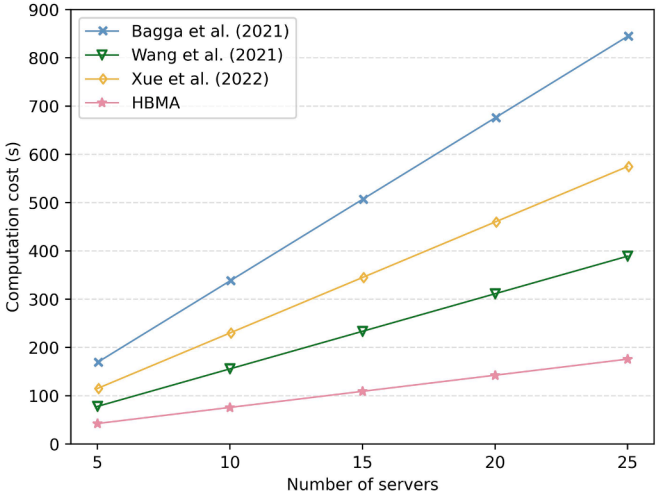

Fig. 4. Computation cost comparison in Case II.

Table 5
Computation cost of various schemes in Case II.

Scheme	Device(ms)	Authentication(ms)	Data server(ms)	Total time(ms)
Shen et al. (2020)	$3mnT_{bp} + 2mnT_e + 5mnT_{ecm} + mnT_{eca} + 4mnT_h = 109.4mn$	$2mnT_{bp} + mnT_e + 2mnT_{ecm} + mnT_{eca} + 2mnT_h = 10.964mn$	$mnT_{bp} + mnT_e + 3mnT_{ecm} + 2mnT_h = 6.92mn$	$127.284mn$
Bagga et al. (2021)	$10mnT_{ecm} + 7mnT_{eca} + 5mnT_h + mnT_{dec} = 26.841mn$	$3T_{bp} + m(7n+1)T_{ecm} + m(6n-3)T_{eca} + m(2n+1)T_h = 4.84mn + 14.871m$	$m(n+1)T_{ecm} + mnT_{eca} + m(n+1)T_h + mnT_{enc} = 2.081mn + 0.729m$	$33.762mn + 15.6m$
Wang et al. (2021)	$5mnT_{ecm} + mnT_{eca} + 2mnT_h = 12.074mn$	$m(2n+2)T_{ecm} + 3m(n-1)T_{eca} + mnT_h = 1.409mn + 1.344m$	$3mnT_{ecm} + mnT_h = 2.077mn$	$15.56mn + 1.344m$
Xue et al. (2022)	$7mnT_{ecm} + 4mnT_{eca} + 6mnT_h = 17.934mn$	$3mnT_{ecm} + 2mnT_{eca} + 3mnT_h = 2.191mn$	$4mnT_{ecm} + 2mnT_{eca} + 3mnT_h = 2.865mn$	$22.99mn$
HBMA	$(2mn+3n)T_{ecm} + (mn+n)T_{eca} + (2mn+2n)T_h = 5.21mn + 7.498n$	$(2n+2)T_{ecm} + (4n-2)T_{eca} + 3nT_h = 1.521n + 1.344$	$(2mn+m)T_{ecm} + mnT_{eca} + 2mnT_h = 1.46mn + 0.674m$	$6.67mn + 9.019n + 0.674m + 1.344$

Table 6
Communication cost of various schemes in Case I.

Scheme	Communication between different entities	Size(bytes)
Shen et al. (2020)	$D_i \xrightarrow{196n} DSP_j \xrightarrow{132n} AS_j \xrightarrow{32n} CBC \xrightarrow{96n} AS_j \xrightarrow{36n} DSP_j \xrightarrow{196n} D_i$	$688n$
Bagga et al. (2021)	$D_i \xrightarrow{520n} DSP_j \xrightarrow{32n} CBC \xrightarrow{36n} DSP_j \xrightarrow{456n} D_i$	$1044n$
Wang et al. (2021)	$D_i \xrightarrow{228n} DSP_j \xrightarrow{96n} CBC \xrightarrow{36n} DSP_j \xrightarrow{228n} D_i$	$588n$
Xue et al. (2022)	$D_i \xrightarrow{200n} DSP_j \xrightarrow{4n} CBC \xrightarrow{72n} DSP_j \xrightarrow{200n} D_i$	$476n$
HBMA	$D_i \xrightarrow{168n} AS_j \xrightarrow{32n} CBC \xrightarrow{164n} AS_j \xrightarrow{32} DSP_j \xrightarrow{136n} D_i$	$500n + 32$

