Hindawi
Security and Communication Networks
Volume 2019, Article ID 8583130, 12 pages
https://doi.org/10.1155/2019/8583130

Research Article

Designated Verifier Proxy Blind Signature Scheme for Unmanned Aerial Vehicle Network Based on Mobile Edge Computing

Lei He,[1,2] Jianfeng Ma,[1] Ruo Mo,[3] and Dawei Wei[1]

[1]*School of Computer Science and Technology, Xidian University, Xi'an 710071, China*
[2]*School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450000, China*
[3]*School of Cyber Engineering, Xidian University, Xi'an 710071, China*

Correspondence should be addressed to Jianfeng Ma; jfma@mail.xidian.edu.cn

Received 24 October 2018; Accepted 7 March 2019; Published 4 April 2019

Guest Editor: Esther Palomar

Unmanned Aerial Vehicle (UAV) has enormous potential in many domains. According to the characteristics of UAV, it is important for UAV network to assure low latency and integrity and authentication of commands sent by command center or command stations to UAV. In this paper, we proposed a UAV network architecture based on mobile edge computing (MEC) which helps guarantee low latency in the UAV network. Afterwards, we proposed a designated verifier proxy blind signature (DVPBS) scheme for UAV network and proved that it is existentially unforgeable under an adaptive chosen message attack in the random oracle model. We compared the efficiency of our DVPBS scheme with other signature schemes by implementing them in jPBC and theoretically analyzing their signature length. The experiment results indicate that our DVPBS scheme is efficient. The signature length of our DVPBS is longer, but it is still short enough compared with the transmission capacity of UAV.

1. Introduction

Unmanned Aerial Vehicle (UAV) is the aircraft without human pilot aboard. It is an emerging technology which can be applied for military applications and civil applications. Its military applications mainly include border surveillance, reconnaissance, and strike.

(i) Border surveillance. Kim et al. proposed a border surveillance system based on UAV [1]. This system uses electrification line system to implement wireless charging for UAV, which extends the flight duration of UAV.

(ii) Reconnaissance. Wang et al. proposed a multiple UAVs reconnaissance task allocation model for heterogeneous targets [2].

The civil applications of UAV mainly contain precision farming, disaster response, communication, equipment inspection and maintenance, etc.

(i) *Precision Farming*. Sona et al. mounted new sensors on UAV and acquired data through these sensors [3]. The multispectral images acquired by UAV can be integrated with ground geophysical data to obtain soil characteristics, which is useful for precious agriculture. Nintanavongsa et al. considered a smart farm platform which is based on UAV equipped with sensor and researched how sensor mobility affects network communication [4]. Pircher et al. designed a prototype hybrid UAV for the application in precision agriculture [5]. The UAV has high area coverage and high degree of automatization. It can perform predefined waypoint missions autonomously.

(ii) *Disaster Response*. Ahn et al. proposed a flying ad hoc network (FANET) routing protocol with bounded end-to-end communication delay in order to ensure that rescue information can be transmitted in time when disaster happens [6]. They also designed and implemented a simulation platform of FANET.

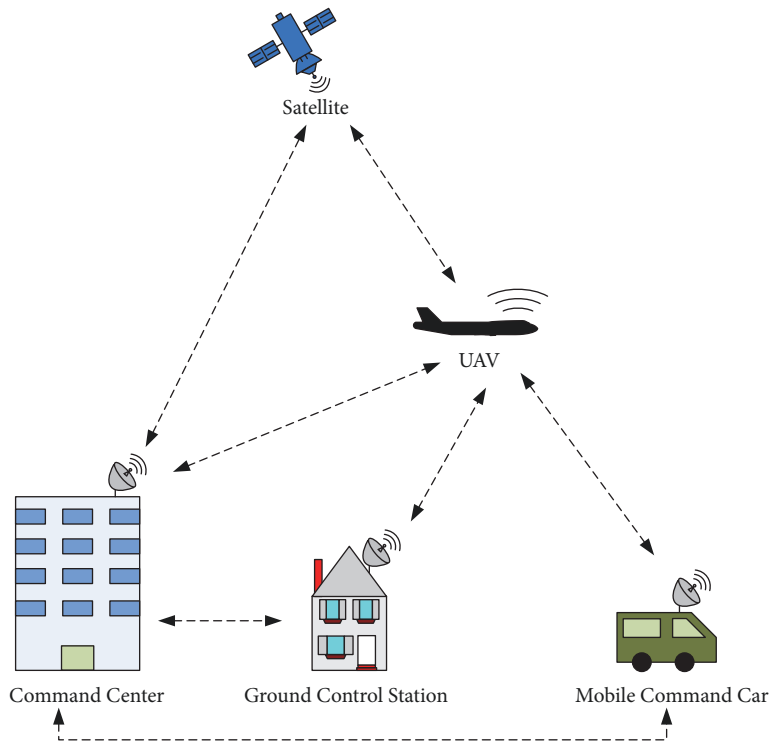

Figure 1: UAV network.

Hayajneh et al. developed a statistical framework to characterize and model large-scale post-disaster recovery cellular networks based on UAV [7]. They thought that there are some parameters which influence optimal deployment of recovery network after analysis. It can optimize the network by adjusting these parameters.

(iii) *Communication*. Deruyck et al. proposed a deployment tool for UAV-aid emergency network to handle a large number of network access requests in the event of disaster [8]. Lyu et al. proposed a hybrid network architecture which uses UAV as aerial mobile base station [9]. The UAV flies along the cell edge to offload data traffic.

(iv) *Equipment Inspection and Maintenance*. Wu et al. proposed a visual inspection method for rail surface defects based on UAV, which can improve efficiency and reduce cost [10]. Addabbo et al. proposed a UAV system for photovoltaic plant inspection [11]. It uses computer vision to identify panels and detect thermal anomalies.

