Cross-domain Authentication for 5G-enabled UAVs: A Blockchain Approach

Bin Liu*
liubin10@foxmail.com
College of Computer Science
Sichuan Normal University
Chengdu, Sichuan, China

Keping Yu
keping.yu@aoni.waseda.jp
Global Information and Telecommunication Institute
Waseda University
Tokyo 169-8050, Japan

Chaosheng Feng
csfenggy@126.com
College of Computer Science
Sichuan Normal University
Chengdu, Sichuan, China

Kim-Kwang Raymond Choo
raymond.choo@fulbrightmail.org
Department of Information Systems and Cyber Security
University of Texas at San Antonio
San Antonio, TX 78249-0631, USA

ABSTRACT

While 5G facilitates high-speed Internet access and makes over-the-horizon control a reality for unmanned aerial vehicles (UAVs), there are also potential security and privacy considerations, for example, authentication among drones. The centralized authentication approaches not only suffer from a single point of failure, but they are also incapable of cross-domain authentication, which complicates the cooperation of drones from different domains. To address these challenges, we propose a blockchain-based solution to achieve cross-domain authentication for 5G-enabled UAVs. Our approach employs multiple signatures based on threshold sharing to build an identity federation for collaborative domains. Reliable communication between cross-domain devices is achieved by utilizing smart contract for authentication. Our performance evaluations show the effectiveness and efficiency of the proposed scheme.

KEYWORDS

5G-enabled UAVs, distributed collaboration, blockchain, smart contract, cross-domain authentication

ACM Reference Format:
Bin Liu, Keping Yu, Chaosheng Feng, and Kim-Kwang Raymond Choo. 2022. Cross-domain Authentication for 5G-enabled UAVs: A Blockchain Approach. In *4th ACM Workshop on Drone-Assisted Wireless Communications for 5G and Beyond (DroneCom'21), January 31-February 4 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3477090.3481053

*Bin Liu and Keping Yu are co-first authors. Corresponding author: Chaosheng Feng

DroneCom'21, January 31-February 4 2022, New Orleans, LA, USA

ACM ISBN 978-1-4503-8705-7/22/01…$15.00
https://doi.org/10.1145/3477090.3481053

1 INTRODUCTION

Intelligent unmanned aerial vehicles (UAVs; also known as drones) based on AI [9] have high degree of autonomy. They can potentially play an important role in social governance, logistics and transportation, and traffic control in the future [14]. Given the mobility of drones and the complexity of their networks, two of the biggest challenges in drone deployments are information security and privacy protection [1]. 5G-enabled Internet of Things (IoT) systems [8] can achieve higher performance in drone communication. However, limitations in existing 5G network environments include the potential of identity forgery, data fraud and privacy leakage [11]. In addition, there exists communication barriers between cross-domain devices due to the existence of different authentication mechanisms among them. These limitations compound the challenge of guaranteeing data privacy and authentication security during collaboration [10]. What's more, the conventional centralized authentication methods suffer from single-node failure and identity management conundrum [13].

Motivated by the aforementioned challenges, a blockchain-assisted cross-domain authentication approach for 5G-enabled UAVs is proposed to achieve secure and effective authentication in this paper. As a permission blockchain, consortium blockchain comprises multiple cooperating peer nodes and new nodes need to be authorized before joining the system [7]. Nodes of consortium blockchain do not fully trust each other but work together under contractual constraints, so it can be used to establish trust between different domains. Smart contracts are digital contracts that run on blockchain [1]. In the proposed scheme, each domain deploys one peer node for maintaining consortium blockchain. The authorized drones are registered to a smart contract that contains an access control table for dynamic identity management and authentication.

The remainder of the paper is organized as follows. Section 2 gives an overview of the system, and the implementation is presented in section 3. Simulation experiments are conducted to evaluate the performance of the proposed scheme in section 4. Finally, the conclusion is drawn.