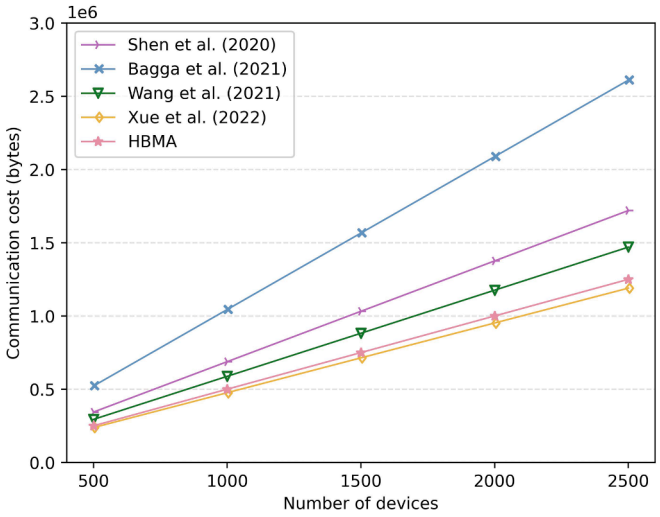

Fig. 5. Communication cost comparison in Case I.

calculation rules, the communication costs for the other schemes are $688n$, $1044n$, $588n$, $476n$, respectively. Assume that the number of devices takes values 500, 1000, 1500, 2000, and 2500 in sequence. The comparison results are shown in Fig. 5. The total communication overhead of HBMA is higher than that of the scheme Xue et al. (2022) and is lower than that of the plans Shen et al. (2020), Bagga et al. (2021), and Wang et al. (2021).

(2) **Case II. n devices to m servers:** Table 7 presents the communication results for Case II. Because HBMA requires the service manager to perform batch identity verification for devices, requesting services from n devices to m data service providers only requires sending n messages instead of nm messages. Case II needs to complete the same number of message exchanges as Case I, which is 5 times. The communication costs incurred by the five messages $D_i toAS_j : M1 = \{PID_i^X, B_i, DV_i, T_i, \left[RID_j^Y\right]_{j=1}^m\}$, $AS_j toCBC : M2 = \{[SD_i^X]_{i=1}^n\}$, $CBC\, toAS_j : M3 = \{PID_i^X, R_i, E_i, gid\}$, $AS_j toDSP_j : M4 = \{TXID\}$, $DSP_j toD_i : M5 = \left\{\left(RID_j^Y, \varpi_j, PV_j^i, \theta_i, T_k\right)\right\}_{j=1}^m$ are sequentially denoted as $|M1| = 32n + 64n + 64n + 4n + 4mn = 164n + 4mn(bytes)$, $|M2| = 32n(bytes)$, $|M3| = 32n + 64n + 64n + 4n = 164n(bytes)$, $|M4| = 32m(bytes)$, $|M5| = m \bullet (4n + 64n + 32n + 32n + 4n) = 136mn(bytes)$. Therefore, the total communication cost for the many-to-many communication type is $|M1| + |M2| + |M3| + |M4| + |M5| = 164n + 4mn + 32n + 164n + 32m + 136mn = 140mn + 32m + 360n$. Similarly, the communication costs for the other schemes are $688mn$, $1044mn$, $588mn$, $476mn$, respectively. Assuming that the number of devices is n = 1000, the number of data servers takes values 5, 10, 15, 20, and 25 in sequence. Fig. 6 depictes the comparison outcomes. The total communication cost of HBMA is significantly lower than that of the other schemes.

6.2. Blockchain performance

To measure the performance of the hybrid blockchain, we implemented a prototype of the HBMA scheme. The experimental environment is Ubuntu18.04(64bit), 11th gen Intel(R) Core™ i5-11400F @2.6GHz× 4 processor, 7.7G RAM. The hybrid blockchain network is built based on Hyperledger Fabric 1.4.12. The docker version is 24.0.1. Smart contract (chaincode) is developed with go 1.20.4. This experiment uses the channel technology of Hyperledger Fabric to build a hybrid blockchain network consisting of a consortium chain and several private chains.