In this paper, we informally classify UAVs into three categories according to their autonomy, namely, non-autonomous UAV, semi-autonomous UAV, and autonomous UAV. A non-autonomous UAV mainly depends on the control signal from operator's handheld controller. An operator sends control signal through the handheld controller to direct UAV to complete task. Non-autonomous UAV usually has small body size, low flight speed, short flight distance, and limited computing and communication capabilities. It typically performs task within the operator's line of sight. In contrast, an autonomous UAV has larger body size, higher flight speed, longer flight distance, and relatively sufficient computing and communication capabilities. It is able to fly farther and perform more complicated task without requiring the operator to command in real time. After it receives the command from operator, the UAV will check this command and decide whether to execute it. A semi-autonomous UAV is a kind of UAV between non-autonomous and autonomous UAVs. In this paper, we focus on the application of autonomous UAV.

Generally, UAV network refers to the network with UAV applications as its core. It has some characteristics such as high mobility, dynamic topology, intermittent link, power constraint, and changing link quality [12]. A UAV network may be comprised of UAV, command center, ground control stations, satellites, mobile command cars, or ships, which is illustrated in Figure 1. In a UAV network, UAV belongs to a command center. It accepts the commands issued by command center and performs related tasks. Ground control stations, satellites and mobile command cars, or ships may be all called command stations and mainly responsible for maintaining the communication between command center and UAV. The command center can communicate with UAV with the help of command station which forwards the commands received from command center to UAV. In addition, the command station also commands temporarily UAV in some cases.

1.1. Problem Statement. When an autonomous UAV performs a task, it will receive the command which contains relevant parameters about task, including time, target location, and

actions to be performed. It can fly autonomously to the target location in the specified time without requiring operator manual navigation. It can even cruise in the air and wait for commands, which reduces response time and accomplishes tasks more efficiently. Therefore, it is important for UAV to confirm whether the command is issued by command center. It typically uses digital signature to confirm the source of command. The overall process is roughly as follows.

(i) A command center generates command and computes corresponding digital signature.

(ii) The command center sends the command and signature to UAV.

(iii) The UAV will verify the signature when it receives the command and signature. If the signature is valid, the UAV will believe that the command is issued by command center and execute this command. Otherwise, UAV will consider that the command is counterfeit and does not execute it.

However, common digital signature scheme is not suitable for UAV network because of its characteristics and security requirements. It needs to propose digital signature scheme for UAV network.

A UAV generally has very fast flight speed. For example, the maximum groundspeed of small UAV is 87 knots according to the rule of Federal Aviation Administration (FAA). The speed of large UAV is faster than small UAV. The loiter velocity of National Aeronautics and Space Administration (NASA) global hawk reaches 343 knots (true air speed). Therefore, a UAV is highly mobile, which requests that it determines whether the command is valid as soon as possible, especially for some location based services. Namely, it is necessary for the UAV to verify the signature timely. It is supposed that command center sends message and corresponding digital signature at t_{send} and UAV completes the signature verification at t_{ver}. Then, we can roughly compute the total delay time $delay_{tdt}=(t_{ver} - t_{send})$. The delay time $delay_{tdt}$ is mainly composed of propagation delay $delay_{ppd}$ and processing delay $delay_{pcd}$. Namely, $delay_{tdt} = delay_{ppd} + delay_{pcd}$.

(i) Propagation delay $delay_{ppd}$ is the delay time in which data is propagated from sender to receiver in the transmission medium. It is computed as $delay_{ppd} = dis/vel$ where dis is the distance between sender and receiver and vel is the propagation speed. It is difficult to improve the propagation speed. It can lower $delay_{ppd}$ by reducing dis. That is, the sender of command should be close to receiver (UAV).

(ii) Process delay $delay_{pcd}$ is mainly the delay time in which the receiver (UAV) verifies the digital signature. It should select efficient digital signature algorithm to lower $delay_{pcd}$.

From the above analysis, we find that it is necessary to shorten the distance between sender of command and UAV and select efficient digital signature algorithm in order to reduce total delay time $delay_{tdt}$.

However, it is inevitable for UAV to perform remote tasks. When the UAV is far from command center to perform tasks, it will experience longer propagation delay to receive command and corresponding signature and even cannot receive them due to weather or terrain. Mobile edge computing (MEC) technology can be introduced to solve this problem. MEC is proposed by European Telecommunication Standard Institute (ETSI). It can bring the function of commanding UAV to the edge of network, which is closer to UAV, to reduce $delay_{ppd}$. Meanwhile, it needs to use proxy signature scheme. The command center may temporarily authorize a command station near to UAV to command it, which can reduce the propagation delay. For example, if a ground control station is much closer to UAV than the command center, the command center may temporarily authorize ground control station to command UAV. The ground control station and UAV use proxy signature in this case. The command center delegates ground control station. The ground control station computes a private key for proxy signature, which allows it to sign on behalf of command center. Hence, the ground control station is able to command UAV.

In some cases, it also needs blind signature to protect user's privacy. It is supposed that there is a leasing company which rents UAVs to users. Namely, the company is lessor and user is lessee. The user rents a UAV to perform some tasks, such as taking pictures, surveying, and investigation. If the user does not want leasing company to know what commands the UAV executed to protect user's privacy, it will use blind signature scheme. Moreover, if the user is far from the command center of leasing company, the company will use proxy signature scheme to authorize a ground control station which is closer to the UAV to command it. Namely, it uses proxy blind signature scheme, which meets the requirements. We call this ground control station as local ground control station (LGCS). After the LGCS obtains authorization from command center of leasing company, the user will blind the command and send it to LGCS. The LGCS computes the signature of command blinded and sends it to the user. Afterwards, the user recovers the signature of original command, which is proxy blind signature of original command. It sends the proxy blind signature to UAV.

