drones

Article

BCDAIoD: An Efficient Blockchain-Based Cross-Domain Authentication Scheme for Internet of Drones

Gongzhe Qiao [1], Yi Zhuang [1,*], Tong Ye [1] and Yuan Qiao [2]

[1] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211100, China; qgz@nuaa.edu.cn (G.Q.); yetong@nuaa.edu.cn (T.Y.)
[2] Harbin Electric Power Bureau, STATE GRID Corporation of China, Harbin 150050, China; qiaoyuan_qy@163.com
* Correspondence: zy16@nuaa.edu.cn

Abstract: During long-distance flight, unmanned aerial vehicles (UAVs) need to perform cross-domain authentication to prove their identity and receive information from the ground control station (GCS). However, the GCS needs to verify all drones arriving at the area it is responsible for, which leads to the GCS being unable to complete authentication in time when facing cross-domain requests from a large number of drones. Additionally, due to potential threats from attackers, drones and GCSs are likely to be deceived. To improve the efficiency and security of cross-domain authentication, we propose an efficient blockchain-based cross-domain authentication scheme for the Internet of Drones (BCDAIoD). By using a consortium chain with a multi-chain architecture, the proposed method can query and update different types of data efficiently. By mutual authentication before cross-domain authentication, drones can compose drone groups to lighten the authentication workload of domain management nodes. BCDAIoD uses the notification mechanism between domains to enable path planning for drones in advance, which can further improve the efficiency of cross-domain authentication. The performance of BCDAIoD was evaluated through experiments. The results show that the cross-domain authentication time cost and computational overhead of BCDAIoD are significantly lower those of than existing methods when the number of drones is large.

Keywords: blockchain; Internet of Drone; cross-domain authentication; security protocol; multi-UAVs; group policy

Citation: Qiao, G.; Zhuang, Y.; Ye, T.; Qiao, Y. BCDAIoD: An Efficient Blockchain-Based Cross-Domain Authentication Scheme for Internet of Drones. *Drones* **2023**, *7*, 302. https://doi.org/10.3390/drones7050302

Academic Editors: Zhihong Liu, Shihao Yan, Yirui Cong and Kehao Wang

Received: 3 April 2023
Revised: 28 April 2023
Accepted: 3 May 2023
Published: 4 May 2023

1. Introduction

The increasing demand for unmanned aerial vehicles (UAVs) and Internet of Drones (IoD) in civil and military applications has been noticed in the last few decades [1]. IoD usually consists of a number of drones, ground control stations (GCS), and a communication network for data exchange. Drones can obtain information through sensors and exchange information through the network. They can also communicate with GCS and other drones for proper navigation [2]. The available space of the air traffic network (ATN) is much larger than that of the ground traffic network (GTN). Reasonable use of the ATN can reduce the burden of the GTN. Furthermore, using ATN can also avoid the congestion of the GTN. Therefore, many companies try to use drones as air transport tools for cargo transportation, so as to promote logistics efficiency [3].

During long-distance flights, drones are likely to enter other IoD domains where they need to pass identity authentication and obtain necessary information such as flight routes to continue their tasks. In the IoD, wireless communication between drones is easy to eavesdrop on, resulting in information leakage [4–7]. Furthermore, attackers can conduct replay attacks or identity forgery to interrupt the IoD [8–10]. In addition, once drone transportation forms a complete industry, the number of drones may be quite large. The resource overhead of complex authentication mechanisms, such as computation and

storage, is a huge challenge for identity authentication servers [11–13]. At the same time, it also increases the cross-domain waiting time for drones to obtain necessary information.

Centralized authentication techniques [14–16] are widely adopted in traditional information systems. However, in an IoD environment, if conventional centralized authentication approaches are applied, the workload of the authentication service center increases exponentially as the system scales up [17]. Therefore, traditional authentication technology is not suitable for the IoD environment. At present, there are also some studies using public key infrastructure (PKI), trusted third party (TTP), or blockchain technology to verify the identity of the drone and the authenticity of the task [18,19]. The existing methods are shown in Figure 1a. However, the existing studies rarely consider the efficiency of cross-domain authentication in the case of a large number of drones. Additionally, when the number of drones is large, the time cost of cross-domain authentication is also very high.

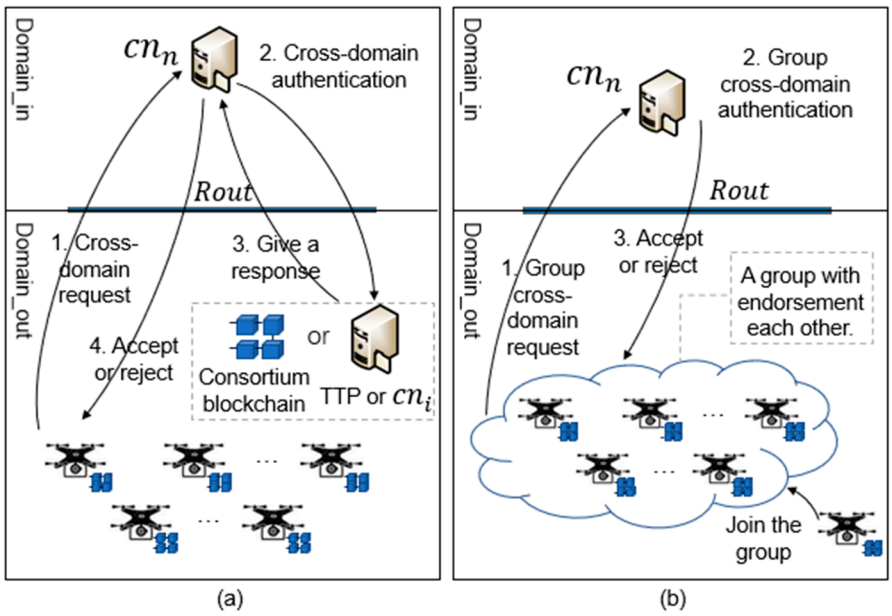

Figure 1. Comparison of existing methods and this paper. (**a**) Ideas of existing methods. (**b**) The idea of the method proposed in this paper.

When facing a large number of drones, existing drone cross-domain authentication methods lead to a surge in communication and computational costs for the IoD, increase the waiting time of the drones, and further reduce the overall operational efficiency of the IoD. At the same time, the existing methods cannot guarantee the anonymity of drones when performing the task, and attackers can obtain the connection between the sender and receiver through the identity and task information of the drone, which may leak the user's privacy. Therefore, existing methods cannot meet the cross-domain authentication requirements for large-scale drone cargo transportation scenarios. Therefore, we propose a novel cross-domain authentication scheme for the IoD based on blockchain technology.

Blockchain has the advantages of decentralization, openness, being tamper-proof, traceability, and anonymity [20]. The decentralization is consistent with the distributed network structure of the IoD. Furthermore, the openness can well satisfy the requirements for drones to join and leave the network freely [21]. The advantages of being tamper-proof and traceability ensure the reliability and non-repudiation of data in the network, and the advantage of anonymity protects the data by addressing with address instead of addressing with identity. However, there are three potential challenges to using blockchain for authentication. (1) In the IoD, there are various types of data and the scale of data is large, meaning the single-chain structure may lead to query inefficiency and throughput bottleneck. (2) In the authentication process, a large number of drones may cause a high

authentication delay time and increase the waiting time of drones. (3) The authentication server needs to authenticate the UAV identity information, verify the transaction in the blockchain, and upload the task information. A large number of drones to be authenticated may cause server interruptions.

To address the above challenges, we propose an efficient cross-domain authentication scheme based on blockchain. As shown in Figure 1b, the main idea is that drones should authenticate each other before cross-domain authentication to form a mutually endorsed group. To ensure that the identities and tasks of drones in each group are real and trusted, we designed an authentication mechanism based on domain signature, encryption, and domain private chain. In this way, drones with the same out-of-domain range (Rout) compose a drone group. Considering the continuity of drone tasks in the process of cross-domain tasks, we designed a notification mechanism between domains combined with the concepts of token, permission, and authority. The main contributions of this paper are as follows.

(1) We propose an efficient blockchain-based cross-domain authentication scheme for the Internet of Drones (BCDAIoD). By using a consortium chain with a multi-chain architecture, the proposed method can query and update different types of data efficiently, which can also facilitate the domain management node to manage and control the drones. Additionally, we describe the mission model of the drones.

(2) In order to improve the efficiency of the cross-domain authentication of drones, we designed an establishment method of drone groups and a group cross-domain authentication method based on blockchain, encryption, and challenge–response game. By mutual authentication before cross-domain authentication, drones can compose drone groups to lighten the authentication workload of domain management nodes and improve the efficiency of cross-domain authentication.

(3) We propose a notification mechanism between domains that can enable the management node of the next domain to know the task information of the drones in advance. The management node of the next domain can plan space resources reasonably and plan the flight path for drones in advance, which can also ensure the continuity of the tasks of drones.

The remainder of this article is organized as follows. The literature is reviewed in Section 2. The framework of BCDAIoD, the consortium blockchain architecture, and the mission model of the drones are presented in Section 3. In Section 4, we first propose the single-drone cross-domain authentication method. Then, we propose the establishment mechanism of drone groups, the drone group cross-domain authentication method, and the notification mechanism between domains. Section 5 describes the simulation experiments and shows the experimental evaluation results of the BCDAIoD method. In Section 6, we analyze the security of BCDAIoD. Finally, Section 7 concludes this paper.

2. Related Works

(1) IoD management scheme

IoD has received widespread attention due to its potential application prospects. Therefore, there are studies targeting the management scheme of IoD. In order to facilitate the information acquisition of drones and users in IoD, Al-Hilo et al. [22] proposed a collaborative and management framework between UAVs and roadside units. Arafeh et al. [23] proposed a blockchain-based UAV management method that can verify the authenticity of information in IoD networks. By using blockchain and trust policies, García-Magariño et al. [24] proposed a UAV management approach which can also maintain security in IoD by corroborating information about events from different sources. However, the above UAV management framework mainly focuses on data security and neglects the management efficiency when facing a large number of drones. Additionally, the blockchain-based methods have not been able to reasonably partition the data storage on the chain, leading to low query efficiency and data isolation.

(2) Cross-domain authentication scheme based on cryptography

During the task execution, drones need to verify their identity through cross-domain authentication. Meanwhile, in order to address potential threats, researchers have proposed some cross-domain authentication schemes based on cryptography. To protect drones against various types of possible attacks, Wazid et al. [25] proposed a remote authentication and key management scheme. Srinivas et al. [26] designed a three-factor authentication scheme which relies on elliptic-curve cryptography (ECC). Tanveer et al. [27] leveraged ECC along with symmetric encryption and hash function, and proposed a robust authentication mechanism for IoD. For the sensitive environment in which an attacker might trap data from the open network channel, Jan et al. [28] proposed a verifiably secure ECC-based authentication scheme for IoD. Additionally, Rajamanickam et al. [29] proposed an ECC-based authentication protocol for insider attack protection in IoD scenarios that can protect the IoD from several known attacks, especially insider attacks. Ever et al. [30] proposed a secure authentication framework using elliptic-curve crypto-systems to ensure data confidentiality. However, the above methods guarantee the security of IoD by performing multiple process parameter calculations on the device, registration center, and control center, which increase the computational burden.

(3) Cross-domain authentication scheme based on blockchain

Considering the interoperability and complexity characteristics of IoD systems, many existing research studies have applied blockchain technology in the cross-domain authentication area of IoT systems. To solve the single point of failure problem, Feng et al. [31] employed blockchain and multiple signatures based on threshold sharing to build a cross-domain authentication framework. To avoid reliance on a trusted third party, Shen et al. [32] introduced a consortium blockchain to construct trust among different domains and present an efficient blockchain-assisted secure device authentication mechanism. By using a hierarchy of local and global smart contracts, Gauhar et al. [33] proposed a decentralized blockchain-based authentication mechanism which uses a proof-of-authenticity mechanism to find and retrieve device hashes stored in local blockchain. Zhang et al. [34] proposed a thoroughly cross-domain authentication scheme based on blockchain, allowing participants from different domains with different settings to complete the authentication. However, the above methods neglect the anonymity of drones when performing the task, which may leak the user's privacy. Additionally, the methods do not take into account the communication and computational costs of IoD when facing a large number of drones.