The authentication protocol of HBMA includes query operation of consortium blockchain and query and write operation of local private blockchain. This paper designs three groups of experiments to test the write operation of the private blockchain, in which the number of nodes is 4, 8, and 12, respectively. We focus on the characteristics of write latency by increasing the number of concurrent transactions within 1 s. The experiment for each condition was conducted 10 times, and the average value was taken as the final test result. Table 8 summarizes the measurement results of write latency. Correspondingly, Fig. 7 illustrates the comparison of write latency in private blockchains with different numbers of nodes. It can be seen that the total latency for writing increases gradually as the concurrent transaction times increase. In addition, as the number of nodes increases, the write latency also increases. For smart agricultural IoT scenarios, the number of server nodes for a single domain is usually in a small number range. And HBMA only needs to perform one write operation to complete one batch authentication. Therefore, the write delay for the batch authentication of massive

Table 7
Communication cost of various schemes in Case II.

Scheme	Communication between different entities	Size (bytes)
Shen et al. (2020)	$D_i \xrightarrow{196mn} DSP_j \xrightarrow{132mn} AS_j \xrightarrow{32mn} CBC \xrightarrow{96mn} AS_j \xrightarrow{36mn} DSP_j \xrightarrow{196mn} D_i$	$688mn$
Bagga et al. (2021)	$D_i \xrightarrow{520mn} DSP_j \xrightarrow{32mn} CBC \xrightarrow{36mn} DSP_j \xrightarrow{456mn} D_i$	$1044mn$
Wang et al. (2021)	$D_i \xrightarrow{228mn} DSP_j \xrightarrow{96mn} CBC \xrightarrow{36mn} DSP_j \xrightarrow{228mn} D_i$	$588mn$
Xue et al. (2022)	$D_i \xrightarrow{200mn} DSP_j \xrightarrow{4mn} CBC \xrightarrow{72mn} DSP_j \xrightarrow{200mn} D_i$	$476mn$
HBMA	$D_i \xrightarrow{4mn+164n} AS_j \xrightarrow{32n} CBC \xrightarrow{164n} AS_j \xrightarrow{32m} DSP_j \xrightarrow{136mn} D_i$	$140mn + 32m + 360n$

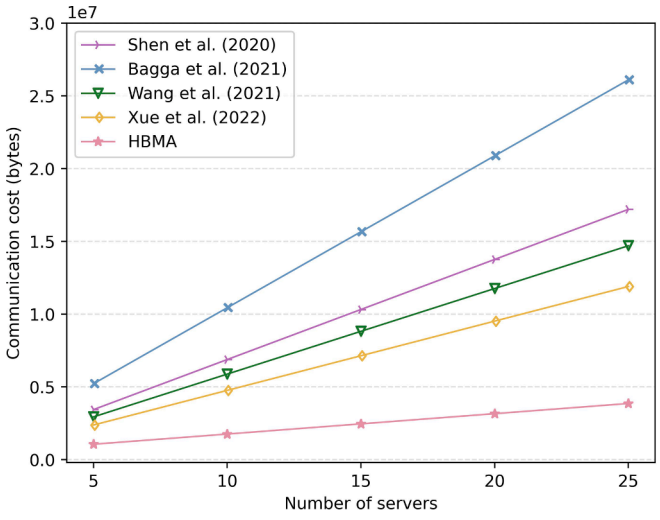

Fig.6. Communication cost comparison in Case II.

Table 8
Measurement results of write latency.

Number of nodes	Number of concurrent transactions				
	50	100	150	200	250
4	0.65 s	1.35 s	1.95 s	2.38 s	3.04 s
8	0.96 s	1.81 s	3.38 s	4.33 s	5.28 s
12	1.62 s	2.90 s	4.90 s	6.51 s	8.68 s

devices does not increase significantly.

A query experiment was also designed to test two different types of blockchains in the hybrid-chain model, with 5 nodes in the private blockchain and 10 nodes in the consortium blockchain. By increasing the number of concurrent transactions within 1 s, we pay attention to the performance metrics of query latency. The measurement results of query latency are shown in Table 9. Furthermore, Fig. 8 illustrates the comparison of query latency between private blockchain and consortium blockchain. It is evident that the query latency rises as the number of concurrent transactions increases. There is no sorting and transaction synchronization process between nodes for query transactions, so the response speed is fast. Query latency is not affected by the number of nodes.