In addition, designated verifier signature is necessary. An adversary may eavesdrop on the messages exchanged among the LGCS, user, and UAV and verify the signature. It can infer the tasks that UAV will perform next and take precautions in advance. Therefore, it is necessary to ensure that the signature can be verified only by the designated verifier. It uses designated verifier signature scheme, which meets the requirement. For example, the user or LGCS sends command and corresponding signature to the UAV, while the signature can only be verified by the specified UAV. Even if an adversary eavesdropped on the command and corresponding signature, it could not verify the signature and confirm the task which will be performed next.

In summary, we draw the following conclusions through the above analysis.

(1) It is suitable to adopt MEC architecture for UAV network to reduce $delay_{ppd}$.

(2) The entity which issues and sends command should be as close as possible to the UAV. It helps to reduce the propagation delay $delay_{ppd}$. The command center uses proxy signature scheme to authorize a command station which is close to UAV to command it on behalf of command center.

(3) A user may not want command station to know what tasks it requires UAV to perform to protect the privacy. It uses proxy blind signature scheme to meet this requirement.

(4) The signature computed by command station can only be verified by the designated verifier, namely, the UAV. It uses designated verifier signature scheme to meet this requirement.

(5) It should use efficient cryptographic algorithm, which helps to reduce the processing delay $delay_{pcd}$.

Therefore, we propose a UAV network architecture based on MEC and a designated verifier proxy blind signature (DVPBS) scheme for the UAV network to meet the requirements above analyzed.

1.2. Our Contributions. For autonomous UAV, the command center or command station commands it by sending instruction and corresponding digital signature. The UAV verifies the signature and decides whether to execute corresponding command based on the verification result. If the signature is valid, the UAV will execute the command. Otherwise, it will refuse to execute the command.

In this paper, it is supposed that a user rents UAV from a leasing company to perform some tasks. The leasing company authorizes a LGCS to command the UAV. The user requires service of UAV through LGCS, and LGCS does not know what service the user requires. The command sent by LGCS can only be verified by the UAV. We make the following contributions for this scenario.

(i) We introduce MEC into UAV network and propose a UAV network architecture based on MEC with low latency.

(ii) We propose a designated verifier proxy blind signature scheme for UAV network based on MEC. This signature scheme is based on elliptic curve cryptography (ECC) which provides efficient computation.

(iii) We analyze the security of our DVPBS scheme based on the random oracle model. The result indicates that our DVPBS scheme is existentially unforgeable under an adaptive chosen message attack.

(iv) We implement simulation experiments of our DVPBS scheme and other signature schemes and theoretically analyze their communication cost. The experimental data indicates that our DVPBS is efficient in computation efficiency. The signature length is short compared with the transmission capacity of UAV.

1.3. Organization of the Remainder Paper. The rest of this paper is organized as follows. In the next section, we review MEC and some digital signature schemes which include blind signature scheme, proxy signature scheme, proxy blind signature scheme, designated verifier signature scheme, and designated verifier proxy signature scheme. In Section 3, we provide some necessary preliminaries. We propose a UAV network architecture based on MEC and a designated verifier proxy blind signature scheme for the UAV network in Section 4. We analyze the security and efficiency of our DVPBS scheme in Section 5. Finally, this paper is concluded in Section 6.

2. Related Work

2.1. Mobile Edge Computing. It is important for UAV network to reduce latency as much as possible. MEC can solve the problem of long latency resulting from long communication distance. ETSI proposed a framework and a reference architecture of MEC [13]. The MEC framework contains user equipment, mobile edge applications, hosts, networks, etc. These entities are divided into system level, host level, and network level. The reference architecture defines the reference points and functional elements which contain mobile edge system. Garg et al. proposed a data-driven transportation optimization model for surveillance in intelligent transportation system [14]. This model mainly contains UAV, dispatcher, aggregator, and edge devices. The UAV captures data from vehicles and validates the data. The dispatcher also validates the data and schedules the processing tasks in the edge computing devices. The aggregator provides secure data transmission and the edge devices perform data analysis. Lee et al. proposed a hierarchical MEC architecture [15]. It efficiently uses resource of MEC server and provides services according to the content type and computing type.

Intharawijitr et al. studied how to impact communication latency and computation latency [16]. They proposed a mathematical model of MEC to estimate the computing latency in edge node and developed three policies for selecting an edge node. Messous et al. proposed a game theory model where the players are a set of UAVs in the network [17]. The model helps UAVs to offload heavy computation tasks to achieve the tradeoff between energy overhead and execution delay. Ansari et al. proposed two dynamic proxy virtual machine migration methods which reduce the end-to-end delay between proxy virtual machine and device [18]. They validated the performance of two methods through simulation. Zhang et al. proposed a mobility-aware hierarchical MEC framework which contains MEC servers and backup computing server which shares computing tasks with MEC servers [19]. They developed an incentive-based optimal computation offloading scheme which reduces energy consumption and task execution time of smart devices.

2.2. Related Digital Signature Schemes. Chaum proposed blind signature scheme in 1983 [20]. In a blind signature scheme, a signer computes signature of the message blinded by provider and sends the signature to the provider of blind signature. The provider recovers the signature of original message from the signature received from signer. Blind signature can be used to establish untraceable payment system for e-commerce. Mambo et al. proposed proxy signature scheme in 1996 [21]. It allows an original signer to delegate a proxy

signer to sign on behalf of the original signer. Tan et al. proposed two proxy blind signature schemes which satisfy the secure properties of proxy signature and blind signature [22]. One is based on discrete logarithm problem and the other is based on elliptic curve discrete logarithm problem. Tan also proposed an efficient identity-based pairing-free proxy blind signature scheme [23]. It is provably secure in the random oracle model. Yang et al. proposed a proxy partially blind signature scheme, which can revoke proxy privileges and provide security features [24]. Verma et al. proposed a proxy blind signature scheme with message recovery [25]. It shortens the size of message signature and is suitable for the applications with low bandwidth. Zhu et al. proposed an efficient identity-based proxy blind signature scheme [26]. It is based on number theorem research unit lattice and can defeat quantum computer attack.