3. Overview of BCDAIoD

3.1. The BCDAIoD Framework

To improve the efficiency of data query, BCDAIoD uses a multi-chain architecture to reasonably partition the data storage on the consortium blockchain. Additionally, domain private chains are used to ensure the anonymity of drones. The framework of the proposed BCDAIoD scheme is shown in Figure 2. The BCDAIoD framework includes four layers: an application layer, service support layer, data storage layer, and network layer.

In the network layer, a P2P network is used for communication between consortium nodes (*CN*s) and UAVs. Additionally, *CN*s can transmit information through the P2P network, such as mission information. Each *CN* manages a certain domain and maintains a private chain. The UAVs can also communicate with each other through the P2P network.

In the data storage layer, the UAV device information, task information, address information of *CN*s, smart contracts, and user registration information are stored in the consortium blockchain (*CBC*) in a specified format. At the same time, the *CN*s store the details of the above information in local databases. The architecture of the consortium blockchain is described in Section 3.2. For identity authentication and UAV grouping, the *CN*s also maintain a private chain to store the device ID on *PBC* (*Pid*) and the out-of-domain range (*Rout*) information of UAVs.

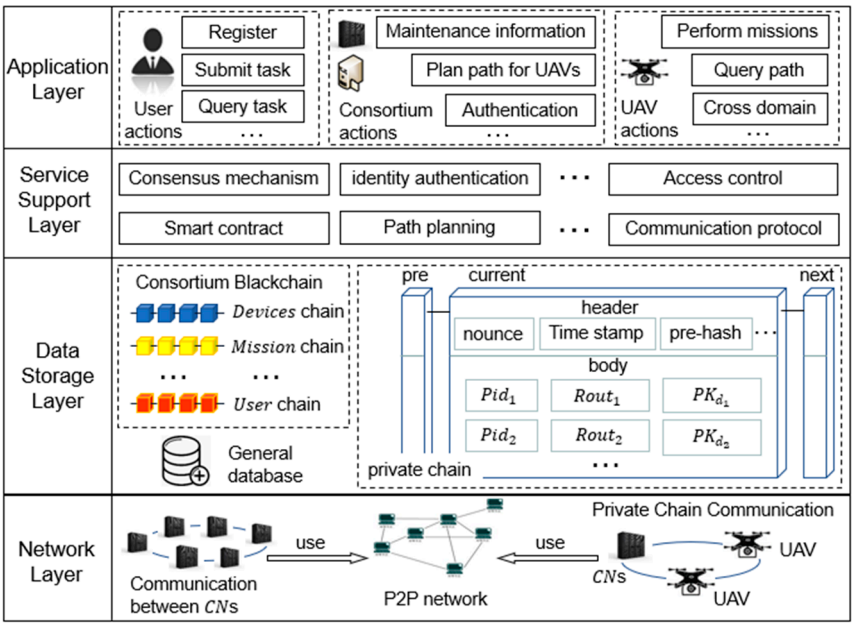

Figure 2. The BCDAIoD framework.

The service support layer provides support for users, *CN*s, and UAVs to interact with the data storage layer and the network layer. It mainly includes the consensus mechanism, smart contract, identity authentication mechanism, access control strategy, path planning algorithm, and communication protocol. The consensus mechanism can ensure that the information of each block is consistent. The smart contract can conduct trusted transactions in the form of commitment without a trusted third party. The identity authentication mechanism and access control strategy can ensure that UAVs enter the correct domains and obtain their own path information. By using the path planning algorithm, the *CN*s obtain the next domain IDs and calculate the paths in their domains for UAVs. The communication protocol supports the communication among users, UAVs, and *CN*s.

By using the functions of the application layer, users can submit registration applications to the *CN*s, release tasks, and query the status of tasks. The *CN*s can manage the information of the UAV devices, tasks, and paths, as well as perform identity authentication and path planning. The UAVs can query task information, view path information, and submit cross-domain authentication requests to perform tasks. To facilitate the introduction of subsequent methods, Table 1 lists the key symbols and their definitions.

Table 1. The symbol definitions.

Symbol	Definition	Symbol	Definition
CBC	Consortium blockchain	PBC	Private blockchain
CN	Consortium node	$CNid$	ID of consortium node
Pid_i	Device ID of i on PBC	Did_i	Device ID of i on CBC
Mid	Task ID	$GList$	Group member list
$Rout$	Out-of-domain range	$Tout$	Expected time out of domain
d_i	Drone marked with i	PK_i	Public key of i
SK_i	Private key of i	$Enc(*)$	Elliptic curve encryption
PBH_i	Block number of the PBC block that includes the task information of i	ks	Session key
$Token$	Cross-domain token	$H(*)$	Hash function

Table 1. *Cont.*

Symbol	Definition	Symbol	Definition
T_{stamp}	Time stamp of the *Token*	M_{d_j}	ID verification information
ack	Challenge	rsp_i	Response of i

3.2. Architecture of Consortium Blockchain

The scale of data to be processed by the *CN*s is very large and there are various types of data. All the data being stored in a single chain leads to low query efficiency, poor data isolation, and difficulty to expand. Therefore, we designed the multi-chain architecture of the *CBC* to improve the efficiency of the *CN*s. As shown in Figure 3, the multi-chain architecture of the *CBC* mainly includes the *Mission* chain, *User* chain, *Address* chain, *Devices* chain, and *Contract* chain. The *CN*s are also responsible for the private blockchain (*PBC*) in their domain. Similar to cellular mobile communication networks, the proposed framework achieves service coverage in large-scale open areas by combining *CN*s. Additionally, there are cross-domain channels between the domains, so as to facilitate the formation of groups and the cross-domain authentication of UAVs with the same *Rout*.

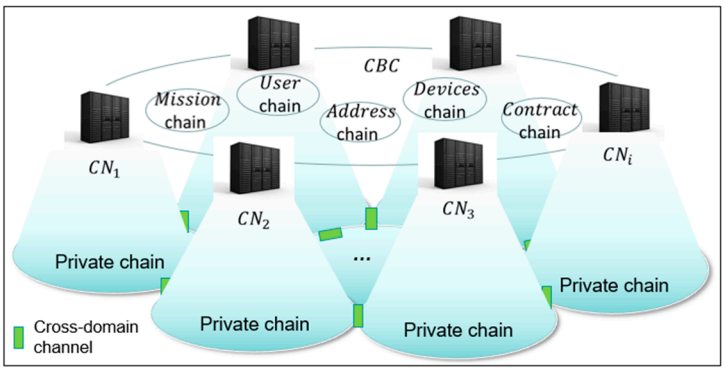

Figure 3. The multi-chain architecture of CBC.

The *Mission* chain mainly stores task information, and the data structure can be expressed as Equation (1), where *Mid* is the current task ID; *Did* is the drone's ID on the *CBC* that can be provided to or queried by the *CN*s; $CNid_n$ is the ID of the *CN* that is responsible for the next domain; $CNid_d$ is the ID of the *CN* that manages the destination domain; $CNid_c$ is the *CN* ID of the current domain; *Rout* is the expected range out of the current domain; and *Tout* represents the current expected time out of the domain.

$$Mission = (Mid, Did, CNid_n, CNid_d, CNid_c, Rout, Tout) \quad (1)$$

The *User* chain mainly stores the necessary information of the registered users. The stored information can be expressed as Equation (2), where *Uid* is the user ID for the consortium authentication, $CNid_r$ is the *CN* ID of the registration place, and *Fu* is the current user account balance. The other registration details of the user are directly stored in the local database of the registration place's *CN*.

$$User = (Uid, CNid_r, Fu) \quad (2)$$

The *Address* chain mainly stores the domain range information of the *CN*, which can be expressed as Equation (3), where *CNid* represents the ID of a *CN*; PK_{cn} represents the public key of the *CN*; and RG_{cn} represents the domain range of the *CN*.

$$Address = (CNid, PK_{cn}, RG_{cn}) \quad (3)$$

The *Devices* chain mainly stores the necessary information of the UAV registered in the *CBC*, which can be expressed as Equation (4), where PK_d is the public key of a UAV; *Mod* is the module of the UAV; and *Pol* is the execution strategy of the UAV. The registration details of other UAVs can be directly stored in the local databases of the registration place's *CN*.

$$Devices = (Did, PK_d, CNid_r, Mod,\ Pol) \tag{4}$$

The *Contract* chain mainly stores the contract information, which can be expressed as Equation (5), where *Cid* is the current contract ID, *V* is the current contract version, and *Cont* is the current contract content.

$$Contract = (Cid, V, Cont) \tag{5}$$

The *PBC* chain mainly stores the necessary information for the intradomain authentication of the UAVs, which can be expressed as Equation (6), where *Pid* represents the temporary ID of a UAV in a domain.

$$PBC = (Pid, Rout, PK_d) \tag{6}$$

3.3. Mission Model of Drone

We designed a scenario of a drone delivery in an Internet of Drones environment, as shown in Figure 4. Each domain *CN* in the consortium blockchain can be regarded as a server with strong computing power and storage capacity, which is responsible for drone management and scheduling in a certain location area. The drone delivery process mainly includes three phases: registration, task release, and task execution.

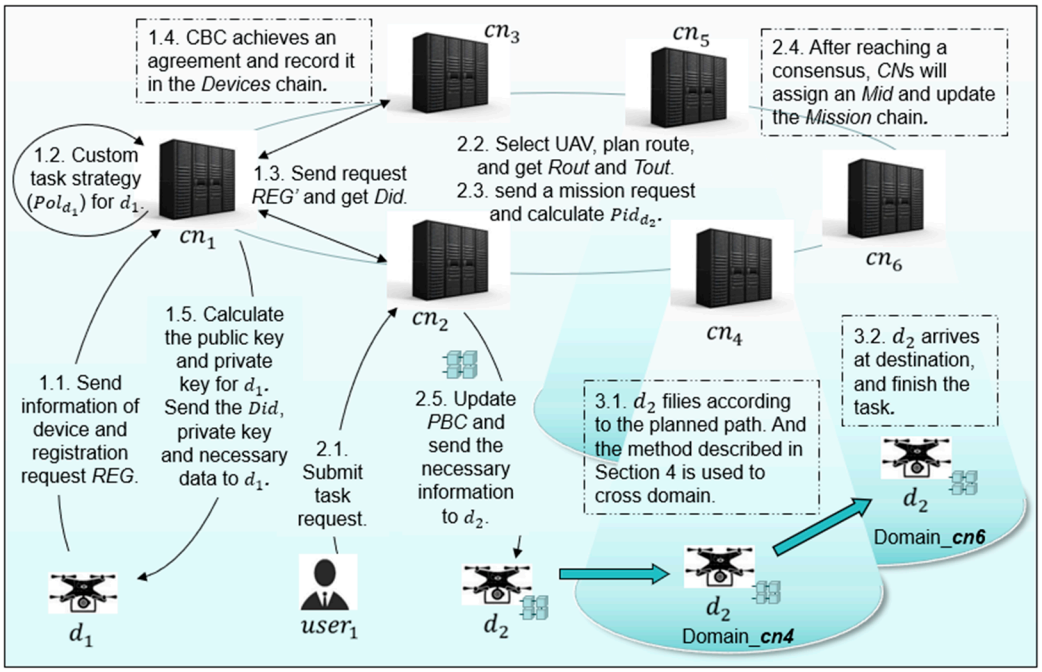

Figure 4. Scenario of drone delivery.

3.3.1. Registration Mechanism

In the proposed framework, both drones and users need to register in the *CBC*. The drone registration process includes the following steps. The user registration process is similar to the drone registration process, which is not described in detail here.

Step 1. Drone d_j submits a registration request $REG = \left(Rrequest,\ mac_{d_j},\ inf_{d_j}\right)$ to cn_i, where *Rrequest* denotes registration request, mac_{d_j} denotes the hardware ID of d_j, and inf_{d_j}

denotes the detailed information of d_j, such as drone model, maximum sailing distance, and payload.