In order to better assess the performance of the hybrid blockchain, throughput comparison experiments were conducted between the hybrid-chain and the single-chain model. The number of domains in the three sets of experiments was set to 2, 4, and 6, with 5 nodes in each domain. Correspondingly, the number of nodes in the single-chain structure was the product of the number of domains and 5. For the hybrid-chain model of the three sets of experiments, the number of nodes in the consortium blockchain was set to 3, 5, and 7 respectively, while the number of nodes in the private blockchain was 5. Throughput (TPS) is a metric that measures the system's ability to process requests or transactions in a unit of time, as shown in Equation (6).

$$TPS = Sumtransactions/\Delta t \tag{6}$$

TPS represents the total number of transactions completed by the blockchain within the time interval Δt. Experiments were conducted separately for the single-chain structure and the hybrid-chain model involving both write and query transactions. The write operation experiment involved initiating 100 intra-domain write transactions and 50 inter-domain write transactions simultaneously in each domain. The query operation experiment involved initiating 300 intra-domain query transactions and 100 inter-domain query transactions simultaneously in each domain. Every set of experiments was conducted 10 times, and the average values were recorded.

Fig. 9 illustrates the comparison of write throughput between the single-chain structure and the hybrid-chain architecture. When the number of domains is 6, the write throughput of the single-chain model is 15, while the hybrid-chain's write throughput is 504. Fig. 10 summarizes the query throughput comparison results between the single-chain model and the hybrid-chain model. When the number of domains is 6, the query throughput of the single-chain architecture is 112, whereas the query throughput of the hybrid-chain is 837. From Figs. 9 and 10, it can be observed that as the number of domains gradually increases, the write throughput of the single-chain decreases linearly, and the query throughput of the single-chain decreases slowly, while the hybrid-chain model exhibits a linear growth trend in both write and query throughputs. In a multi-domain environment, the write and query throughputs of the hybrid-chain are significantly higher than those of the single-chain. This is mainly because the hybrid blockchain network

Table 9
Measurement results of query latency.

Measurement object	Number of concurrent transactions				
	50	100	150	200	250
Private blockchain (5 nodes)	0.39 s	0.77 s	1.12 s	1.65 s	1.98 s
consortium blockchain (10 nodes)	0.42 s	0.77 s	1.25 s	1.66 s	2.08 s

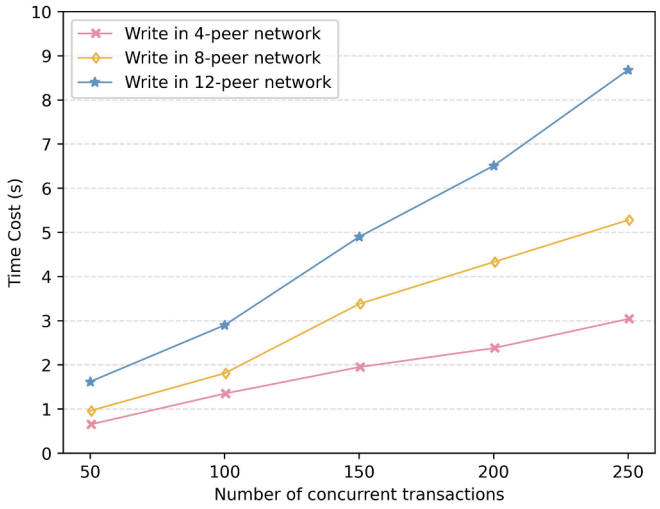

Fig. 7. Write latency of private blockchain.

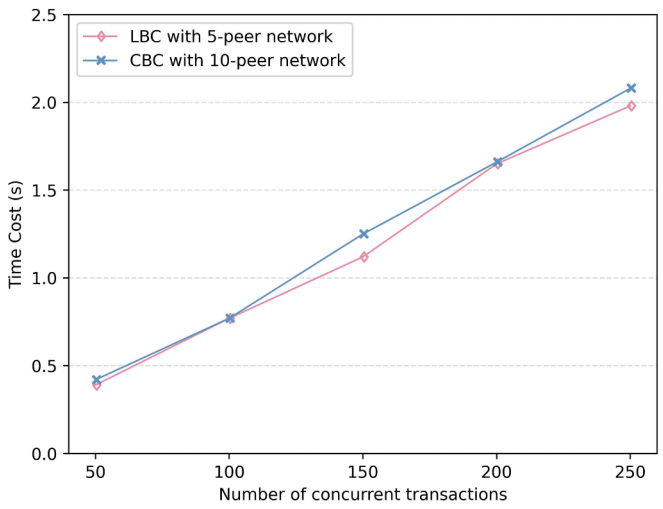

Fig. 8. Query latency of two types of blockchain.