Jakobsson et al. proposed solutions of designated verifier proof which can be used to propose designated verifier signature scheme [27]. Dai et al. further proposed a designated-receiver proxy signature scheme [28]. It has the properties of designated verifier signature scheme and proxy signature scheme. An original signer delegates a proxy signer to sign on behalf of the original signer. Moreover, the signature computed by proxy signer can only be verified by the designated verifier. Huang et al. described the notion of short designated verifier proxy signature (ShDVPS) scheme and proposed a short DVPS scheme [29]. It has short signature length and suitable for the applications with low bandwidth. Shim proposed a short DVPS scheme which is based on BLS signature scheme and gave security proof in the random oracle model [30]. Islam et al. proposed an efficient identity-based strong designated verifier proxy signature (ID-SDVPS) scheme which is based on bilinear pairing [31]. There is a private key generator (PKG) to generate private keys for all entities. Hu et al. proposed a weak DVPS (WDBPS) scheme and a strong DVPS (StDVPS) scheme [32]. The former cannot compute a simulated designated verifier proxy signature, while the latter can compute such signature. They gave a formal security proof in the random oracle model.

3. Preliminaries

3.1. Complexity Assumptions. It is assumed that G_1 is a cyclic additive group and its order is a prime q. The following problems defined over an elliptic curve are assumed to be difficult to solve within polynomial time.

Assumption 1 (elliptic curve discrete logarithm problem (ECDLP)). Given P, Q $\in G_1$, find the integer $a \in Z_q^*$ so that $Q = aP$.

Assumption 2 (computational Diffie-Hellman (CDH) problem). Given a randomly chosen $P \in G_1$ and $aP, bP \in G_1$ for unknown $a, b \in Z_q^*$, compute abP.

Assumption 3 (decisional Diffie-Hellman (DDH) problem). Given a randomly chosen $P \in G_1$ and $aP, bP, cP \in G_1$ for unknown $a, b, c \in Z_q^*$, decide whether $c=ab$.

Assumption 4 (gap Diffie-Hellman (GDH) problem). Given a randomly chosen $P \in G_1$ and $aP, bP \in G_1$ for unknown $a, b, c \in Z_q^*$, solve CDH problem with the help of DDH oracle.

3.2. Outline of Designated Verifier Proxy Blind Signature Scheme. Generally, a DVPBS scheme consists of the following algorithms.

Setup. It takes the security parameter as input and outputs the system parameters.

Key Generation. It takes the security parameter as input and outputs some public-private key pairs (pk, sk) for the designated verifier, original signer, proxy signer, and blind signature provider.

Delegation Generation. Given the system parameter, private key of original signer, and warrant w, this algorithm outputs delegation d.

Delegation Verification. Given the system parameter, public key of original signer, and warrant w, this algorithm can determine whether the delegation is successful. If it is successful, it will compute the private key for proxy signature. Otherwise, it will require that the delegation generation algorithm is executed again.

Proxy Blind Signature Generation. It takes the system parameter, private key for proxy signature, and warrant w as inputs and outputs proxy blind signature.

Designated Verifier Proxy Blind Signature Generation. It takes the system parameter, warrant, private key for proxy signature, proxy blind signature, public key of designated verifier, and conversion of message as inputs and outputs the signature.

Designated Verifier Proxy Blind Signature Verification. It computes the public key for proxy signature. Afterwards, it takes the system parameter, public key for proxy signature, private key of designated verifier, warrant, message, and the signature as inputs and determines whether the signature is valid. It will return *True* if the signature is valid; otherwise, it will return $\perp$ which means termination.

Transcript Simulation. It takes the system parameter, warrant, private key of designated verifier, public key for proxy signature, and message as inputs and computes a simulated signature which is indistinguishable from the original designated verifier proxy blind signature.

3.3. Formal Security Notation. We provide a formal definition of existential unforgeability of our DVPBS scheme under an adaptive chosen message attack (EUF-CMA). It is defined using the following game between a challenger C and an adversary A.

Key Generation. Given the security parameter, C runs the algorithm to obtain public-private key pairs of original signer,

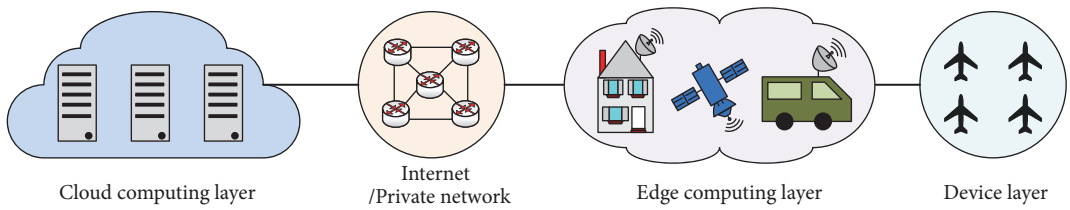

Figure 2: Our architecture of UAV network based on MEC.

proxy signer, provider of blind signature, and designated verifier.

DVPBS Sign Queries. When A requests a signature on a message m under a warrant w, C will run the DVPBS generation algorithm to generate a signature σ and return it to A.

DVPBS Verify Queries. When A requests a signature verification on a message m and a signature σ, C will responds with *True* if the signature is correct, or $\perp$ otherwise.

Output. Finally, A outputs a new pair (m^*, σ^*) where m^* has never been queried during the DVPBS sign queries and σ^* is a valid designated verifier proxy blind signature of message m^* under warrant w^*.

The success probability of an adversary to win the game is defined as $Succ_{DVPBS,A}^{EUF-CMA}$.