Step 2. After receiving the registration request, cn_i determines the acceptable task type of d_j (Pol_{d_j}), such as selecting tasks with reasonable distance according to power consumption.

Step 3. Then, cn_i sends the registration request $REG' = \left(Rrequest,\ mac_{d_j}, CNid_i, Pol_{d_j}\right)$ to the other nodes in the CBC.

Step 4. After the members of the CBC reach a consensus, the CNs assign a device ID in the CBC (Did_{d_j}) to the d_j and update the *Devices* chain. Furthermore, cn_i calculates the public key (PK_{d_j}) and private key (SK_{d_j}) for d_j. Then, cn_i sends the SK_{d_j} to d_j.

Step 5. cn_i sends the Did_{d_j} and its public key PK_{cn_i} to d_j. Additionally, cn_i calculates $Enc_{SK_{cn_i}}\left(Did_{d_j}\right)$ and sends it to d_j. Then, cn_i stores the registration time, model, and other detailed information of d_j in the local database.

3.3.2. Task Release Mechanism

When a user submits a delivery task request, the CN in charge of the current domain selects a suitable drone to perform the task. The process includes the following steps.

Step 1. When a user submits a delivery task request to the cn_i, cn_i queries the available drone from the local database, selects an appropriate drone d_k according to the acceptable task type, and plans a delivery path for d_k. Additionally, cn_i calculates $Rout$ and $Tout$.

Step 2. At the same time, cn_i queries the *Address* chain to find the ID of the destination domain ($CNid_d$) and the next domain ($CNid_n$) that d_k will pass through.

Step 3. Then, cn_i sends a mission request $mission = (Did, CNid_n,\ CNid_d,\ CNid_c, Rout, Tout)$ to the other CNs in the CBC and calculates the ID on the PBC (Pid_{d_k}) for d_k. Then, cn_i uses SK_{cn_i} to generate $Enc_{SK_{cn_i}}\left(Pid_{d_k}\right)$.

Step 4. After the members of the CBC reach a consensus, the CNs assign the mission ID (Mid) and update the *Mission* chain with $mission' = (Mid,\ Did,\ CNid_n,\ CNid_d, CNid_c, Rout, Tout)$.

Step 5. Then, cn_i updates the PBC, and sends the necessary information to the drone d_k. The information is shown in Table 2. Then, d_k starts the task. Correspondingly, d_k updates its own PBC during the execution of the task.

Table 2. The symbol definitions.

Symbol	Definition
Flight path	The flight path in the current domain.
Pid_{d_k}	Device ID of d_k on the PBC.
SK_{d_k}	Private key of d_k.
PK_{cn_i}	Public key of cn_i for intradomain authentication between drones.
PK_{cn_n}	Public key of cn_n(the $CN$of the next domain) for cross-domain authentication.
$Enc_{SK_{cn_i}}\left(Did_{d_k}\right)$	The identification for cross-domain authentication.
$Enc_{SK_{cn_i}}\left(Pid_{d_k}\right)$	The identification for intradomain authentication.
PBC	The private blockchain of the current domain.
PBH_{d_k}	The block number of the PBC block that includes the task information of d_k.

Step 6. Finally, the cn_i stores the detailed information in the local database and removes the drone d_k from the available drones of the local database.

Furthermore, we make the following assumptions:

(1) We consider that each CN has the public keys of the other CNs, and the private key of each CN is not leaked; (2) The private key of the drone carrying out the task is not

leaked; (3) Attackers cannot deduce the private key from the public key, or it takes too much time.

3.3.3. Task Execution Method

After the task starts, the drone d_k flies to the destination according to the planned path. If d_k needs to pass through other domains, the method described in Section 4 is used for cross-domain authentication. After successful cross-domain authentication, d_k enters the next domain and receives the necessary information from the CN, in a similar method to Step 5 in the task release phase. In this way, d_k flies to the destination domain.

When d_k reaches the destination domain and completes the task, the CN in charge of the destination domain (cn_d) publishes a task completion confirmation, $mission_F = (Mid,\ Did,\ 0,\ CNid_d, CNid_d, 0, 0)$, in the *Mission* chain to prove that the task has been completed.

Finally, the cn_d is responsible for recycling the drone d_k. By querying the Did of d_k and other information in the *Devices* chain, cn_d stores the necessary information of d_k in the local database and updates the available device database.

4. Cross-Domain Authentication of Drones

Drones may fly to other domains during the execution of tasks. Therefore, we designed a UAV cross-domain authentication method combining public key infrastructure and blockchain technology. In this section, we first propose a single-drone cross-domain authentication method. Then, considering that the cross-domain authentication of a large number of UAVs may cause a long waiting time for UAVs, we propose an establishment mechanism of UAV groups and a drone group cross-domain authentication method. Additionally, we propose a notification mechanism between domains to let the CNs prepare for UAV cross-domain and path planning in advance.

4.1. Single-Drone Cross-Domain Authentication Method

The single-drone cross-domain authentication method is shown in Figure 5. This method uses public key infrastructure and a challenge–response mechanism to ensure the authenticity of the UAV's identity, and ensures the authenticity of the UAV's task by querying the *Mission* chain of the CBC. At the same time, a session key can also be generated during this process.

When drone d_j wants to fly to the next domain ($Domain_n$), d_j firstly send its cross-domain request (CRM_{d_j}) to the CN in charge of $Domain_n$ (cn_n). The CRM_{d_j} can be expressed as Equation (7), where $Crequest$ represents the cross-domain request and $Enc_{SK_{cn_i}}\left(Did_{d_j}\right)$ represents the Did of d_j encrypted with the private key of the current domain CN.

$$CRM_{d_j} = \left(Crequest, Enc_{SK_{cn_i}}\left(Did_{d_j}\right)\right) \tag{7}$$

After receiving the CRM_{d_j} from d_j, cn_n uses the PK_{cn_i} to decrypt the $Enc_{SK_{cni}}\left(Did_{d_j}\right)$ to obtain the Did_{d_j}. Then, cn_n checks whether there is the incomplete *Mission* chain transaction $trans^*$ of the corresponding Did_{d_j}. The $trans^*$ can be generated through the notification mechanism (detailed in Section 4.4). Then, cn_n searches the *Devices* chain to obtain the public key (PK_{dj}) of d_j according to the Did_{d_j}. If $trans^*_{d_j}$ exists, cn_n sends a random number x with PK_{d_j} encryption to d_j as a challenge ack. Then, d_j decrypts ack with SK_{d_j} to obtain x, and sends back $x+1$ and a random number y with PK_{cn_n} encryption as a response, $rsp_j = Enc_{PK_{cnn}}(x+1||\ y)$. Then, cn_n checks rsp_j to determine whether d_j has the declared identity and obtains y. If d_j passes the verification, cn_n sends back a response, $rsp_{cn} = Enc_{PK_{d_j}}(y+1)$, as a confirmation message. Then, the session key between d_j and cn_n can be generated by $ks = H(x\ ||\ y)$.

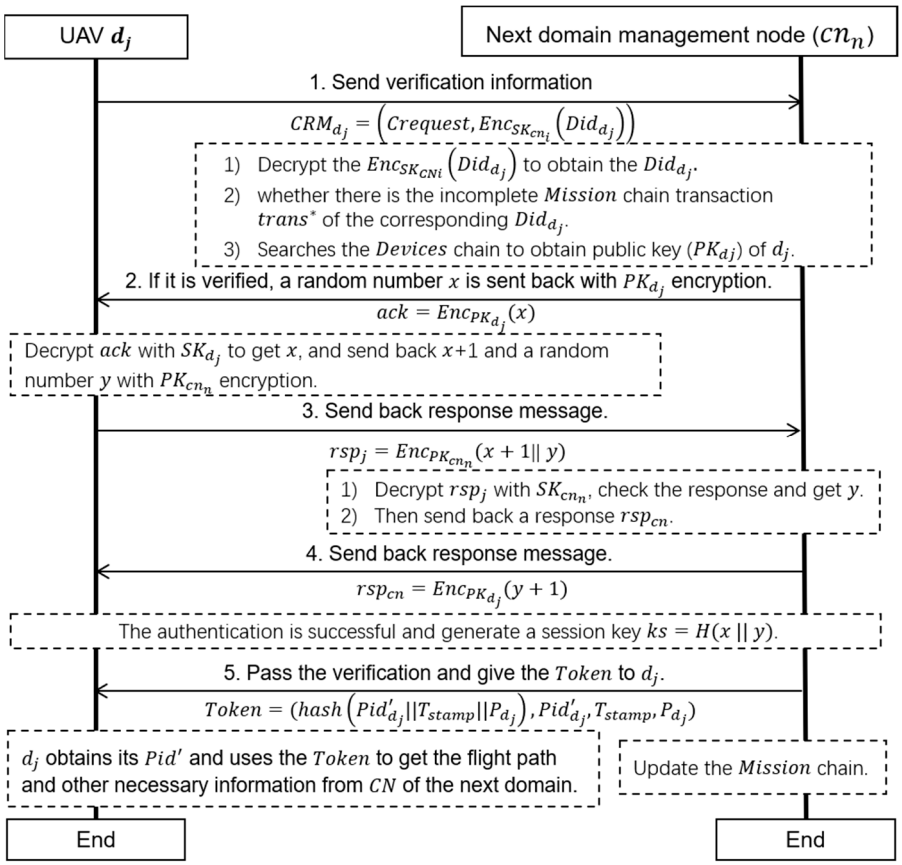

Figure 5. The single-drone cross-domain authentication method.

In this way, cn_n can determine the identity and task information of d_j. Then, cn_n sends the *Token* to d_j. The *Token* of d_j can be expressed as Equation (8), where, Pid'_{d_j} represents the device ID of d_j in the $Domain_n$, T_{stamp} represents the time stamp of the *Token*, P_{d_j} represents the permission that d_j has for obtaining the necessary information, and $hash\left(Pid'_{d_j}||T_{stamp}||P_{d_j}\right)$ represents the hash value of the combination of Pid'_{d_j}, T_{stamp}, and P_{d_j}.

$$Token = \left(hash\left(Pid'_{d_j}||T_{stamp}||P_{d_j}\right), Pid'_{d_j}, T_{stamp}, P_{d_j}\right) \quad (8)$$

Finally, d_j obtains its Pid'_{d_j} and uses the *Token* to obtain the flight path and other necessary information from the *CN* of the next domain. At the same time, cn_n updates the *Mission* chain by using the method proposed in Section 4.4.

In the UAV cargo transportation scenario, there are many UAVs flying to the same next domain. Therefore, we propose a method of drone group cross-domain authentication to improve the efficiency of UAV cross-domain authentication. The idea of this method is that UAVs compose a group through mutual authentication before the cross-domain authentication. In this way, the proposed method can lighten the authentication workload of the *CN* and improve the speed of the UAV cross-domain authentication. The method is mainly divided into two stages: (1) the formation of a UAV group (in Section 4.2), and (2) the cross-domain authentication of a UAV group (in Section 4.3).

4.2. Establishment Mechanism of UAV Groups

The establishment mechanism of UAV groups mainly includes two parts: (1) the method of building a new drone group and (2) the method of joining a drone group. In the process of building or joining a drone group, drones need to use verification strategies to verify each other. The verification strategy is shown in Figure 6.