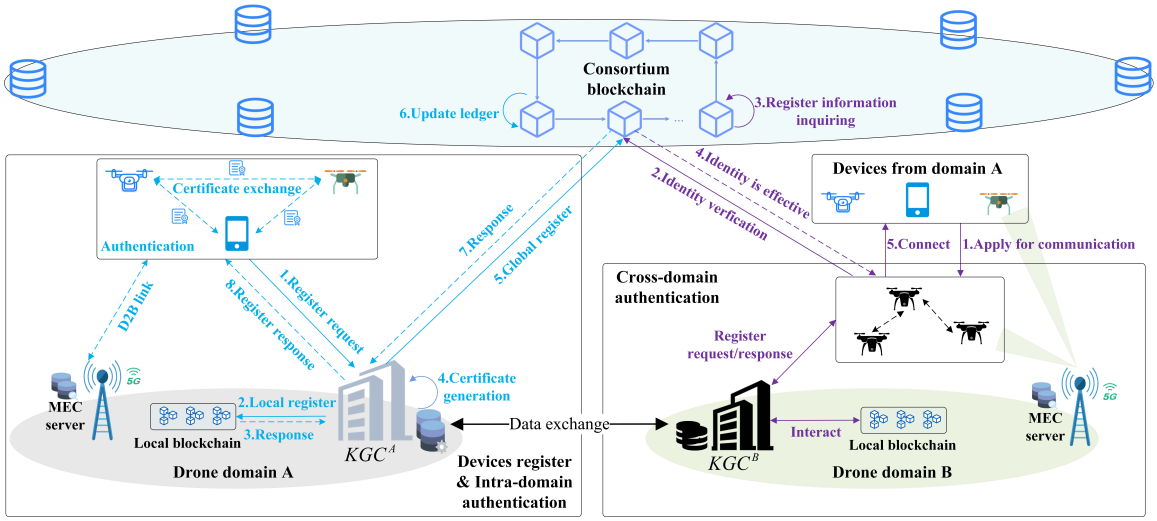

Figure 1: System Framework.

2 SYSTEM OVERVIEW

The proposed scheme is constructed on blockchain [12] and bilinear mapping [5] technology. For intra-domain drones, authentication can be achieved by employing an ad hoc network to exchange digital certificates obtained from *KGC* (for the case of no Internet coverage) or by applying consortium blockchain; for cross-domain drones, authentication needs to be achieved by applying consortium blockchain. This paper focuses on the realization of cross-domain authentication for drones. The framework is shown in Fig.1 which contains the following roles:

Blockchain: Local private blockchain and consortium blockchain (*CBC*) are applied. The private blockchain is used for the management of intra-domain resources. The consortium blockchain is oriented to the authorized nodes for data sharing, and is mainly used for cross-domain authentication of drones.

Registration contract: An access control table that stores information of domains and registered devices.

Smart contract deployer: The deployer needs to register an account on the consortium blockchain. A (t, n) threshold multi-signature smart contract is created to eliminate the trust barriers between domains, where n denotes the total number of domains and t denotes the minimum number of signatures needed to deploy the contract.

Terminals: Terminals include drones that collect data and the management devices.

Mobile Edge Computing (MEC): The *MEC* provides data analysis, processing and caching services. The consensus nodes can also be deployed on the *MEC* server to maintain the blockchain ledger.

Key Generation Center (KGC): *KGC* is unique among authorized domains. It is responsible for the management of blockchain and terminals.

3 BLOCKCHAIN-ASSISTED CROSS-DOMAIN AUTHENTICATION

In this section, the design details of the proposed authentication scheme are presented. Set the global parameters $\{\mathbb{G}_1, g, p, H\}$, where $\mathbb{G}_1$ is the elliptic curve multiplicative cyclic group of order p, g is the generator, and H is the hash function $H : \{0,1\}^* \rightarrow \mathbb{G}_1$. The global parameters are broadcast to the participants and stored in consortium blockchain for sharing. Each KGC^{N_i} generates a public-private key pair $\{pk_{N_i}, sk_{N_i}\}$ for the domain N_i, where $sk_{N_i} \in \mathbb{Z}_p$ is chosen randomly and $pk_{N_i} = g^{sk_{N_i}}, i = 1, 2, \ldots, n$.

3.1 Threshold Multi-signature Construction

We employ a broadcast multiple digital signature method with threshold (t, n) based on secret sharing [4]. The members involved in multi-signature are: the signature request initiator who is also the sub-secret distributor UI, the domains involved in signing $U_i (i = 1, 2, \ldots, n)$, the signature collector UC, and the signature verifier UV. Each U_i has its identifier GID_i, and it should be noted that UC and UV are the same object in the proposed scheme, i.e., the contract deployer. The initiator of the contract creation request broadcasts the compiled bytecode σ of the contract to $U_i (i = 1, 2, \ldots)$, where σ serves as the message that needs to be signed. The construction process of multi-signature is as follows.