Define 1. A DVPBS scheme is existentially unforgeable under an adaptive chosen message attack if the success probability of any polynomially bounded adversary in the above game is negligible. That is, $Succ_{DVPBS,A}^{EUF-CMA} \leq \varepsilon$, where ε is negligible.

4. Designated Verifier Proxy Blind Signature for UAV Network

4.1. Architecture of UAV Network Based on MEC. The architecture of UAV network based on MEC is illustrated in Figure 2. This architecture is divided into three layers, cloud computing layer, edge computing layer, and device layer.

(i) *Cloud Computing Layer*. It has centralized cloud computing platform and sufficient resource in the cloud computing layer. It may consist of many servers. The command center of UAV network lies in this layer and is the original signer in our DVPBS scheme. It can directly command UAV and analyze the data sent back by UAV.

(ii) *Edge Computing Layer*. The cloud layer connects with edge computing layer through Internet or private network. This layer contains some command stations which may be authorized by command center to command the UAV, including satellites, ground control stations, and mobile command cars. These command stations can provide computation and storage services. They perform as virtual servers at the edge of network and are called edge servers. They are proxy signers in our DVPBS scheme and closer to UAV than command center, which is helpful to reduce latency.

(iii) *Device Layer*. It mainly contains UAV in the device layer. A UAV receives commands from command center in cloud computing layer or command stations in edge computing layer, performs tasks, and sends response.

4.2. Overview of DVPBS. It is assumed that a user rents UAV from leasing company to perform some tasks, such as taking pictures and surveying. The leasing company authorizes a LGCS which is close to the UAV to temporarily command UAV. Here leasing company, LGCS, user, and UAV are original signer (OS), proxy signer (PS), blind signature provider (BSP), and designated verifier (DV) for our DVPBS scheme, respectively. The brief process of our DVPBS lists as follows and is illustrated in Figure 3.

(1) Leasing company delegates a LGCS to sign on behalf of the company.

(2) LGCS verifies the delegation and computes a private key for proxy signature, sk_p.

(3) User blinds message m to obtain m_b.

(4) User sends m_b to LGCS.

(5) LGCS signs the blinded message, m_b, to obtain the signature $sig(sk_p, m_b)$ on behalf of leasing company.

(6) LGCS sends $sig(sk_p, m_b)$ to user.

(7) User recovers signature of message m from $sig(sk_p, m_b)$ to obtain $sig(sk_p, m)$ and transform m to $t(m)$, where the transformation is one way.

(8) User sends $t(m)$ and $sig(sk_p, m)$ to LGCS.

(9) LGCS computes the designated verifier proxy blind signature *dvpbs*.

(10) LGCS sends $t(m)$ and *dvpbs* to the user.

(11) User sends m and *dvpbs* to UAV.

(12) UAV computes the public key for proxy signature, PK_p, and verifies the signature *dvpbs* received from user.

4.3. System Initialization. In the system initialization phase, the leasing company sets some parameters. It chooses an elliptic curve over a prime finite field F_p where the variable p is a large prime. It uses the symbol G_1 to denote the cyclic additive group. It is assumed that G is the base point on the elliptic curve and a prime q is the order of G. The leasing company, LGCS, UAV, and user respectively choose their own private keys, sk_{LC}, sk_{LGCS}, sk_{UAV}, and $sk_U \in Z_q^*$. They correspondingly compute the public keys, $PK_{LC}=sk_{LC}G$, $PK_{LGCS}= sk_{LGCS}G=(x_{LGCS}, y_{GCS})$, $PK_{UAV} = sk_{UAV}G$, and $PK_U=sk_UG$. The leasing company also chooses

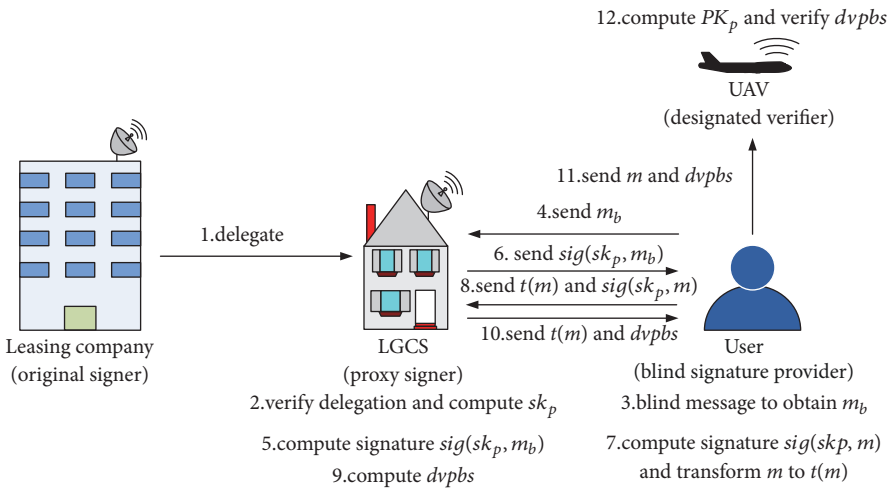

Figure 3: Brief process of DVPBS.

two cryptographic hash functions H_1: $\{0,1\}^* \longrightarrow Z_q^*$, H_2: $Z_q^* \times \{0,1\}^* \times G_1 \times G_1 \longrightarrow Z_q^*$.

4.4. Delegation Generation (DG) Phase. When a leasing company rents a UAV to a user, it will delegate a LGCS which is close to the UAV to command UAV. The leasing company performs the following steps to implement the delegation for LGCS.

(1) It generates a warrant w which explicitly describes the delegation relation for the LGCS.

(2) It chooses a random number $r_p \in Z_q^*$ and computes $R_p=r_p\mathrm{G} \pmod q=(x_{Rp}, y_{Rp})$.

(3) It computes $d=(sk_{LC}+r_p x_{Rp})H_1(w)$ and sends (d, w, R_p) to the LGCS.