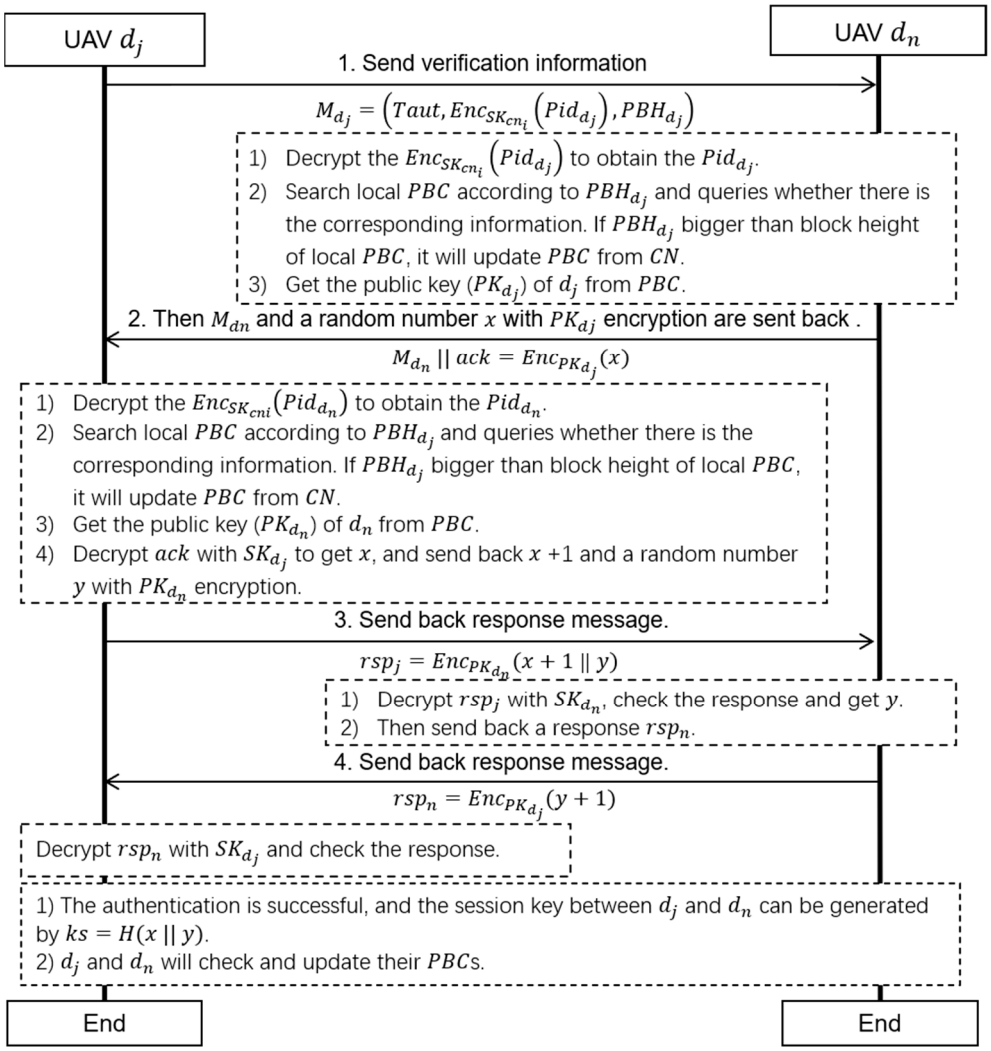

Figure 6. Two-way authentication strategy between drones.

When drone d_j and drone d_n start the verification, d_j firstly sends its verification information M_{d_j} to d_n. M_{d_j} can be expressed as Equation (9), where, $Taut$ represents the two-way authentication request, $Enc_{SK_{cn_i}}\left(Pid_{d_j}\right)$ represents the Pid of d_j encrypted with the private key of the current domain CN, and PBH_{d_j} represents the PBC block height of the block that includes the task information of d_j.

$$M_{d_j} = \left(Taut, Enc_{SK_{cn_i}}\left(Pid_{d_j}\right), PBH_{d_j}\right) \tag{9}$$

After receiving the verification information from d_j, d_n uses the PK_{cn_i} to decrypt the $Enc_{SK_{cn_i}}\left(Pid_{d_j}\right)$ to obtain the Pid_{d_j}. Then, d_n searches the local PBC according to PBH_{d_j} and queries whether the corresponding information is there. If PBH_{d_j} is bigger than the block height of the local PBC, it updates the PBC from the CN. After that, d_n obtains the public key (PK_{d_j}) of d_j from the PBC. Then, a random number x is encrypted by PK_{d_j} as a challenge ack, and d_n sends its M_{d_n} and ack to d_j.

After receiving the M_{d_n} and ack, d_j decrypts the $Enc_{SK_{cni}}(Pid_{d_n})$ to obtain the Pid_{d_n}. Additionally, d_j searches the local PBC according to PBH_{d_j} and queries whether the corresponding information is there. If PBH_{d_j} is bigger than the block height of the local PBC, it updates the PBC from the CN. Then, d_j obtains the public key (PK_{d_n}) of d_n from the PBC. Additionally, d_j decrypts the ack with SK_{d_j} to obtain x, and sends back $x + 1$ and a random

number y with PK_{d_n} encryption as a response, $rsp_j = Enc_{PK_{d_n}}(x+1||y)$. According to the notification mechanism between domains described in Section 4.4, local *PBC*s saved by drones store task information for a period of time in the future, and the *PBC*s can be updated by drones after mutual authentication. Therefore, in theory, drones do not need or rarely need to update blocks through the *CN*, and they only need to update blocks through the *CN* at most once during a two-way authentication period.

Next, d_n decrypts rsp_j with SK_{d_n} and checks whether $x+1$ is received within a certain period of time to determine whether d_j has the declared identity. Then, d_n obtains y and sends back a response, $rsp_n = Enc_{PK_{d_j}}(y+1)$, to d_j. After receiving the rsp_n, d_j decrypts the rsp_n with SK_{d_j} and checks the response. After successful identity authentication, the session key between d_j and d_n can be generated by $ks = H(x \,||\, y)$. Then, d_j and d_n can communicate with each other and update their *PBC*s.

By using the proposed verification strategy, drones can confirm each other's identity, generate session keys, and update the *PBC*s. In the process of moving, drones try to join a group or build a new one, as shown in Figure 7.

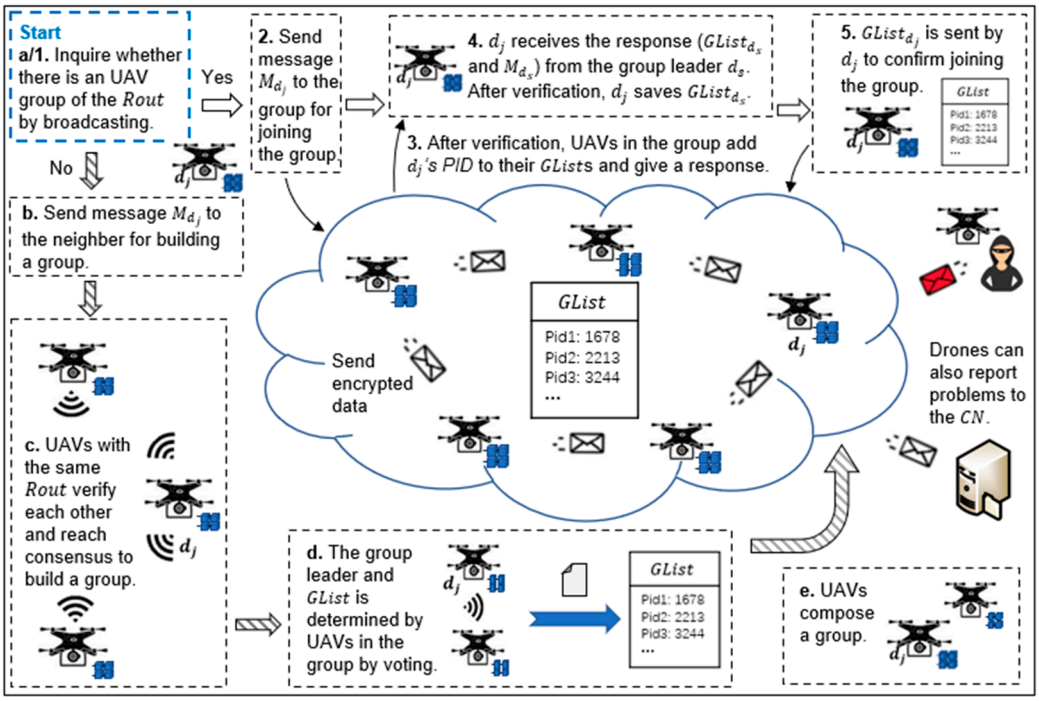

Figure 7. The establishment method of UAV groups.

(1) Method of building a new drone group

During flight in the current domain, a drone d_j broadcasts to inquire whether there is a drone group with the same *Rout*. If there is no drone group, d_j tries to build a new group and continues to broadcast to inquire whether there are drones with the same *Rout*. If d_j finds other drones with the same *Rout*, the drones and d_j send each other verification information, verify each other (as shown in Figure 6), and then reach consensus to build a group. Usually, the number of initial group members is small (about 2–4 drones). In order to stabilize the drone group, the initial members need to authenticate each other and build communication links with a session key.

Each drone group selects a group leader d_l by voting. For cross-domain authentication, d_l generates a member list, *GList*, of the drone group. The *GList* can be expressed as Equation (10), where Pid_{d_i} represents the *Pid* of d_i. Then, the drones compose a drone group successfully.

$$GList = \left\{ Pid_{d_i} \mid \forall\, d_i \text{ in the group} \right\} \tag{10}$$

(2) Method of joining a drone group

If d_j finds a group after broadcasting, it sends verification information M_{d_j} to the group leader (d_l) to join the group. Then, d_l randomly chooses one drone to verify the d_j together. After that, d_l determines whether d_j can join the group according to the authentication result. If d_j passes the verification, d_l adds the *Pid* and public key of d_j to the $GList_{d_l}$ maintained by itself, and send its $GList_{d_l}$ to d_j. Then, d_j saves the $GList_{d_l}$ and broadcasts its own $GList_{d_j}$ as a confirmation of joining the group. Additionally, the other drones in the group update their *GList*.

At the same time, in order to prevent the loss of drones, the drones in the group send inquiry and response signals regularly. Additionally, drones with forged identities cannot build or join a group because they cannot send the correct response.

4.3. Drone Group Cross-Domain Authentication Method

The drone group cross-domain authentication method proposed in this paper is shown in Figure 8. When a drone group is ready for cross-domain authentication, the group leader of the group (d_l) sends a group cross-domain request (GCM_{d_l}) to the *CN* of the next domain (cn_n). The GCM_{d_l} sent by d_l can be expressed as Equation (11), where *GCrequest* represents the group cross-domain request, $Enc_{SK_{cn_i}}(Did_{d_l})$ represents the *Did* of d_l encrypted with the private key of the current domain *CN* (cn_i), and $GList_{d_l}$ is the group member list of d_l.

$$GCM_{d_l} = \left(GCrequest,\ Enc_{SK_{cn_i}}(Did_{d_l}), GList_{d_l}\right) \tag{11}$$

After receiving the GCM_{d_l} from d_l, cn_n verifies the group leader through the single-drone cross-domain authentication method proposed in Section 4.1. Then, cn_n obtains the $GList_{d_l}$. If d_l passes validation, cn_n sends d_l a *Token*. After receiving the *Token*, d_l sends a group cross-domain signal to the drone group. Then, the other drones in the group send group cross-domain requests to cn_n. The group cross-domain request sent by d_j (GCM_{d_j}) can be expressed as Equation (12), where $Enc_{PK_{cn_n}}\left(Pid_{d_j}, x_j\right)$ represents the device ID on the PBC and a random number x_j encrypted with the public key of cn_n. Additionally, $Enc_{PK_{cn_n}}\left(Pid_{d_j}, x_j\right)$ is generated by d_j after d_j joins or builds a group.

$$GCM_{d_j} = \left(GCrequest, Enc_{PK_{cn_n}}\left(Pid_{d_j}, x_j\right), Enc_{SK_{cn_i}}\left(Did_{d_j}\right)\right) \tag{12}$$

After receiving the GCM_{d_j}, cn_n decrypts the $Enc_{SK_{cn_i}}\left(Did_{d_j}\right)$ to obtain the *Did* of d_j. Additionally, cn_n checks whether the incomplete *Mission* chain transaction $trans^*$ of the corresponding Did_{d_j} is there. Then, cn_n searches the *Devices* chain to obtain the public key of d_j. Additionally, cn_n decrypts the $Enc_{PK_{cn_n}}\left(Pid_{d_j}, x_j\right)$ to obtain the *Pid* and x_j of d_j. If Pid_{d_j} is in the $GList_{d_l}$, cn_n generates the *Token* for d_j according to the equivalence between *Pid* and *Did*. Next, cn_n sends a response, $rsp^j_{cn} = Enc_{PK_{d_j}}(y_j)$, to d_j. After decrypting rsp^j_{cn} and obtaining y_j, d_j can generate a session key by $ks_j = H(x_j \,||\, y_j)$.