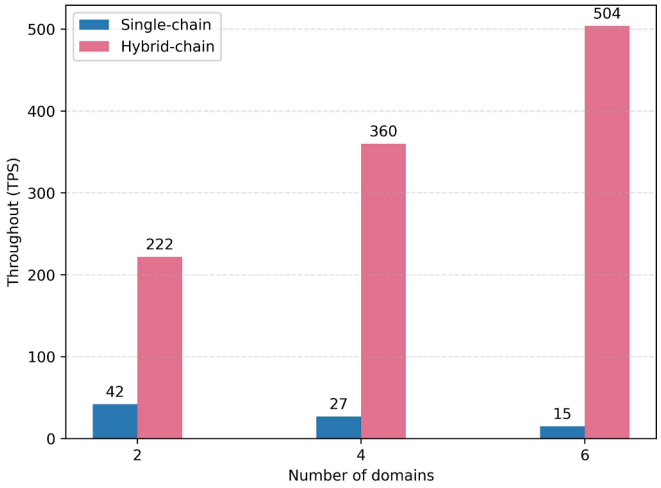

Fig. 9. Write throughput of single-chain and hybrid-chain.

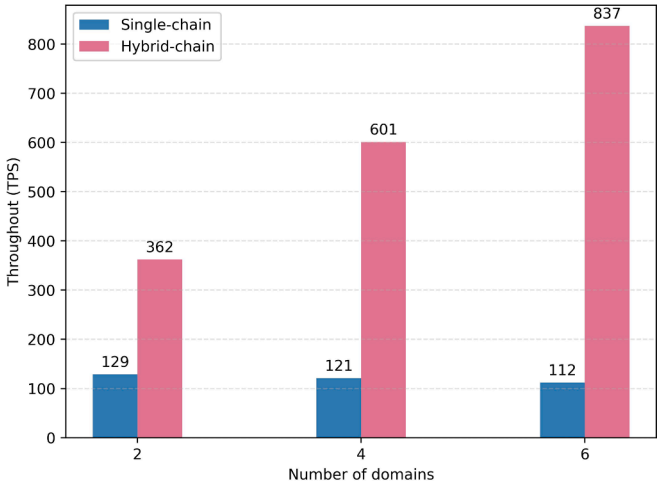

Fig. 10. Query throughput of single-chain and hybrid-chain.

is more decentralized, enhancing the efficiency of parallel transaction processing across domains. For HBMA, all transactions within a domain and certain operations in cross-domain authentication are conducted through the private blockchain. This reduces the load on the consortium blockchain, effectively enhancing the efficiency of cross-domain authentication. Thus, the hybrid blockchain is more suitable for implementing cross-domain authentication solutions than the single-chain model.

There is heterogeneity in password settings and access control for devices within different domains, and the single-chain model results in a mix-up of identity information and transaction data across multiple domains. Compared to a single blockchain type, the proposed hybrid blockchain model in this paper exhibits superior security features. Firstly, in the hybrid blockchain structure, the ability to choose the appropriate chain for executing smart contracts or storing data based on business requirements allows for adaptability to various security configurations, thereby enhancing system scalability. The private blockchain is only open internally, strictly controlling access to the internal network. For specific business scenarios, execution on a private chain can be chosen to ensure network security and device privacy protection. Secondly, in some industries, compliance with specific regulations and regulatory standards is necessary. The hybrid blockchain model introduced in this paper incorporates agricultural regulatory agencies into the consortium blockchain, regulating local agricultural organizations' cooperation in accordance with legal provisions and agricultural production principles to ensure the compliance and fairness of agricultural production.

Based on the comprehensive analysis above, it can be concluded that the hybrid blockchain model proposed in this paper possesses superior performance, excellent scalability and robust security features.

7. Conclusion and future work

This paper proposes a hybrid blockchain-based many-to-many cross-domain authentication scheme for agricultural IoT systems with varying cryptographic configurations, addressing issues related to certificate management and key escrow. An algorithm for batch verification with grouping is introduced, allowing for the adjustment of batch sizes based on specific tasks, thereby improving the versatility of cross-domain batch verification. Additionally, the pseudonym update and device revocation protocols protect the privacy of devices and effectively punish malicious devices. Security analysis and performance evaluation show the practical security, efficiency, and affordability of this scheme. The HBMA solution incurs relatively low computational overhead and communication costs in many-to-many cross-domain authentication. Moreover, the incurred expenses do not exhibit a substantial increase with the growth of devices and servers. In contrast to a single-chain structure, the hybrid blockchain model implemented in this solution not only boosts transaction throughput for cross-domain collaboration but also fortifies the scalability of agricultural production organizations and the safety control for cross-domain cooperation.