1) Init: Each participant delivers pk_{N_i} as a key shadow to UI. UI has to confirm that each participant's secret shadow is exclusive, i.e., $pk_{N_i} \neq pk_{N_j}$, where $i \neq j$. If the same key shadow is found, participants are required to re-choose their private keys sk_{N_i} and resend pk_{N_i} to UI. Next, UI randomly chooses a secret $k \in \mathbb{Z}_p$ and promulgates the secret shadow $K = g^k$. Then the construction of the secret polynomial is performed. Choosing $t-1$ integers $a_1, ..., a_{t-1}$ and randomly generating polynomial $f(x) = k + a_1 x + ... + a_{t-1} x^{t-1}$. Then the sub-secret $d_i = f(GID_i)$ is calculated. UI randomly selects an integer s as the master key, calculates $P = g^s$ and the subkey $s_i = pk_{N_i}^s$. Next, the R_i is obtained by performing calculation on d_i and s_i of $t' (t \leq t' \leq n)$ participants:

$$R_i = d_i \oplus H(s_l) \oplus H(s_m) \oplus ... \oplus H(s_j), \tag{1}$$

where l, m, and j represent the members involved in the secret sharing. Finally, (GID_i, R_i) and P are published, where $i \in [1, t']$.

2) Sub-secret recovery: Assuming that one participant is planning on recovering the secret and thus generating multi-signature. First, the participant U_i sends the subkey s_i to the secret recoverer to estimate whether the equation $e(g, s_i) = e(P, pk_{N_i})$ holds. If it does not hold, the participant needs to be re-verified. Conversely, when t' participants are verified, the identity of each participant is confirmed and each d_i is obtained by the following calculation:

$$d_i = R_i \oplus H(s_l) \oplus H(s_m) \oplus ... \oplus H(s_j). \tag{2}$$

Finally, d_i is sent to the corresponding U_i.

3) Single signature generation: UI sends σ to each U_i, and if U_i accepts the contract creation request, it computes $sig_i = H(\sigma)^{d_i}$. Next, U_i sends signature (σ, sig_i) on σ to UC and discloses sub-secret shadow $p_i = g^{d_i}$.

4) Single signature verification: After receiving (σ, sig_i), UC sets the number of valid signatures m to zero and verifies the validity of the signature by the following equation:

$$e(g, sig_i) = e(H(\sigma), p_i). \tag{3}$$

5) Multi-signature generation: When $m \geqslant t$, UC calculates

$$\begin{aligned} S &= \prod_{i=1}^{t} sig_i{}^{\Delta_{i,U_X}(0)} \\ &= \prod_{i=1}^{t} H(\sigma)^{d_i \cdot \Delta_{i,U_X}(0)} \\ &= H(\sigma)^{\sum_{i=1}^{t} d_i \cdot \Delta_{i,U_X}(0)} \\ &= H(\sigma)^{d}, \end{aligned} \tag{4}$$

where $\Delta_{i,U}(x) = \prod_{j \in U, j \neq i} \frac{x-j}{i-j}$ is the Lagrange coefficient. Finally, UC sends (σ, S) to UV as a multi-signature.

6) Multi-signature verification: UV calculates whether the equation $e(g, S) = e(H(\sigma), K)$ holds. If it holds the multi-signature is valid, otherwise it is invalid. When the verification is passed, UV submits a transaction $Tx_{global}(\sigma, S, GID_i)$, $i = 1, 2, ..., m$ to consortium blockchain, which is used to record the multi-signature and the signed domains involved.

3.2 Cross-domain Registration

The contract is deployed when no fewer than t valid signatures are collected by the contract deployer. The structure of the access control table in registration contract is shown in Fig.2. The domain A needs to register CBC account for d_i^A in advance. The CBC generates the public-private key pair $\{PK_i, SK_i\}$ for d_i^A, and PK_i is processed and encoded with secp256k1 elliptic curve to obtain account $Addr_i$. The process of cross-domain registration is shown in Fig.3, and includes the following steps:

1) d_i^A sends query request $Qrequest$ to inquire whether valid registration information of $Addr_i$ exists in domain N. If d_i^A has been registered in N but the registration validity T_i is expired, then d_i^A delegates KGC^A to submit the validity update request to KGC^N. KGC^N generates signature $sig = (Addr_i||T_j)^{sk_N}$ and sends it to KGC^A, which then calls updating algorithm to complete the updating. Specifically, T_j denotes the new validity period. If d_i^A has not been registered in domain N, then d_i^A submits a request $register_{global} = (GRrequest||Addr_i||h_i)$ to KGC^A, where $GRrequest$ is a global registration request and $h_i = H(ID_i)$). ID_i denotes the identifier of d_i^A. $GRrequest$ denotes global registration request for d_i^A in domain A or in other domains $N_i (i = 1, 2, ..., n)$. For the first category, KGC^A generates signature $sig = (Addr_i||h_i)^{sk_A}$

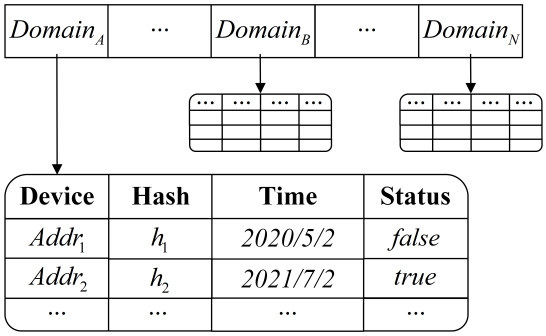

Figure 2: Access Control Table.

Algorithm 1 Cross-domain Registration Algorithm

Input:
Account address, $Addr_i$;
The hash of device's identifier, h_i;
An expiration date, T_i;
A signature, sig_i;
The signature domain's identifier, GID.
Output:
Smart contract address.
1. Get public key $pk_i = pk[GID]$;
2. Let $m = Addr_i||h_i$;
3. Compute $e_1 = e(m, pk_i)$;
4. Compute $e_2 = e(g, sig_i)$, if $e_2 = e_1$ doesn't hold, output $null$;
5. Get domain object $D = Domain[GID]$ and the index of the last registered device $l = D.endindex$;
6. Set $D.Device[l+1] = Addr_i, D.Hash[l+1] = h_i, D.Time[l+1] = T_i, D.Status[l+1] = true$;
7. Output Wsc.

Algorithm 2 Cross-domain Authentication Algorithm

Input:
Account address, $Addr_i$;
The domain's identifier, GID;
A session key ks.
Output:
Ciphertext encrypted by ks.
1. Get domain object $D = Domain[GID]$;
2. Get index $i = D.indexof(Addr_i)$ if $i = null$ holds, output $null$;
3. Get $status = D.Status[i], T_i = D.Time[i]$, and $h_i = D.Hash[i]$;
4. Compute $c = E_{ks}(Addr_i||status||T_i||h_i)$;
5. Output c.

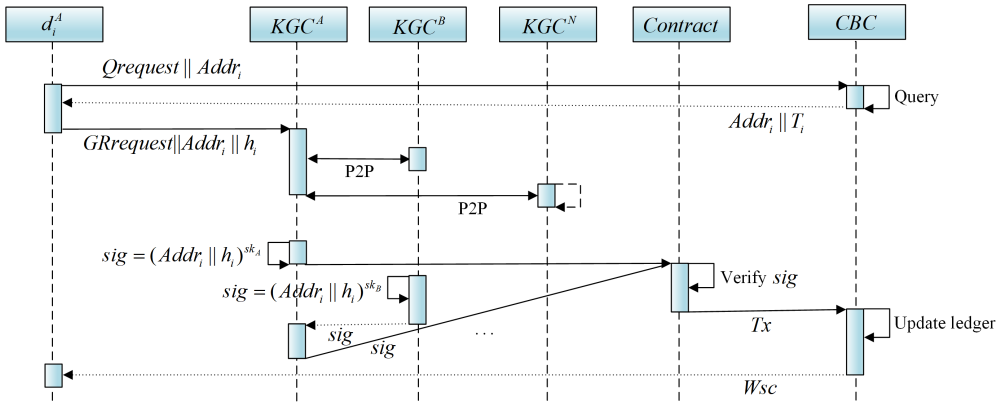

Figure 3: Overflow of Cross-domain Registration.