In this way, cn_n verifies the drones and distributes the *Token*s to the drones in the group. Finally, the drones in the group obtain their *Pid*s and use their *Token*s to obtain the flight path and other necessary information from cn_n. At the same time, cn_n updates the *Mission* chain by using the method proposed in Section 4.4.

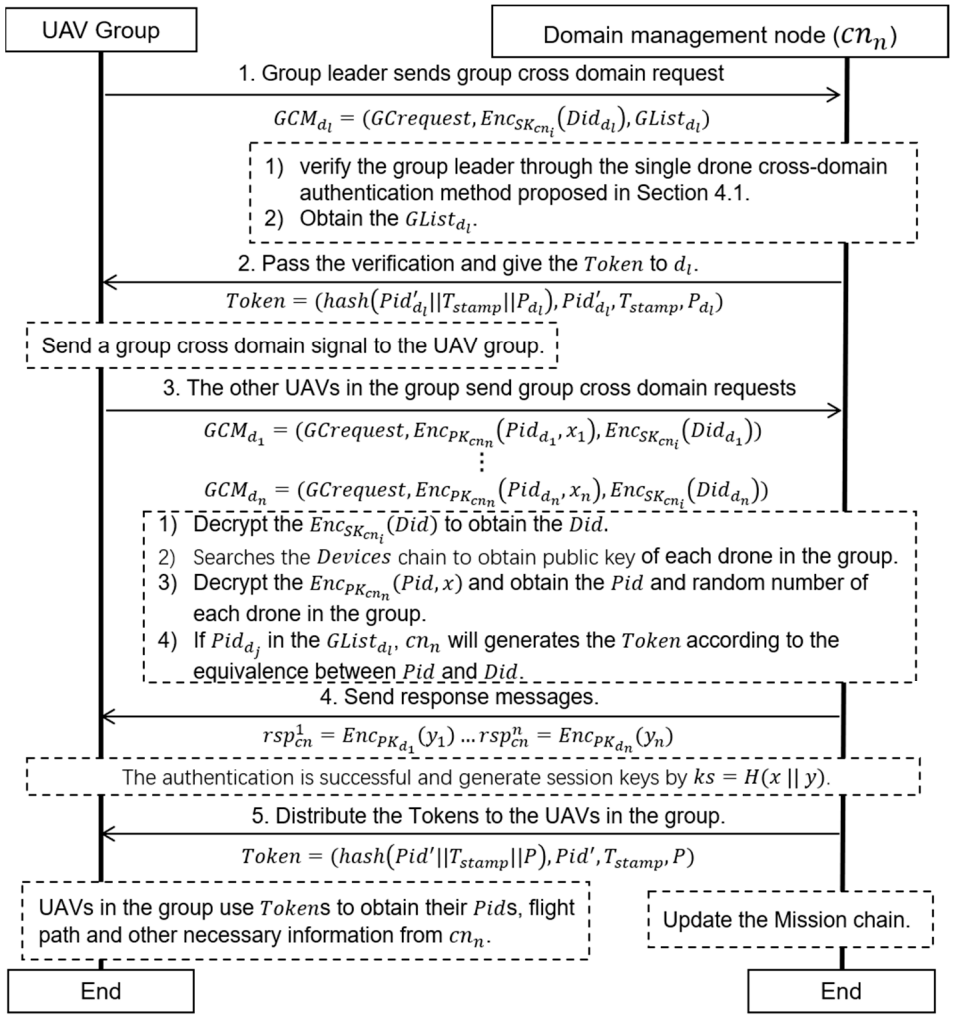

Figure 8. Drone group cross-domain authentication strategy.

4.4. Notification Mechanism between Domains

When a drone enters a new domain, the *CN* of the domain needs to publish a transaction to the *Mission* chain for uploading and updating the task information of the drone. After reaching a consensus, a *CN* packages a certain number of transactions and generates a new block on the *Mission* chain. The *CN*s of the other domains query the *Mission* chain at regular intervals and collect the task information of drones flying to their own domains. In this way, the *CN*s can plan the path for the drones in advance according to the *Rout*, the destination, and other information of the task. The specific method is as follows.

In the domain $Domain_i$, the *CN* of $Domain_i$ (cn_i) uses Algorithm 1 to find the transactions of the next domain ($CNid_n$), which is $Domain_i$, at regular intervals. Firstly, the algorithm obtains the latest block height of the *Mission* chain. Then, it searches blocks that have not been queried to obtain the transactions that include the latest drone task information. By comparing the $CNid_n$ in the transaction ($trans.CNid_n$) and the *CNid* of cn_i ($cn_i.CNid$), the algorithm determines whether the next domain in the transaction is the current domain, and then saves the task information. Then, cn_i obtains the list of transactions ($List_{trans}$) and the latest block height (NH_{bc}).

Algorithm 1. Task information query algorithm.

Input: Mission, LH_{bc} // Mission chain and the block height of the last query.
Output: $List_{trans}$, NH_{bc} // List of transactions and the latest block height.
1. Initialize variable NH_{bc} // Initialize the latest block height.
2. Initialize variable $List_{trans}$ // Initialize the list of transactions.
3. NH_{bc} = getHeight(*Mission*) // Obtain the latest block height of the *Mission* chain.
4. for *block* in range(LH_{bc},NH_{bc}) // Search blocks that have not been queried.
5. *trans* = read(*block*) // Read the transactions on the blocks.
6. if($trans.CNid_n$ = = $cn_i.CNid$) // Determine whether the next domain in the *trans* is the current domain.
7. $List_{trans}$.add(*trans*) // Save the task information.
8. else continue
9. end if
10. end for
11. return $List_{trans}$, NH_{bc}

When it has obtained the $List_{trans}$ and the relevant task information, cn_i calls on Algorithm 2 to preprocess the tasks. For each transaction in the $List_{trans}$, the algorithm receives Mid, Did, $CNid_d$, $Rout$, and $Tout$ from the transaction. Then, it calls on the path planning algorithm to plan a flight path for the drone and obtain the CN ID of the next domain ($CNid_n^{'}$), the range out of the current domain ($Rout^{'}$), and the current expected time cost out of the domain (TC^*). For drone cross-domain authentication, the algorithm reads the *Address* chain and obtains the public key (PK_{cn_n}) of $CNid_n^{'}$. Then, it reads the *Devices* chain and obtain the PK_d of the drone. Additionally, it generates the device ID in this domain (Pid') and the permission (P) for the drone. Then, the algorithm submits the transaction (Pid', $Rout^{'}$, PK_d) to the PBC. In this way, the PBC of the drone currently flying to the next domain has the information about the drones performing tasks in that domain for a period of time in the future. Additionally, it generates an incomplete *Mission* chain transaction, $trans^* = \left(Mid,\ Did,\ CNid_n^{'},\ CNid_d,\ CNid_c^{'},\ Rout^{'},\ TC^*\right)$, and the TC^* in $trans^*$ is updated when the drone arrives. At the same time, cn_i packages the PBC transactions and generates a new block on the PBC at certain intervals, or the number of transactions meets the requirement.

Algorithm 2. Task preprocessing algorithm.

Input: $List_{trans}$
Output: $trans^*$, Pid', P, PK_{cnn}
1. Initialize variable $Rout'$, TC^* // Initialization range and expected time cost out of the current domain.
2. Initialize variable $CNid_n^{'}$ // Initialization variable $CNid$ of the next domain.
3. Initialize variable $CNid_c^{'} = cn_i.CNid$ // Initialization variable $CNid$ of the current domain.
4. for each transaction in $List_{trans}$
5. Get $Mid, Did, CNid_d, Rout, Tout$ from the transaction.
6. Use path planning algorism to plan flight path for the drone and get $CNid_n^{'}$, $Rout_c^{'}$, and TC^*.
7. Read *Address* chain and get the PK_{cnn} of $CNid_n^{'}$.
8. Read *Devices* chain and get the PK_d of the drone.
9. Create Pid' and P // Generate device ID in this domain and the permission for the drone.
10. submit(Pid', $Rout'$, PK_d) ->PBC // Submit the PBC transaction to the PBC.
11. $trans^* = (Mid, Did, CNid_n^{'}, CNid_d, CNid_c^{'}, Rout', TC^*)$ // Generate the *Mission* chain transaction.
12. end for
13. cn_i packages PBC transactions and generates a new block on the PBC at certain intervals or the number of transactions meets the requirement.
14. return $trans^*$, Pid', P, PK_{cn_n}.

When a drone d_j has passed the cross-domain authentication and entered the domain $Domain_i$, cn_i uses Algorithm 3 to publish a transaction for updating the task information of

d_j. Above all, cn_i obtains the task start time (TST_{dj}) of d_j in the domain. Then, cn_i calculates the expected time out of the domain by $Tout'_{dj} = TST_{dj} + TC^*$. After that, the *Mission* chain transaction including the task information of d_j is published, which can be expressed as $trans = \left(Mid,\ Did_{dj},\ CNid'_n,\ CNid_d,\ CNid'_c,\ Rout'_{dj},\ Tout'_{dj}\right)$. We consider that all the *CN*s in the *CBC* are trusted. Therefore, this paper uses the Raft consensus mechanism to package the transactions and generate new blocks. In addition, the Pid'_{dj}, P_{dj}, and PK_{cn_n} generated in Algorithm 2 are sent to the d_j during the cross-domain authentication process.

Algorithm 3. Task information update algorithm.

Input: Mission information
Output: True or False
1. Initialise variable isUploaded = FALSE // Initialization variable, upload success or not.
2. Get task start time (TST_{d_j}) of d_j in the domain.
3. Compute $Tout'_{d_j} = TST_{d_j} + TC^*$ // Calculate the expected time out of the domain.
4. $trans = \left(Mid,\ Did_{d_j},\ CNid'_n,\ CNid_d,\ CNid'_c,\ Rout'_{d_j},\ Tout'_{d_j}\right)$ // Generate transaction including task information of d_j.
5. Publish the *trans* to the *Mission* chain.
6. isUploaded = TRUE // Upload successfully.
7. return isUploaded.

5. Performance Evaluation

5.1. Experimental Settings

We analyzed the performance of the proposed scheme by conducting simulation experiments. The performance of the method proposed in this paper was measured in terms of computational overhead, communication overhead, and cross-domain authentication time cost. The configuration of the PC for the experiments is: CPU: Intel Core i7-8550, RAM: 8 GB, OS: Ubuntu 18.04, 64-bit. Hyperledger Fabric is an open source project from the Linux Foundation. We used Hyperledger Fabric v1.4 to build the blockchain, and the consensus on the consortium blockchain was reached through the Raft algorithm. Additionally, we used the JPBC v2.0 bilinear pair cryptography library from Italy GAS Lab to generate the public and private keys, and to encrypt and decrypt messages and ciphertext, respectively. The applied elliptic curve is a Type A elliptic curve with an order length of 160 bits ($y^2 = x^3 + x$). Raspberry Pi, as an embedded single-board computer (SBC) from Uk Raspberry Pi Foundation that is easy to use for coding and other implementations, is widely used in the existing studies. To further evaluate the feasibility of the proposed scheme, we used Raspberry Pi 4B SBCs to simulate the drones. The configuration of the Raspberry Pi 4B is: CPU: Quad-core Cortex-A72, RAM: 8 GB, OS: Ubuntu 18.04, 64-bit. We also compared the proposed method with existing methods [25–27] that use a ground station as a trusted third party for identity authentication, as well as existing methods [31–33] for identity authentication through ground stations and blockchain architectures.