Through comprehensive analysis, it is evident that HBMA is well-suited for large-scale agricultural IoT scenarios across multiple domains. Next, we will improve the PBFT consensus algorithm of the blockchain based on the cross-domain behavior analysis of devices.

CRediT authorship contribution statement

Fengting Luo: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Validation, Visualization, Writing – original draft. **Ruwei Huang:** Conceptualization, Investigation, Methodology, Project administration, Resources, Supervision, Validation, Writing –review & editing. **Yuqi Xie:** Investigation, Data curation, Writing – original draft.

Funding

This work was supported by the National Natural Science Foundation Project of China under Grant No. 62062009.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Ali, R., Pal, A.K., Kumari, S., Karuppiah, M., Conti, M., 2018. A secure user authentication and key-agreement scheme using wireless sensor networks for agriculture monitoring. Futur. Gener. Comput. Syst. 84, 200–215.

Alyahya, S., Khan, W.U., Ahmed, S., Marwat, S.N.K., Habib, S., 2022. Cyber secure framework for smart agriculture: robust and tamper-resistant authentication scheme for iot devices. Electronics 11, 963.

Bagga, P., Sutrala, A.K., Das, A.K., Vijayakumar, P., 2021. Blockchain-based batch authentication protocol for internet of vehicles. J. Syst. Archit. 113, 101877.

Bera, B., Vangala, A., Das, A.K., Lorenz, P., Khan, M.K., 2022. Private blockchain-envisioned drones-assisted authentication scheme in iot-enabled agricultural environment. Computer Standards & Interfaces 80, 103567.

Bothe, A., Bauer, J., Aschenbruck, N., 2019. Rfid-assisted continuous user authentication for iot-based smart farming. In: 2019 IEEE International Conference on RFID Technology and Applications (RFID-TA). IEEE. pp. 505–510.

Chae, C.J., Cho, H.J., 2018. Enhanced secure device authentication algorithm in p2p-based smart farm system. Peer-to-Peer Netw. Appl. 11, 1230–1239.