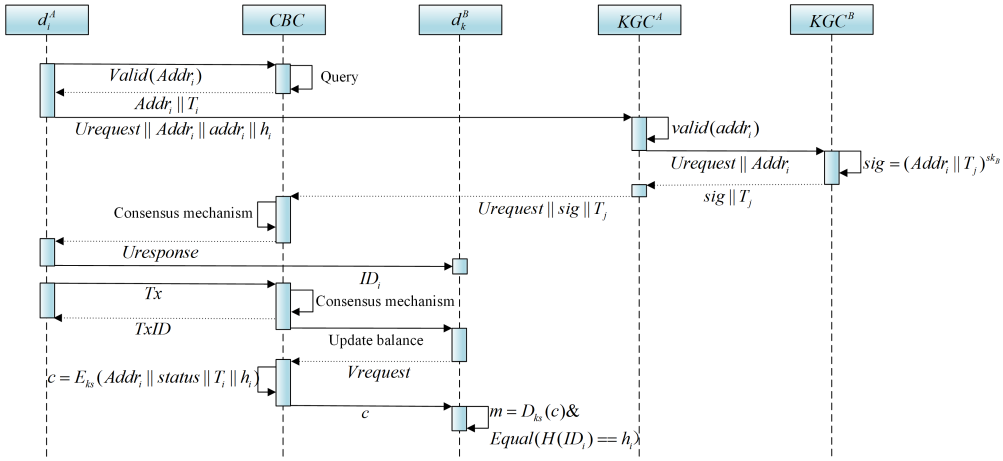

Figure 4: Overflow of Cross-domain Authentication.

for d_i^A and then jumps to step 3); for the second category, it proceeds to step 2).

2) KGC^A submits a cross-domain registration request to KGC^N to obtain $sig = (Addr_i||h_i)^{sk_N}$.

3) KGC^A takes the device information and sig as the input of the contract, then calls Algorithm 1 to complete cross-domain registration. The device information is written to the access control table in the contract where the signature field is located only if sig is legitimate. Then a transaction Tx is initiated to update the status in CBC. Finally, the CBC returns the registration contract address Wsc to d_i^A.

3.3 Corss-domain Authentication

Communication barriers exist between cross-domain devices due to differences in communication protocols or device types. The traditional model of using a central authority for authentication brings about enormous communication and storage overheads. We apply decentralized blockchain for the authentication of cross-domain devices to tackle these challenges. Assuming that drone d_i^A of domain A wants to communicate with drone d_k^B of domain B, the process of d_k^B authenticating d_i^A is shown in Fig.4, with the following steps.

1) d_i^A calls contract function $Valid(Addr_i)$ to check whether its registration time in domain B is expired, then the result $(Addr_i||T_i)$ is returned. If T_i is invalid, then d_i^A submits an update request $Urequest$ and its device information $(Addr_i||addr_i||h_i)$ to KGC^A, where $addr_i$ denotes private blockchain account; if T_i is valid, the operation in step 3) is carried out directly.

2) KGC^A calls function $valid(addr_i)$ to check whether d_i^A belongs to A, if it does, KGC^A sends updating request $Urequest$ to KGC^B on behalf of d_i^A. After obtaining signature $sig = (Addr_i||T_j)^{sk_B}$ and new validity T_j, KGC^A calls the registration contract and initiate an updating transaction. The nodes of CBC update the ledger and return result $Uresponse$ to d_i^A after reaching consensus.

3) d_i^A sends its ID_i to d_k^B through a secure channel, then computes $ct = Enc_{pk_A}(ID_i)$, $h_i = H(ID_i)$ and generates a connection request $op = (connect||ct||h_i)$. Next, d_i^A initiates a transaction Tx, with the payer being d_i^A's account $Addr_i$ and the payee being d_k^B's