5.2. Computational Overhead

5.2.1. Materials and Methods

To evaluate the computational overhead of the proposed framework, we analyzed the computational operations required by each entity in different phases of tasks. Simple operations, such as integer addition and concatenation operation, were not taken into consideration because of their low computational expense. Specific notations are listed as follows. CN: A consortium node in charge of a domain; d_j: A drone in a drone group or a single drone; d_l: The group leader of a drone group or a single drone; RG: Registration mechanism; TR: Task-release mechanism; SC: Single-drone cross-domain authentication method; TA: Two-way authentication strategy; NG: Method of building a new group; JG: Method of joining a drone group; DGC: The drone group cross-domain authentication

method; NMD: Notification mechanism between domains; DAT: Operation of determining the acceptable task type of a drone; BC: One reading or writing operation on blockchain; GKP: Operation of generating a public–private key pair for d_j; CAS: One asymmetric encrypt/decrypt operation; LD: One reading or writing operation on local database; PP: One path planning operation; HO: One hash operation; N: The number of drones to compose a group or in a group; and GL: Operation of generating, updating, or distributing a group member list.

Table 3 shows the computational overhead that each entity needs to undertake in different task model phases. For example, in the process of DGC, the total computational cost that a *CN* needs to undertake is 4CAS + BC + HO + (N − 1) × (3CAS + HO). Specifically, it denotes the total overhead of performing four asymmetric encrypt/decrypt operations, one blockchain operation, one hash operation, and N – 1 times (3CAS + HO). The function 3CAS + HO represents the computational overhead required by the *CN* for the cross-domain authentication of an ordinary drone in the drone group. Additionally, the computational overhead that the *CN* needs to undertake for the cross-domain authentication of the group leader (d_l) is 4CAS + BC + HO, which is the same as the cost of the *CN* in the process of SC. In addition, the computational overhead of d_j in the process of DGC is CAS + HO, which is reduced by two asymmetric encrypt/decrypt operations compared to that of d_j in the process of SC. In the process of NG, although the computational cost of d_j is (N − 1) × TA, the overall computational overhead of the drone group is $\frac{N(N-1)}{2}$TA because only one TA is required between two drones.

Table 3. The computational overhead in different phases of tasks.

	RG	TR	SC	TA	NG	JG	DGC	NMD
CN	DAT + GKP + BC + CAS + LD	PP + 3BC + CAS	4CAS + BC + HO	-	-	-	4CAS + BC + HO + (N−1) × (3CAS + BC + HO)	4BC + PP
d_j	-	-	3CAS + HO	4CAS + BC + HO	(N − 1) × TA	2TA + GL	CAS + HO	
d_l	-	-	-	4CAS + BC + HO	(N − 1) × TA + GL	TA + GL	3CAS + HO	

Note: "-" means no relevant operation.

To evaluate the efficiency of the proposed cross-domain authentication scheme, we firstly evaluated the computational time cost of the single-drone cross-domain (SC) authentication and the drone group cross-domain (DGC) authentication. Additionally, we evaluated the computational time consumption of the CN side and the UAV side in different situations. Secondly, we compared the computational time cost of our method with that of existing methods [25–27,31–33]. To further illustrate the advantages of our method, we also evaluated the variation in the computational time overhead with the number of drones, and compared it with that of existing methods [25,31].

5.2.2. Results and Discussion

In the cases of SC and DGC, the computational time of the main operations is shown in Table 4. The time cost in the table is the average time cost of executing the corresponding operation 100 times on the corresponding platform.

Table 4. The computational time of the main operations.

Notation	Description	CN	UAV
T_{EE}	ECC encrypt operation	2.43 ms	7.24 ms
T_{ED}	ECC decrypt operation	3.61 ms	9.31 ms
T_{HO}	Hash function	0.03 ms	0.32 ms
T_{BC}	Blockchain query	0.17 ms	0.63 ms

The computational time consumption of the CN side and the UAV side in different situations is shown in Figure 9. In the SC case, the time consumption of the CN side is $\{2 \times 3.61 + 2 \times 2.43 + 0.17 + 0.03\} = 12.28$ ms, and the time consumption of the UAV side is $\{2 \times 9.31 + 7.24 + 0.32\} = 26.18$ ms. In the case of DGC, the computational time cost needs to consider the scale of the drones. The time consumption of the UAV group leader is $\{2 \times 9.31 + 7.24 + 0.32\} = 26.18$ ms, and the time cost of an ordinary UAV in the group is $\{9.31 + 0.32\} = 9.63$ ms. The minimum average time consumption of the CN side is the time consumption when dealing with ordinary UAVs, i.e., $\{2 \times 3.61 + 2.43 + 0.17 + 0.03\} = 9.85$ ms. The maximum average time consumption on the CN side is when there are only two drones, that is, $\{(12.28 + 9.85)/2\} = 11.07$ ms. Therefore, the average computational time consumption interval of the CN side is (9.85 ms, 11.07 ms).

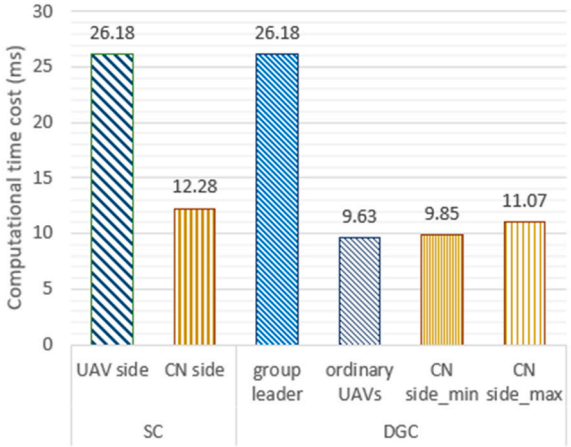

Figure 9. The computational time cost of SC and DGC.

The computational time cost of the proposed method and existing methods are shown in Figure 10. In the SC case, the computational time cost of our method is $\{12.28 + 26.18\} = 38.46$ ms. The figure also shows the maximum average computational time cost in the case of DGC, which is the average computational time cost required for each drone to cross domains when two drones perform DGC. The time cost is calculated by $\{(26.18 + 9.63 + 9.85 + 12.28)/2\} = 28.97$ ms. The existing methods for authenticating identity through a ground station as a trusted third party, as reported by Wazid et al. [25], Srinivas et al. [26], and Tanveer et al. [27], require 42.36 ms, 39.32 ms, and 38.12 ms, respectively. The existing methods for identity authentication through ground stations and blockchain architectures, as reported by Feng et al. [31], Shen et al. [32], and Gauhar et al. [33], require 32.93 ms, 36.87 ms, and 34.52 ms, respectively. Although the computational time cost of SC is not significantly different from that of existing methods [25–27,31–33], that of DGC is lower than that of other methods. Therefore, it can be considered that the DGC method can reduce the computational time cost of UAV cross-domain authentication. Figure 11 shows the computational time cost of cross-domain authentication when the number of drones increases. The computational time cost of DGC can be expressed as $\{26.18 + 12.28 + (N - 1)(9.63 + 9.85)\} = 19.48N + 18.98$ ms. As shown in

Figure 11, the time cost of each method increases linearly as the number of drones increases. Compared with the existing methods [25,31], the DGC method proposed in this paper has significant advantages when the number of drones is large.

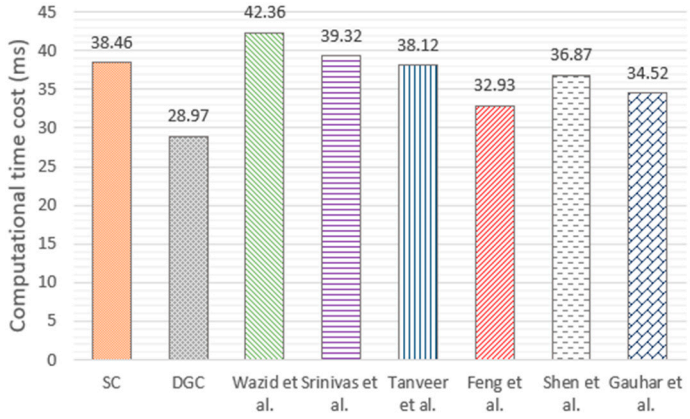

Figure 10. The comparison of computational time cost of different methods [25–27,31–33].

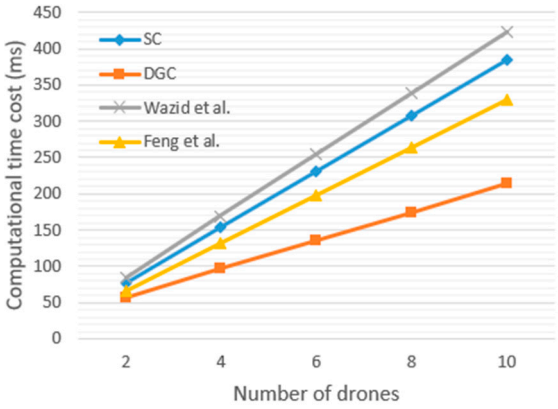

Figure 11. The computational time cost with increasing number of drones [25,31].

5.3. Communication Overhead

5.3.1. Materials and Methods

To evaluate the communication overhead of the proposed framework, we analyzed the number of communicated messages (bits) transmitted in different task model phases and compared it with existing advanced authentication schemes. In the SC case, the communicated messages are: $CRM: \left\{Crequest,\ Enc_{SK_{cn_i}}\left(Did_{d_j}\right)\right\}$, $ack: \left\{Enc_{PK_{d_j}}(x)\right\}$, $rsp_j: \left\{Enc_{PK_{cn_n}}(x+1||\ y)\right\}$, and $rsp_{cn}: \left\{Enc_{PK_{d_j}}(y+1)\right\}$. The length of the CRM, ack, rsp_j, and rsp_{cn} is {64 + 4026} = 4090 bits, 1094 bits, 1094 bits, and 1094 bits, respectively. Thus, the total communication cost of the SC is 7372 bits. In the two-way authentication (TA) case, the communicated messages are: $M_{d_j}: \left\{Taut, Enc_{SK_{cn_i}}\left(Pid_{d_j}\right), PBH_{d_j}\right\}$, $M_{d_n}: \left\{Taut, Enc_{SK_{cn_i}}(Pid_{d_n}), PBH_{d_n}\right\}$, $ack: \left\{Enc_{PK_{d_j}}(x)\right\}$, $rsp_j: \left\{Enc_{PK_{d_n}}(x+1\Big|\Big|y\ )\right\}$, and $rsp_n: \left\{Enc_{PK_{d_j}}(y+1)\right\}$. The length of the M_{d_j}, M_{d_n}, ack, rsp_j, and rsp_n is {32 + 2350 + 128} = 2510 bits, 2510 bits, 1094 bits, 1094 bits, and 1094 bits, respectively. Thus, the total communication cost of the SC is 8302 bits. The communicated messages in the DGC case can be divided into two parts: (a) the communicated messages in the group leader authentication process, and (b) the communicated messages in an ordinary group member authentication process. The communicated messages in the group leader authentication process are: $GCM_{d_l}: \left\{GCrequest,\ Enc_{SK_{cn_i}}(Did_{d_l}),\ GList_{d_l}\right\}$, $ack: \left\{Enc_{PK_{d_l}}(x)\right\}$, $rsp_j:$

$\{Enc_{PK_{cn_n}}(x+1||\ y)\}$, and rsp_{cn} : $\left\{Enc_{PK_{d_j}}(y+1)\right\}$. The length of the GCM_{d_j}, ack, rsp_j, and rsp_{cn} is {64 + 4026 + N × 64} = 4090 + 64N bits, 1094 bits, 1094 bits, and 1094 bits, respectively. Thus, the total communication cost of (a) is 7372 + 64N bits, where N is the number of members in the drone group. The communicated messages in an ordinary group member authentication process are: GCM_{d_j} : $\left\{GCrequest,\ Enc_{PK_{cn_n}}\left(Pid_{d_j}, x_j\right), Enc_{SK_{cn_i}}\left(Did_{d_j}\right)\right\}$, and $rsp^j_{cn} = Enc_{PK_{d_j}}(y_j)$. The length of the GCM_{d_j} and rsp^j_{cn} is {32 + 2570 + 4026} = 6628 bits and 1094 bits, respectively. Thus, the total communication cost of (b) is 7722 bits. For a drone group with N members, the total communication cost is {a) + (N − 1)b)} = {7372 + 64N + 7722N−7722} = 7786N-350 bits. The average cost per drone is $7786 - 350/N$ bits.