Chen, M., Lee, T.F., Pan, J.I., 2019. An enhanced lightweight dynamic pseudonym identity based authentication and key agreement scheme using wireless sensor networks for agriculture monitoring. Sensors 19, 1146.
Dos Santos, U.J.L., Pessin, G., da Costa, C.A., da Rosa Righi, R., 2019. Agriprediction: a proactive internet of things model to anticipate problems and improve production in agricultural crops. Comput. Electron. Agric. 161, 202–213.
Ferrag, M.A., Derdour, M., Mukherjee, M., Derhab, A., Maglaras, L., Janicke, H., 2018. Blockchain technologies for the internet of things: Research issues and challenges. IEEE Internet Things J. 6, 2188–2204.
Housley, R., Ford, W., Polk, W., Solo, D., 1999. Internet X. 509 public key infrastructure certificate and CRL profile. Technical Report.
Huang, H., Zhou, S., Lin, J., Zhang, K., Guo, S., 2020. Bridge the trustworthiness gap amongst multiple domains: a practical blockchain-based approach. In: ICC 2020-2020 IEEE International Conference on Communications (ICC). IEEE. pp. 1–6.
Katz, J., Lindell, Y., 2014. Introduction to Modern Cryptography. Chapman & Hall/CRC 10, b17668.
Khanal, S., Fulton, J., Shearer, S., 2017. An overview of current and potential applications of thermal remote sensing in precision agriculture. Comput. Electron. Agric. 139, 22–32.
Li, Y., Chen, W., Cai, Z., Fang, Y., 2016. Caka: a novel certificateless-based cross-domain authenticated key agreement protocol for wireless mesh networks. Wirel. Netw 22, 2523–2535.
Liu, Q., Gong, B., Ning, Z., 2020. Research on clpkc-idpkc cross-domain identity authentication for iot environment. Comput. Commun. 157, 410–416.
Liu, H., Luo, P., Wang, D., 2008. A scalable authentication model based on public keys. J. Netw. Comput. Appl. 31, 375–386.
Makhdoom, I., Abolhasan, M., Lipman, J., Liu, R.P., Ni, W., 2018. Anatomy of threats to the internet of things. IEEE Commun. Surv. Tutorials 21, 1636–1675.
Menezes, A.J., Vanstone, S.A., 1993. Elliptic curve cryptosystems and their implementation. J. Cryptol. 6, 209–224.
Pointcheval, D., Stern, J., 2000. Security arguments for digital signatures and blind signatures. J. Cryptol. 13, 361–396.
Rangwani, D., Sadhukhan, D., Ray, S., Khan, M.K., Dasgupta, M., 2021. An improved privacy preserving remote user authentication scheme for agricultural wireless sensor network. Trans. Emerg. Telecommun. Technol. 32, e4218.
Shamir, A., 1985. Identity-based cryptosystems and signature schemes. In: Advances in Cryptology: Proceedings of CRYPTO 84 4, Springer. pp. 47–53.
Shen, M., Zhang, J., Zhu, L., Xu, K., Tang, X., 2019. Secure svm training over vertically-partitioned datasets using consortium blockchain for vehicular social networks. IEEE Trans. Veh. Technol. 69, 5773–5783.
Shen, M., Liu, H., Zhu, L., Xu, K., Yu, H., Du, X., Guizani, M., 2020. Blockchain-assisted secure device authentication for cross-domain industrial iot. IEEE J. Sel. Areas Commun. 38, 942–954.
Torky, M., Hassanein, A.E., 2020. Integrating blockchain and the internet of things in precision agriculture: analysis, opportunities, and challenges. Comput. Electron. Agric. 178, 105476.
Vangala, A., Das, A.K., Kumar, N., Alazab, M., 2020. Smart secure sensing for iot-based agriculture: blockchain perspective. IEEE Sens. J. 21, 17591–17607.
Vangala, A., Sutrala, A.K., Das, A.K., Jo, M., 2021. Smart contract-based blockchain-envisioned authentication scheme for smart farming. IEEE Internet Things J. 8, 10792–10806.
Vangala, A., Das, A.K., Mitra, A., Das, S.K., Park, Y., 2022. Blockchain-enabled authenticated key agreement scheme for mobile vehicles-assisted precision agricultural iot networks. IEEE Trans. Inf. Forensics Secur. 18, 904–919.
Wang, W., Huang, H., Zhang, L., Su, C., 2021. Secure and efficient mutual authentication protocol for smart grid under blockchain. Peer-to-Peer Netw. Appl. 14, 2681–2693.
Wu, H.T., Tsai, C.W., 2019. An intelligent agriculture network security system based on private blockchains. J. Commun. Networks 21, 503–508.
Xue, L., Huang, H., Xiao, F., Wang, W., 2022. A cross-domain authentication scheme based on cooperative blockchains functioning with revocation for medical consortiums. IEEE Trans. Netw. Serv. Manag. 19, 2409–2420.
Yang, Y., Wei, L., Wu, J., Long, C., Li, B., 2021. A blockchain-based multidomain authentication scheme for conditional privacy preserving in vehicular ad-hoc network. IEEE Internet Things J. 9, 8078–8090.
Yuan, C., Zhang, W., Wang, X., 2017. Eimakp: Heterogeneous cross-domain authenticated key agreement protocols in the eim system. Arab. J. Sci. Eng. 42, 3275–3287.
Zhang, J., Zhong, H., Cui, J., Xu, Y., Liu, L., 2020. Smaka: Secure many-to-many authentication and key agreement scheme for vehicular networks. IEEE Trans. Inf. Forensics Secur. 16, 1810–1824.

Fengting Luo received her B.Eng. degree in computer science and technology at Guilin University of electronic technology in 2019, PR China. She is a M.Eng degree candidate in information security at Guangxi University. She is presently engaged in blockchain and information security of IoT.

Ruwei Huang received her B.Eng., M.Eng. degree in computer science and technology at Guangxi University in 2001 and 2004 respectively, received Ph.D. degree in computer science and technology at Xi'an Jiaotong University in 2012, PR China. She is resently engaged in cryptography and cloud computing security as an associate professor at Guangxi University.

Yuqi Xie received his B.Eng. degree in Information Warfare Technology at Southwest University of science and technology in 2021, PR China. He is a M.Eng degree candidate in information security at Guangxi University. He is presently engaged in cryptography and cloud computing security.