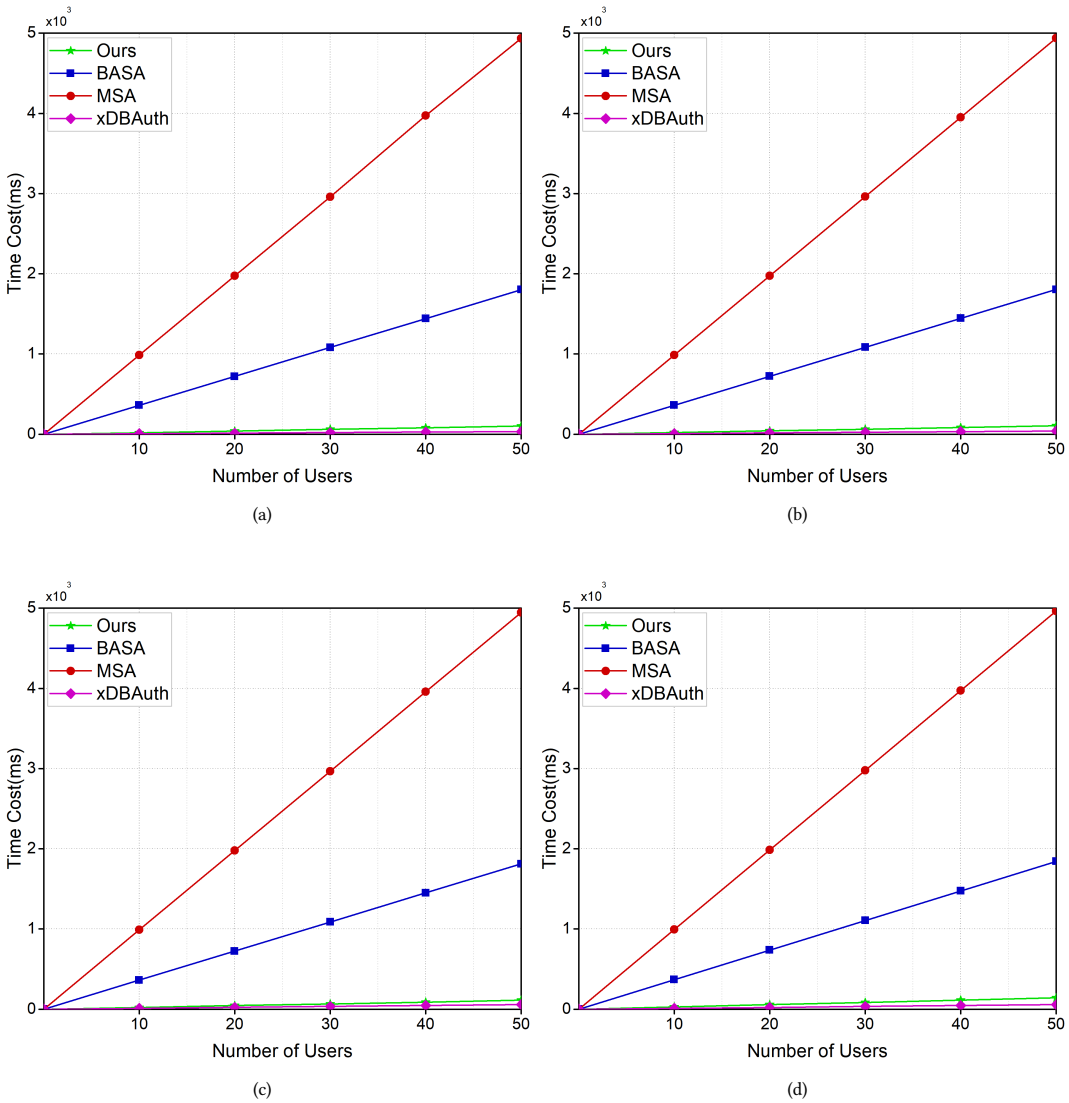

Figure 5: Time-consuming comparison of authentication: (a) 2 domains (b) 4 domains (c) 6 domains (d) 8 domains.

account $Addr_k$. The transaction amount is $coin_i$, and the additional information is op.

4) Nodes of CBC verify and add the transaction to the new block after reaching consensus, then return the transaction identifier $TxID$ to d_i^A and update the balance of account $Addr_k$. After receiving the transfer, d_k^B calls the contract through Wsc to obtain registration status of $Addr_i$ in the access control table of domain B in the contract. The authentication request submitted by d_k^B is $Vrequest = (Wsc, Addr_i, GID_B, ks)$, where ks indicates the session key negotiated in advance between CBC node and d_k^B.

5) Node of CBC obtains ciphertext $c = E_{ks}(Addr_i||status||T_i||h_i)$ by using Algorithm 2 which applies symmetric encryption technology. Then c is sent to d_k^B. If $Addr_i$ has been registered in domain B and the validity of registration date T_i is effective then $status = true$; otherwise, $status = false$.