5.3.2. Results and Discussion

Figure 12 shows the communication overhead required at different stages of tasks. The average minimum overhead for the DGC case is in the situation when two drones compose a group (7611 bits). To evaluate the cross-domain communication overhead, we compared our method with the existing novel methods, as shown in Figure 13. Among them, the methods proposed by Wazid et al. [25], Srinivas et al. [26], and Tanveer et al. [27], authenticating identity through a ground station as a trusted third party, require 6642 bits, 5938 bits, and 5522 bits, respectively. The above methods [25–27] mainly guarantee the credibility of the identity by performing multiple process parameter calculations on the device, registration center, and control center. Therefore, the communication overhead of the above methods is relatively lower than that of our method, but it increases the computational burden. The methods for identity authentication through ground stations and blockchain architectures proposed by Feng et al. [31], Shen et al. [32], and Gauhar et al. [33] require 7168 bits, 9280 bits, and 7680 bits, respectively. We noticed that the communication cost of DGC is higher than that of SC, and the maximum communication overhead difference between DGC and SC is 7786 − 7372 = 414 bits. Additionally, Figure 10 shows that the computational time cost of SC is 38.46 ms, and the maximum average computational time cost of DGC is 28.97 ms. SC is 9.49 ms slower than DGC, and the difference increases as the number of drones increases. Therefore, it can be concluded that when the communication rate is higher than 414/9.49 = 43.6 b/ms = 43.6 kbps, DGC outperforms SC. As far as we know, a communication rate of 43.6 kbps is easily achievable. Therefore, while ensuring the overall efficiency of cross-domain authentication, it is feasible to consider increasing a portion of communication overhead to ensure security.

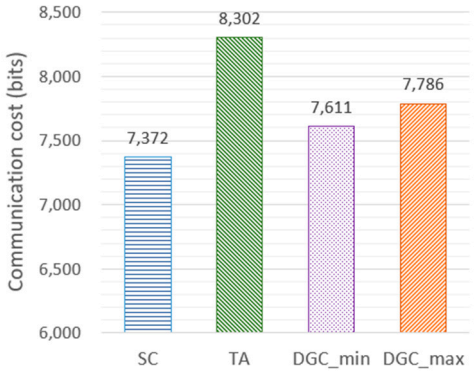

Figure 12. The communication cost in different task model phases.

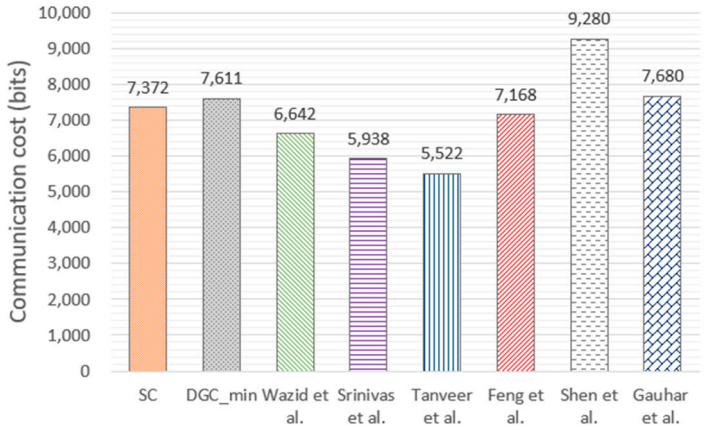

Figure 13. The comparison of communication cost of different methods [25–27,31–33].

The total communication time cost mainly includes transmission delay and propagation delay. The transmission delay can be expressed as $Sizedata/Tr$, where $Sizedata$ represents the size of the transmission data, and Tr represents the transmission rate of the channel. According to different transmission frequencies and communication bandwidths, Tr varies from tens of Kb to tens of Mb per second. Figure 14 shows how the transmission delay in the communicated messages changes with the transmission rate from 200 Kbps to 10 Mbps. We selected the minimum communication cost (Tanveer et al. [27]) and the maximum communication cost (Shen et al. [32]) among the comparison methods to compare them with our method. When the transmission rate is 5 Mps, the transmission time taken by SC, DGC_max, Tanveer et al. [27], and Shen et al. [32] is 1.43 ms, 1.51 ms, 1.07 ms, and 1.8 ms, respectively. The propagation delay can be expressed as $Dis/Velwave$, where Dis is the distance between the two sides of the communication and $Velwave$ is the propagation speed of the wave in the vacuum (about 3×10^5km/s). Generally, the range of the domain is around 1 km. Therefore, propagation delay can be ignored.

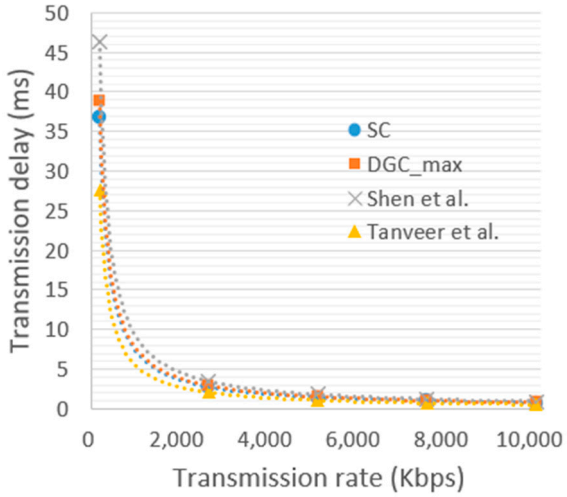

Figure 14. The transmission delay with increasing transmission rate [27,32].

5.4. Cross-Domain Authentication Time Cost

The total cross-domain time includes the computational time and communication time. Based on the experimental results in Section 5.2 and 5.3, we further evaluated the total cross-domain time taken by the method proposed in this paper by comparing it with the existing novel methods [25–27,31–33].

The experimental results are shown in Figure 15. The communication time cost of each scheme is that recorded when the transmission rate is 5 Mps. The total cross-domain time

taken by the DGC proposed in this paper can be expressed as {19.48N + 18.98 + 1.55N − 0.07} = 21.03N + 18.91 ms, where N is the number of members in the drone group. In the case of cross-domain authentication for one drone, the SC method, Wazid et al. [25], Srinivas et al. [26], Tanveer et al. [27], Feng et al. [31], Shen et al. [32], and Gauhar et al. [33] require 39.89 ms, 43.69 ms, 40.51 ms, 39.19 ms, 34.37 ms, 38.67 ms, and 36.06 ms, respectively. The communication time cost of DGC in the figure (30.49 ms) is the average value for two drones. It can be seen that DGC has a better cross-domain authentication performance compared with the other methods [25–27,31–33]. Figure 16 shows the cross-domain authentication time cost when the number of drones increases. It can be seen that the DGC method proposed in this article has significant advantages when the number of drones is large.

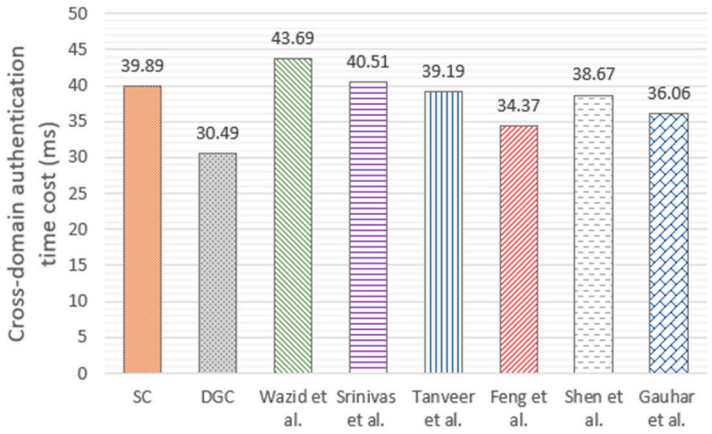

Figure 15. The comparison of cross-domain authentication time cost of different methods [25–27,31–33].

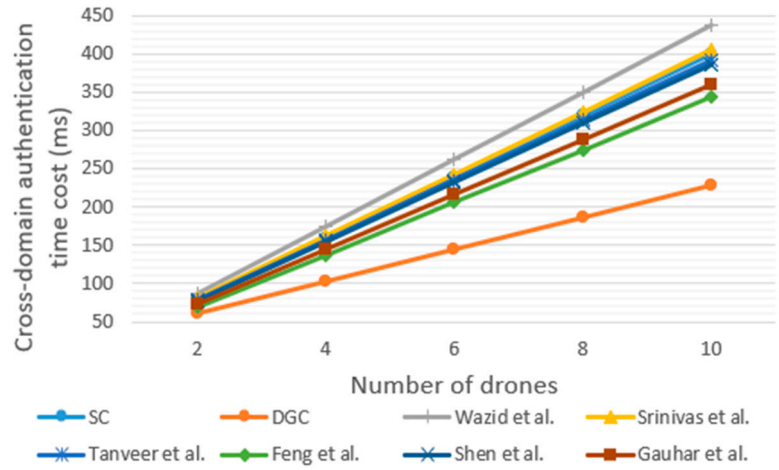

Figure 16. The cross-domain authentication time cost with increasing number of drones [25–27,31–33].

6. Security Analysis

We used the widely used Dolev and Yao (DY) threat model [35] to evaluate the security of the proposed method. In the DY threat model, a malicious attacker (MA) can inject, delete, eavesdrop, forge, or modify the exchanged messages over a public channel [36]. In this way, an MA can perform various security attacks on drones or CNs. The possible attacks and descriptions are as follows:

(1) Replay attack: An MA replays authentication messages to deceive the *CN*.
(2) Forgery attack: An MA generates an illegal or false ID to deceive the *CN*.
(3) Impersonation attack: An MA obtains authentication messages by impersonating terminals or eavesdropping on a channel, and impersonates a legitimate device to deceive the *CN*.

(4) Man-in-the-middle attack: An MA captures authentication messages and spoofs both parties of the communication.
(5) Database tampering: An MA attempts to tamper with the identity information in the database to pass the authentication.

Additionally, we made two assumptions: (a) The private keys of the *CN*s and drones are not revealed; and (b) An MA cannot deduce the private key from the public key, or it takes a lot of time. Considering the potential threats, we analyzed and compared the proposed scheme with the existing cross-domain authentication methods in terms of mutual authentication, cross-domain authentication, decentralization, anonymity, task path untraceability, path planning in advance, resilience to replay attacks, resilience to forgery attacks, resilience to impersonation attacks, resilience to man-in-the-middle attacks, and resilience to database tampering. The security analysis and comparison results are shown in Table 5.

Table 5. Security analysis and comparison results.

Features	[25]	[26]	[27]	[31]	[32]	[33]	Ours
Mutual authentication	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Cross-domain authentication	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Decentralization	No	No	No	Yes	Yes	Yes	Yes
Anonymity	Yes	Yes	Yes	Yes	No	No	Yes
Task path untraceability	No	No	No	No	No	No	Yes
Path planning in advance	No	No	No	No	No	No	Yes
Resilient to replay attack	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Resilient to forgery attack	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Resilient to impersonation attack	Yes	Yes	Yes	Yes	No	Yes	Yes
Resilient to man-in-the-middle attack	Yes	Yes	Yes	Yes	Yes	No	Yes
Resilient to database tampering	No	No	No	Yes	Yes	Yes	Yes

All the methods can well support mutual authentication and cross-domain authentication functions. At the same time, due to the use of blockchain technology and temporary intradomain ID methods, our method also has good decentralization and anonymity. Drones have different temporary IDs in different domains, and their device IDs on the *CBC* and all mission information can only be queried by the consortium nodes. Therefore, an MA cannot obtain the complete flight path of the drone, namely, task path untraceability. The notification mechanism between domains designed in this paper allows *CN*s to plan their paths in advance, which can improve their perception of the overall network situation. For possible attacks, we make the following analysis:

(1) Resilience to replay attacks: During the process of cross-domain authentication, the *CN*s and drones use PKI and a challenge–response mechanism to perform identity authentication and generate a session key. An MA cannot obtain useful information through this attack.
(2) Resilience to forgery attacks: The *CN*s need to query the *Devices* chain and the *Mission* chain transaction to confirm identity, and an MA cannot forge identity on the consortium chain.
(3) Resilience to impersonation attacks: Unregistered drones cannot obtain a legal *Did*, public key, and private key. In the process of the challenge–response game, an MA cannot decrypt the ciphertext to complete the verification. Therefore, it is difficult to implement an impersonation attack.