4.5. Delegation Verification (DV) Phase. When the LGCS receives the message sent by leasing company, it will verify the delegation. If the delegation is invalid, the LGCS will request delegation again. If the delegation is valid, the LGCS will compute the private key for proxy signature. It performs the following steps to verify the delegation and compute private key.

(1) The LGCS computes and verifies whether $d\mathrm{G}=(PK_{LC}+x_{Rp}R_p)H_1(w)$ holds or not. If $d\mathrm{G}$ is not equal to $(PK_{LC}+x_{Rp}R_p)H_1(w)$, the delegation will fail and LGCS will request delegation from leasing company again. If they are equal, the delegation will success.

(2) The LGCS computes $sk_p=d+sk_{LGCS}x_{LGCS}H_1(w)$ and sets it as the private key for proxy signature. The LGCS uses this key to implement proxy signature and command the UAV on behalf of leasing company.

4.6. Proxy Blind Signature Generation (PBSG) Phase. The user rents UAV to complete some tasks and does not want the LGCS to know what mission UAV performs in order to protect privacy. The user uses blind signature in such case. It performs the following steps with LGCS to compute proxy blind signature.

(1) The LGCS chooses a random number $r_b \in Z_q^*$ and computes $R_b=r_b\mathrm{G} \pmod{\mathrm{q}}=(x_{Rb}, y_{Rb})$.

(2) The user chooses two random numbers $b_1, b_2 \in Z_q^*$ and computes $L=b_1\mathrm{G}+b_2R_b \bmod q=(x_L, y_L)$. It blinds the command message m as $m_b=b_2m \bmod q$ and sends m_b to the LGCS.

(3) The LGCS computes the signature of m_b as $s_b=(sk_p+r_bm_b)x_{Rb}^{-1} \bmod q$, namely, $x_{Rb}s_b=(sk_p+r_bm_b) \bmod q$. It sends $\{x_{Rb}s_b, R_p\}$ to user.

(4) The user obtains the proxy blind signature by computing $s_{pb}=(b_1m+x_{Rb}s_b)\,x_L^{-1} \bmod q$, namely, $x_Ls_{pb}=(b_1m+x_{Rb}s_b) \bmod q$. It sends $\{H_1(m), x_Ls_{pb}\}$ to the LGCS.

4.7. Designated Verifier Proxy Blind Signature Generation (DVPBSG) Phase. (1) The LGCS computes $s_{pb}x_L\mathrm{G}, K_{DVPBS} = sk_pPK_{UAV}$, and $s_{DVPBS}=H_2(H_1(m), w, s_{pb}x_L\mathrm{G}, K_{DVPBS})$ which is the designated verifier proxy blind signature. It sends $\{H_1(m), w, R_p, s_{DVPBS}\}$ to the user.

(2) The user sends $\{m, w, L, R_p, s_{DVPBS}\}$ to the UAV.

4.8. Designated Verifier Proxy Blind Signature Verification (DVPBSV) Phase. (1) The UAV computes the public key of proxy signature as $PK_p=(PK_{LC}+x_{Rp}R_p+x_{LGCS}PK_{LGCS})H_1(w) \bmod q$ and $T = PK_p+mL \bmod q$. It computes $K'_{DVPBS}= sk_{UAV}PK_p \pmod q$ and $s'_{DVPBS}=H_2(H_1(m), w, T, K'_{DVPBS})$.

(2) If $s'_{DVPBS}=s_{DVPBS}$, the UAV will determine that the signature is valid and perform corresponding tasks. Otherwise, it will think that the signature is invalid.

4.9. Transcript Simulation. The designated verifier, UAV, is also able to compute the signature s''_{DVPBS} which is indistinguishable from the signature generated by LGCS. The UAV computes s''_{DVPBS} as the following steps.

(1) It computes $PK_p = (PK_{LC} + x_{Rp}R_p + x_{LGCS}PK_{LGCS})H_1(w)$, $T'' = PK_p + mL \bmod q$, and $K''_{DVPBS}=sk_{UAV}PK_p \bmod q$.

(2) It computes $s''_{DVPBS}=H_2(H_1(m), w, T'', K''_{DVPBS})$ which is the result of transcript simulation.

5. Analysis of Our DVPBS Scheme

5.1. Correctness Proof. In this subsection, we provide the correctness proof of our DVPBS scheme.

(1) The LGCS computes a private key for proxy signature, $sk_p=d+sk_{LGCS}x_{LGCS}H_1(w)$. It is necessary for the UAV to compute corresponding public key, PK_p, to verify the signature. The UAV computes PK_p as follows.

$$\begin{aligned} PK_p &= \left(PK_{LC} + x_{Rp}R_p + x_{LGCS}PK_{LGCS}\right) H_1(w) \\ &= \left(PK_{LC} + x_{Rp}R_p\right) \mathrm{H}_1(w) + x_{LGCS}PK_{LGCS}H_1(w) \\ &= d\mathrm{G} + x_{LGCS}sk_{LGCS}\mathrm{G}H_1(w) \\ &= \left(d + sk_{LGCS}x_{LGCS}H_1(w)\right)\mathrm{G} = sk_pG \end{aligned} \tag{1}$$

Hence, the public key and private key for proxy signature meet the requirement $PK_p=sk_pG$.