6) d_k^B runs the symmetric decryption algorithm to obtain $m = D_{ks}(c) = (Addr_i||status||T_i||h_i)$. If $status = false$ then exit; otherwise judge the effectiveness of T_i. Let now be the current timestamp. If $T_i < now$ means that the registration date is expired, then exit. If $T_i > now$, then d_k^B performs hashing on ID_i to obtain $h = H(ID_i)$. If the function $Equal(h == h_i)$ outputs $true$, that means the identity of d_i^A is legal and the authentication is successful; otherwise, it fails. Note that the above authentication method is also applicable to realize intra-domain authentication.

4 PERFORMANCE EVALUATION

In this section, we analyze the performance of the proposed scheme based on several metrics by conducting simulation experiments. The configuration of the PC used in the experiments is: Intel Core i7-6700U CPU @2.6GHz, 16GB RAM, and Ubuntu 18.04 64-bit OS.

The JPBC bilinear pair cryptography library and Hyperledger Fabric were used to conduct the experiments. Bilinear and exponential operations depend on JPBC v2.0 and elliptic curve group with order length of 160 bits. The applied elliptic curve is a super singular elliptic curve $y^2 = x^3 + x$ on a 512-bit finite field. Moreover, JDK1.8 and Hyperledger Fabric v1.4 are installed on PC, where Fabric runs smart contracts through a docker container. The version of installed docker is v19.03. Consensus is reached through the Raft algorithm.

4.1 Computational Overhead

To calculate time cost of the proposed scheme, the above experimental environment was conducted for simulation experiments. It should be noted that system initialization as well as device registration were done in advance. The devices were only required to undertake a small number of computational tasks, which is friendly to resource-constrained drones. The time complexity and space complexity of both Algorithm 1 and Algorithm 2 are $O(1)$. The time-consuming operation of Algorithm 1 and Algorithm 2 are bilinear pairing and symmetric encryption, respectively. By conducting extensive experiments, we have averaged 50 sets of experimental data and found that it took 5ms to perform a bilinear pairing and 0.06ms to perform a symmetric encryption operation. Therefore, the time required to perform Algorithm 1 and Algorithm 2 is about 10ms and 0.06ms, which shows that the proposed algorithms are efficient.

Our scheme was compared with the cross-domain authentication mechanisms *BASA* [6], *MSA* [3] and *xDBAuth* [2] under the same experimental settings. Fig.5 shows the time cost of authentication when the number of domains was increased from two to eight. The time cost increased linearly for all of the schemes as the number of users increased. It should be noted that when the variable was just the number of domains, it only affected the time to read and write data on blockchain. Therefore, as the number of domains increased, it simply produced a slight increase in the time taken for read and write operations. From Fig.5, we can see that the proposed scheme outperformed *BASA* and *MSA*, and was only milliseconds away from *xDBAuth* in terms of authentication efficiency. The scheme *xDBAuth* requires multiple transactions to complete the authentication process. Transactions in the blockchain may be rejected, which will lead to additional time spent on failed transactions. Therefore, the proposed scheme performs better than other schemes because it only needs to submit one transaction in the authentication phase.

4.2 Communication Overhead

Because the length of the message varies with the task, only the message related to authentication is calculated. For cross-domain authentication, d_i^A initiates a 264 bytes transaction to d_k^B, d_k^B receives the transfer and submits a 232 bytes identity query request to *CBC*. Then *CBC* returns the query result in the form of 400 bytes of ciphertext to d_k^B, d_k^B decrypts and verifies the result to complete the process. The total communication overhead of authentication is 896 bytes. It should be noted that the total overhead of using blockchain for intra-domain authentication is also 896 bytes.

5 CONCLUSION AND FUTURE WORK

AI and 5G technology are two key enabling technologies in promoting and supporting drone deployments. To address the identity security limitations in drone deployments, we proposed a cross-domain authentication scheme for 5G-enabled UAVs based on blockchain. Specifically, the identity of drone is dynamically managed by applying a multi-signature smart contract. Entities from different domains can authenticate each other without knowing the true identities. The blockchain provides technical support for the security audit and the establishment of accountability mechanism. The performance of the proposed scheme was then evaluated to verify its effectiveness and efficiency. We also discussed the limitations of our approach. First, drones may have to initiate more than one transaction to complete the authentication if there exists a high failure rate of transactions. Second, write and read data on blockchain using smart contracts will introduce latency. Our future research will focus on overcoming these two limitations.

REFERENCES

[1] Chaosheng Feng, Keping Yu, Ali Kashif Bashir, Yasser D Al-Otaibi, Yang Lu, Shengbo Chen, and Di Zhang. 2021. Efficient and secure data sharing for 5G flying drones: a blockchain-enabled approach. *IEEE Network* 35, 1 (2021), 130–137.