(4) Resilience to man-in-the-middle attacks: The communication data are encrypted by a public key or session key, which solves the problem of private data leakage. Even if the data are captured, the MA cannot decrypt the ciphertext to obtain the message.
(4) Resilience to database tampering: The important data are stored on the consortium blockchain. Only when the MA holds more than 51% of the nodes can it change the data in the blockchain, which is impracticable.

7. Conclusions

During long-distance flights for cargo transportation, drones need to apply cross-domain authentication mechanisms to enter the next domain. However, due to public wireless communication channels, drones are vulnerable to various security attacks in the process of cross-domain authentication. When facing a large number of cross-domain requests from drones, a CN requires significant computational and time overhead, which may lead to long waiting times for the cross-domain authentication of drones. To address this problem, we proposed BCDAIoD, an efficient blockchain-based cross-domain authentication scheme for the Internet of Drones. The BCDAIoD method includes a single-drone cross-domain authentication method, an establishment mechanism of drone groups, a drone group cross-domain authentication method, and a notification mechanism between domains. By taking advantage of blockchain, PKI, and the challenge–response game, BCDAIoD can ensure the authenticity and integrity of data, and can effectively prevent various attacks on drones and CNs. Furthermore, BCDAIoD uses the CBC and notification mechanism between domains to enable CNs to plan paths for drones in advance, which can further improve the efficiency of drone cross-domain authentication and task execution. The main contribution of this article is that BCDAIoD can improve the efficiency and security of the cross-domain authentication of drones. Experiment results show that the cross-domain authentication time cost and computational overhead of BCDAIoD are significantly lower than those of the existing state-of-the-art methods when facing a large number of drones.

Nevertheless, there are still limitations when applying BCDAIoD. First, blockchain brings additional communication and storage costs to the drone network. For example, drones in the IoD communicate with each other and update their local blockchains. Second, a small number of drones flying to the same destination or drones being far apart from each other may lead to drone group establishment failure. Hence, to address the above limitations, we seek to further simplify storage data in the block and design block pruning algorithms for the PBC to reduce communication and storage costs in future extensions of this work. At the same time, we will also attempt to design an optimization algorithm that dynamically adjusts between single-drone and drone group cross-domain methods based on the current state of IoD.

Author Contributions: Conceptualization, G.Q. and Y.Z.; methodology, G.Q. and T.Y.; software, T.Y. and G.Q.; investigation, G.Q. and Y.Q.; validation, G.Q., Y.Z. and T.Y.; result analysis, T.Y., Y.Q.; writing—original draft preparation, G.Q.; writing—review and editing, Y.Z. and G.Q.; supervision, Y.Z.; funding acquisition, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (General Program) under Grant No. 61572253.

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

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

References

1. Hassan, M.A.; Javed, A.R.; Hassan, T.; Band, S.S.; Sitharthan, R.; Rizwan, M. Reinforcing communication on the internet of aerial vehicles. *IEEE Trans. Green Commun. Netw.* **2022**, *6*, 1288–1297. [CrossRef]
2. Salah, K.; Rehman, M.H.U.; Nizamuddin, N.; Al-Fuqaha, A. Blockchain for AI: Review and open research challenges. *IEEE Access* **2019**, *7*, 10127–10149. [CrossRef]
3. Farah, M.F.; Mrad, M.; Ramadan, Z.; Hamdane, H. Handle with Care: Adoption of Drone Delivery Services. In Proceedings of the Advances in National Brand and Private Label Marketing: Seventh International Conference, Barcelona, Spain, 17–20 June 2020; pp. 22–29.
4. Makhdoom, I.; Zhou, I.; Abolhasan, M.; Lipman, J.; Ni, W. PrivySharing: A blockchain-based framework for privacy-preserving and secure data sharing in smart cities. *Comput. Secur.* **2020**, *88*, 101653. [CrossRef]
5. Li, X.; Wang, Y.; Vijayakumar, P.; He, D.; Kumar, N.; Ma, J. Blockchain-based mutual-healing group key distribution scheme in unmanned aerial vehicles ad-hoc network. *IEEE Trans. Veh. Technol.* **2019**, *68*, 11309–11322. [CrossRef]
6. Qiu, J.; Grace, D.; Ding, G.; Yao, J.; Wu, Q. Blockchain-Based Secure Spectrum Trading for Unmanned-Aerial-Vehicle-Assisted Cellular Networks: An Operator's Perspective. *IEEE Internet Things J.* **2020**, *7*, 451–466. [CrossRef]
7. Bera, B.; Chattaraj, D.; Das, A.K. Designing secure blockchain-based access control scheme in IoT-enabled Internet of Drones deployment. *Comput. Commun.* **2020**, *153*, 229–249. [CrossRef]
8. Yapıcı, Y.; Rupasinghe, N.; Güvenç, I.; Dai, H.; Bhuyan, A. Physical layer security for NOMA transmission in mmWave drone networks. *IEEE Trans. Veh. Technol.* **2021**, *70*, 3568–3582. [CrossRef]
9. Asheralieva, A.; Niyato, D. Distributed dynamic resource management and pricing in the IoT systems with blockchain-as-a-service and UAV-enabled mobile edge computing. *IEEE Internet Things J.* **2020**, *7*, 1974–1993. [CrossRef]
10. Li, T.; Liu, W.; Wang, T.; Ming, Z.; Li, X.; Ma, M. Trust data collections via vehicles joint with unmanned aerial vehicles in the smart Internet of Things. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e3956. [CrossRef]
11. Nakamura, S.; Enokido, T.; Takizawa, M. Information flow control based on the CapBAC (capability-based access control) model in the IoT. *Int. J. Mob. Comput. Multimed. Commun.* **2019**, *10*, 13–25. [CrossRef]
12. Ali, G.; Ahmad, N.; Cao, Y.; Ali, Q.E.; Azim, F.; Cruickshank, H. BCON: Blockchain based access CONtrol across multiple conflict of interest domains. *J. Netw. Comput. Appl.* **2019**, *147*, 102440. [CrossRef]
13. Wang, Y.; Wang, H.; Wei, X.; Zhao, K.; Fan, J.; Chen, J.; Jia, R. Service Function Chain Scheduling in Heterogeneous Multi-UAV Edge Computing. *Drones* **2023**, *7*, 132. [CrossRef]
14. Jha, S.; Sural, S.; Atluri, V.; Vaidya, J. Specification and verification of separation of duty constraints in attribute-based access control. *IEEE Trans. Inf. Forensics Secur.* **2017**, *13*, 897–911. [CrossRef]
15. Sandhu, R.S.; Coyne, E.J.; Feinstein, H.L.; Youman, C.E. Role-based access control models. *Computer* **1996**, *29*, 38–47. [CrossRef]
16. Xu, S.; Ning, J.; Li, Y.; Zhang, Y.; Xu, G.; Huang, X.; Deng, R.H. Match in my way: Fine-grained bilateral access control for secure cloud-fog computing. *IEEE Trans. Dependable Secur. Comput.* **2022**, *19*, 1064–1077. [CrossRef]
17. Wang, K.; Zhang, X.; Qiao, X.; Li, X.; Cheng, W.; Cong, Y.; Liu, K. Adjustable Fully Adaptive Cross-Entropy Algorithms for Task Assignment of Multi-UAVs. *Drones* **2023**, *7*, 204. [CrossRef]
18. Abdel-Malek, M.A.; Akkaya, K.; Bhuyan, A.; Ibrahim, A.S. A proxy Signature-Based swarm drone authentication with leader selection in 5G networks. *IEEE Access* **2022**, *10*, 57485–57498. [CrossRef]
19. Fysarakis, K.; Soultatos, O.; Manifavas, C.; Papaefstathiou, I.; Askoxylakis, I. XSACd-Cross-domain resource sharing & access control for smart environment. *Future Gener. Comput. Syst.* **2018**, *80*, 572–582.
20. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. In *Decentralized Business Review*; Scholastica: Seoul, Korea, 2008; p. 21260.
21. Mehta, P.; Gupta, R.; Tanwar, S. Blockchain envisioned drone networks: Challenges, solutions, and comparisons. *Comput. Commun.* **2020**, *151*, 518–538. [CrossRef]
22. Al-Hilo, A.; Samir, M.; Assi, C.; Sharafeddine, S.; Ebrahimi, D. Cooperative content delivery in UAV-RSU assisted vehicular networks. In Proceedings of the 2nd ACM MobiCom Workshop on Drone Assisted Wireless Communications for 5G and Beyond, London, UK, 21–25 September 2020; pp. 73–78.
23. Arafeh, M.; El Barachi, M.; Mourad, A.; Belqasmi, F. A blockchain based architecture for the detection of fake sensing in mobile crowdsensing. In Proceedings of the 2019 4th International Conference on Smart and Sustainable Technologies (SpliTech), Split, Croatia, 18–21 June 2019; pp. 1–6.
24. García-Magariño, I.; Lacuesta, R.; Rajarajan, M.; Lloret, J. Security in networks of unmanned aerial vehicles for surveillance with an agent-based approach inspired by the principles of blockchain. *Ad Hoc Netw.* **2019**, *86*, 72–82. [CrossRef]
25. Wazid, M.; Das, A.K.; Kumar, N.; Alazab, M. Designing authenticated key management scheme in 6G-enabled network in a box deployed for industrial applications. *IEEE Trans. Ind. Inform.* **2020**, *17*, 7174–7184. [CrossRef]
26. Srinivas, J.; Das, A.K.; Wazid, M.; Vasilakos, A.V. Designing secure user authentication protocol for big data collection in IoT-based intelligent transportation system. *IEEE Internet Things J.* **2020**, *8*, 7727–7744. [CrossRef]
27. Tanveer, M.; Alkhayyat, A.; Naushad, A.; Kumar, N.; Alharbi, A.G. RUAM-IoD: A robust user authentication mechanism for the Internet of Drones. *IEEE Access* **2022**, *10*, 19836–19851. [CrossRef]
28. Jan, S.U.; Abbasi, I.A.; Algarni, F.; Khan, A.S. A verifiably secure ECC based authentication scheme for securing IoD using FANET. *IEEE Access* **2022**, *10*, 95321–95343. [CrossRef]

29. Rajamanickam, S.; Vollala, S.; Ramasubramanian, N. EAPIOD: ECC based authentication protocol for insider attack protection in IoD scenario. *Secur. Priv.* **2022**, *5*, e248. [CrossRef]
30. Ever, Y.K. A secure authentication scheme framework for mobile-sinks used in the internet of drones applications. *Comput. Commun.* **2020**, *155*, 143–149. [CrossRef]
31. Feng, C.; Liu, B.; Guo, Z.; Yu, K.; Qin, Z.; Choo, K.K.R. Blockchain-based cross-domain authentication for intelligent 5G-enabled internet of drones. *IEEE Internet Things J.* **2021**, *9*, 6224–6238. [CrossRef]
32. Shen, M.; Liu, H.; Zhu, L.; Xu, K.; Yu, H.; Du, X.; Guizani, M. Blockchain-assisted secure device authentication for cross-domain industrial IoT. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 942–954. [CrossRef]
33. Ali, G.; Ahmad, N.; Cao, Y.; Khan, S.; Cruickshank, H.; Qazi, E.A.; Ali, A. xDBAuth: Blockchain based cross domain authentication and authorization framework for Internet of Things. *IEEE Access* **2020**, *8*, 58800–58816. [CrossRef]
34. Zhang, H.; Chen, X.; Lan, X.; Jin, H.; Cao, Q. BTCAS: A blockchain-based thoroughly cross-domain authentication scheme. *J. Inf. Secur. Appl.* **2020**, *55*, 102538. [CrossRef]
35. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
36. Yu, S.; Das, A.K.; Park, Y.; Lorenz, P. SLAP-IoD: Secure and lightweight authentication protocol using physical unclonable functions for internet of drones in smart city environments. *IEEE Trans. Veh. Technol.* **2022**, *71*, 10374–10388. [CrossRef]