(2) The $T=s_{pb}x_L\mathrm{G} \bmod q$ because we have

$$\begin{aligned} s_{pb}x_L\mathrm{G} &= \left(b_1m + x_{Rb}s_b\right)\mathrm{G} = \left(b_1m + sk_p + r_bm_b\right)\mathrm{G} \\ &= \left(b_1m + sk_p + r_bb_2m\right)\mathrm{G} \\ &= sk_p\mathrm{G} + mb_1\mathrm{G} + mb_2r_b\mathrm{G} \\ &= PK_p + m\left(b_1\mathrm{G} + b_2R_b\right) = PK_p + mL \bmod q \\ &= T \end{aligned} \tag{2}$$

(3) The $K_{DVPBS}=K'_{DVPBS}$ because we have

$$sk_pPK_{UAV} = sk_psk_{UAV}\mathrm{G} = sk_{UAV}PK_p \tag{3}$$

We have proved $PK_p = sk_pG$, $T = s_{pb}x_L\mathrm{G}$, and $K_{DVPBS} = K'_{DVPBS}$. Therefore, we draw the conclusion $s_{DVPBS} = s'_{DVPBS}$. Namely, we have proved the correctness of our DVPBS scheme.

5.2. Security Proof

Theorem 5. *Our DVPBS scheme is a designated verifier signature scheme.*

Proof. It needs to use the private key of UAV, sk_{UAV}, to verify the signature in the signature verification phase. Hence, a third party other than the signer and verifier cannot verify the validity or invalidity of this signature. The UAV can generate a valid signature by computing $s''_{DVPBS}=H_2(H_1(m), w, T'', sk_{UAV}PK_p)$, which is validated by verification algorithm and is indistinguishable from the signature generated by the LGCS. If the UAV does not generate the signature, it will believe that the signature is generated by proxy signer. Therefore, our DVPBS scheme is a designated verifier signature scheme. □

Theorem 6. *It is supposed that an EUF-CMA adversary* ***A*** *breaks our DVPBS scheme; namely, it can forge a valid signature of our DVPBS scheme, with success probability* $Succ^{EUF-CMA}_{DVPBS,A}$. *It makes* q_H *queries to the* H_2: $Z_q^* \times \{0,1\}^* \times G_1 \times G_1 \longrightarrow Z_q^*$, q_S *queries to the signing algorithm, and* q_V *queries to the verifying algorithm in polynomial time* t. *There is an algorithm* ***B*** *which uses* ***A*** *to solve an instance of the GDH problem with the probability* $Succ^{GDH}_B \geq Succ^{EUF-CMA}_{DVPBS,A} - q_V/(2^k - q_H - q_S)$ *where* k *is the system's security parameter.*

Proof. If there is an EUF-CMA adversary ***A*** who can forge a valid signature of our DVPBS scheme, we will prove that there is an algorithm ***B*** who can solve an instance of GDH problem. It is given a random instance (G, aG, bG) of GDH problem where $a, b \in Z_q^*$. The algorithm ***B*** can use the adversary ***A*** to obtain the value of abG with the DDH oracle. We regard the hash function H_2 as random oracle H. In the proof, the algorithm ***B*** will simulate all oracles and maintain a list, namely, *H-list*, to record the queries and corresponding responses. It is assumed that the adversary ***A*** never repeats the same query in the simulation. The algorithm ***B*** performs the following simulation.

Setup. The algorithm ***B*** will set private keys for original signer and designated verifier before the simulation begins. It thinks that the leasing company is original signer and UAV is designated verifier. It sets the private key of original signer as $sk_{LC} \in Z_q^*$ and computes its public key $PK_{LC}=sk_{LC}G$. Correspondingly, it sets the private key of designated verifier as $sk_{UAV} \in Z_q^*$ and computes its public key as $PK_{UAV} = sk_{UAV}G$. The forger ***A*** obtains the two public keys, PK_{LC} and PK_{UAV}. It can make query to the hash oracle, signing algorithm, and verifying algorithm. Finally, it outputs a valid signature σ^* of m^* which has never been queried in the signing oracle.

H-Queries. The algorithm ***B*** maintains *H-list* which records the queries and corresponding responses. This list comprises some tuples $(m_i, w_i, D_i, \sigma_i, coin)$, where m_i is the *i-th* message queried. The (m_i, w_i, D_i) is the input of H and σ_i is the output. When ***A*** queries the oracle H with (m_i, w_i, D_i), ***B*** will submit (aG, bG, D_i) to DDH oracle. If $D_i=abG$, then *coin* =1, otherwise *coin*=0. The DDH oracle determines whether $D_i=abG$ or not and responds to ***B*** with the result.

(1) If $D_i=abG$, ***B*** will set *coin* =1 and check the *H-list*.

(a) If there is a tuple $(m_i, w_i, \perp, \sigma_i, 1)$ in the *H-list*, ***B*** returns σ_i as the response to query of ***A***.

(b) If there is not a tuple $(m_i, w_i, \perp, \sigma_i, 1)$ in the *H-list*, ***B*** chooses a random $\sigma_i \in Z_q^*$. It is different from these σ_is which have been stored in the *H-list*. Afterwards, ***B*** adds the tuple $(m_i, w_i, D_i, \sigma_i, 1)$ to the *H-list* and responds σ_i to ***A***.

(2) If $D_i \neq abG$, ***B*** also chooses a random $\sigma_i \in Z_q^*$. It is different from these σ_is which have been stored in the *H-list*. Afterwards, ***B*** adds the tuple $(m_i, w_i, D_i, \sigma_i, 0)$ to the *H-list* and responds σ_i to ***A***.

Table 1: Communication cost of different signature schemes.

Signature scheme	Our DVPBS	Huang's ShDVPS	Shim's ShDVPS	Islam's ID-SDVPS	Hu's WDVPS	Hu's StDVPS
Signature length	$2\lvert G_1\rvert + 1\lvert Z_q^*\rvert$	$1\lvert Z_q^*\rvert$	$1\lvert G_2\rvert$	$1\lvert G_1\rvert + 2\lvert Z_q^*\rvert$	$2\lvert G_1\rvert$	$1\lvert G_1\rvert + 1\lvert G_2\rvert$

DVPBSG Queries. When $\boldsymbol{A}$ makes *DVPBSG* queries with m_i, $\boldsymbol{B}$ will check the *H-list*.