[2] Ali Gauhar, Naveed Ahmad, Yue Cao, Shahzad Khan, Haitham Cruickshank, Ejaz Ali Qazi, and Azaz Ali. 2020. xDBAuth: Blockchain based cross domain authentication and authorization framework for Internet of Things. *IEEE Access* 8 (2020), 58800–58816.

[3] Shaoyong Guo, Fengning Wang, Neng Zhang, Feng Qi, and Xuesong Qiu. 2020. Master-slave chain based trusted cross-domain authentication mechanism in IoT. *Journal of Network and Computer Applications* 172 (2020), 102812.

[4] Lein Harn and Changlu Lin. 2009. Detection and identification of cheaters in (t, n) secret sharing scheme. *Designs, Codes and Cryptography* 52, 1 (2009), 15–24.

[5] Hang Li, Keping Yu, Bin Liu, Chaosheng Feng, Zhiguang Qin, and Gautam Srivastava. 2021. An Efficient Ciphertext-Policy Weighted Attribute-Based Encryption for the Internet of Health Things. *IEEE Journal of Biomedical and Health Informatics* (2021).

[6] Meng Shen, Huisen Liu, Liehuang Zhu, Ke Xu, Hongbo Yu, Xiaojiang Du, and Mohsen Guizani. 2020. Blockchain-assisted secure device authentication for cross-domain industrial IoT. *IEEE Journal on Selected Areas in Communications* 38, 5 (2020), 942–954.

[7] Na Shi, Liang Tan, Wenjuan Li, Xin Qi, and Keping Yu. 2020. A blockchain-empowered AAA scheme in the large-scale HetNet. *Digital Communications and Networks* (2020). https://doi.org/10.1016/j.dcan.2020.10.002

[8] Liang Tan, Huan Xiao, Keping Yu, Moayad Aloqaily, and Yaser Jararweh. 2021. A blockchain-empowered crowdsourcing system for 5G-enabled smart cities. *Computer Standards Interfaces* 76 (2021), 103517. https://doi.org/10.1016/j.csi.2021.103517

[9] Helin Yang, Arokiaswami Alphones, Zehui Xiong, Dusit Niyato, Jun Zhao, and Kaishun Wu. 2020. Artificial-intelligence-enabled intelligent 6G networks. *IEEE Network* 34, 6 (2020), 272–280.

[10] Helin Yang, Jun Zhao, Zehui Xiong, Kwok-Yan Lam, Sumei Sun, and Liang Xiao. 2021. Privacy-Preserving Federated Learning for UAV-Enabled Networks: Learning-Based Joint Scheduling and Resource Management. *IEEE Journal on Selected Areas in Communications* (2021).

[11] Keping Yu, Long Lin, Mamoun Alazab, Liang Tan, and Bo Gu. 2020. Deep Learning-Based Traffic Safety Solution for a Mixture of Autonomous and Manual Vehicles in a 5G-Enabled Intelligent Transportation System. *IEEE Transactions on Intelligent Transportation Systems* (2020), 1–11. https://doi.org/10.1109/TITS.2020.3042504

[12] Keping Yu, Liang Tan, Xinglin Shang, Junjie Huang, Gautam Srivastava, and Pushpita Chatterjee. 2021. Efficient and Privacy-Preserving Medical Research Support Platform Against COVID-19: A Blockchain-Based Approach. *IEEE Consumer Electronics Magazine* 10, 2 (2021), 111–120. https://doi.org/10.1109/MCE.2020.3035520

[13] Ke-Ping Yu, Liang Tan, Moayad Aloqaily, Hekun Yang, and Yaser Jararweh. 2021. Blockchain-Enhanced Data Sharing with Traceable and Direct Revocation in IIoT. *IEEE Transactions on Industrial Informatics* (2021), 1–1. https://doi.org/10.1109/TII.2021.3049141

[14] Yin Zhang, Yujie Li, Ranran Wang, Jianmin Lu, Xiao Ma, and Meikang Qiu. 2020. PSAC: Proactive Sequence-Aware Content Caching via Deep Learning at the Network Edge. *IEEE Transactions on Network Science and Engineering* 7, 4 (2020), 2145–2154. https://doi.org/10.1109/TNSE.2020.2990963