(1) If there is a tuple $(m_i, w_i, D_i, \sigma_i, 1)$ in the *H-list*, $\boldsymbol{B}$ will output σ_i to $\boldsymbol{A}$ as the signature.

(2) If there is not such a tuple in the *H-list*, $\boldsymbol{B}$ will choose a random $\sigma_i \in Z_q^*$. It adds the tuple $(m_i, w_i, \perp, \sigma_i, 1)$ to the *H-list* and returns σ_i to $\boldsymbol{A}$ as the signature.

DVPBSV Queries. When $\boldsymbol{A}$ makes *DVPBSV* queries with (m_i, σ_i), $\boldsymbol{B}$ will check the *H-list*.

(1) If there is not a σ_i in the *H-list* which is equal to the σ_i queried by $\boldsymbol{A}$, $\boldsymbol{B}$ will reject the σ_i queried by $\boldsymbol{A}$ as a valid signature.

(2) If there is a σ_i in the *H-list* which is equal to the σ_i queried by $\boldsymbol{A}$, $\boldsymbol{B}$ will continue to check the form of tuple in which the σ_i is located. If the σ_i is located in the tuple which has the form of $(m_i, w_i, \perp, \sigma_i, 1)$ or $(m_i, w_i, D_i, \sigma_i, 1)$, $\boldsymbol{B}$ will accept it as a valid signature. Otherwise, $\boldsymbol{B}$ will refuse to consider it is a valid signature.

When σ_i is a valid signature of m_i and σ_i is not queried from the oracle H, there will be a difference between $Succ_{DVPBS,A}^{EUF-CMA}$ and $Succ_B^{GDH}$. It happens with the possibility less than $1/(2^k - q_H - q_S)$ because the output of H is uniformly distributed; namely, $Succ_{DVPBS,A}^{EUF-CMA} - Succ_B^{GDH} \leq 1/(2^k - q_H - q_S)$. The forger $\boldsymbol{A}$ can perform q_V verifying algorithm. We will obtain $Succ_{DVPBS,A}^{EUF-CMA} - Succ_B^{GDH} \leq q_V/(2^k - q_H - q_S)$ when we sum up all q_V DVPBSV queries.

Output. After the above queries are completed, $\boldsymbol{A}$ outputs a new valid signature σ^* of message m^* which has never been queried during the *DVPBSG* queries. The signature σ^* is returned as the hash value of $\boldsymbol{A}$'s query (m^*, D^*). Namely, there is a tuple $(m^*, w^*, D^*, \sigma^*, 1)$ in the *H-list* and $D^* = abG$ in this tuple. Therefore, $\boldsymbol{B}$ successfully solves an instance of the GDH problem with the probability

$$Succ_B^{GDH} \geq Succ_{DVPBS,A}^{EUF-CMA} - \frac{q_V}{2^k - q_H - q_S}. \quad (4)$$

□

Theorem 7. *The designated verifier can distinguish the proxy signature generated by proxy signer from the signature generated by original signer. Namely, our DVPBS scheme has distinguishability.*

Proof. After the proxy signer successfully performs the delegation verification, it will compute the private key for proxy signature $sk_p = d + sk_{LGCS} x_{LGCS} H_1(w)$. This key is only obtained by the proxy signer, LGCS, because it needs to use the private key of proxy signer to compute sk_p. The original signer cannot compute s_{DVPBS} or s''_{DVPBS} because it does not obtain sk_p and sk_{UAV}. Therefore, it can neither compute the proxy signature nor simulate it. If the designated verifier, UAV, believes that a signature is valid, it can distinguish that the signature was generated by proxy signer or original signer. Hence, our DVPBS scheme has distinguishability. □

Theorem 8. *The proxy signer does not know the original message which has been signed by it. Namely, our DVPBS scheme has blindness.*

Proof. In our DVPBS scheme, the blind signature provider chooses a random number and uses it to blind the original message. According to the process of blinding message, the proxy signer cannot recover the original message from the message blinded. Moreover, it also does not receive or infer the original message from the following communication content. Hence, the proxy signer does not know the original message. Namely, our DVPBS scheme has blindness. □

5.3. Efficiency Analysis. In this section, we compare the efficiency of our DVPBS scheme with some other related digital signature schemes. We mainly compare them from two aspects. One is the time spent in the process of signature computation and the other is the length of signature.

Firstly, we implement our DVPBS scheme and some other related digital signature schemes in a PC with Intel i5-4590 CPU and 4 GB RAM. We use the Java pairing-based cryptography (jPBC) library which is developed by Caro et al. and select the type *A* elliptic curve [33]. In our experiment, we obtain the time cost for computation in the delegation generation phase, delegation verification phase, DVPBSG phase, and DVPBSV phase of different signature schemes. The results are illustrated in Figure 4.

From the experimental data, we find that the time spent in the delegation generation phase of our DVPBS scheme is the shortest, while the time cost in the delegation verification phase is longer than other schemes. Both the time spent in the DVPBSG phase and the time spent in the DVPBSV phase are longer than Huang's ShDVPS [29] and Islam's ID-SDVPS [31], while shorter than Shim's ShDVPS [30] and Hu's WDVPS and StDVPS [32]. We sum the time spent in delegation generation phase, delegation verification phase, DVPBSG phase, and DVPBSV phase of different signature schemes and obtain the results, which is illustrated in Figure 5. The results show that the total time of our DVPBS scheme is longer than Islam's ID-SDVPS and shorter than Huang's ShDVPS, Shim's ShDVPS, and Hu's WDVPS and StDVPS. Hence, we think that our PVDBS is efficient.

Secondly, we theoretically analyze the signature length of different signature schemes and list them in Table 1. The shorter the signature length is, the less the time and energy are taken to send and receive signature. The result indicates that the signature length of our DVPBS scheme is longer, but

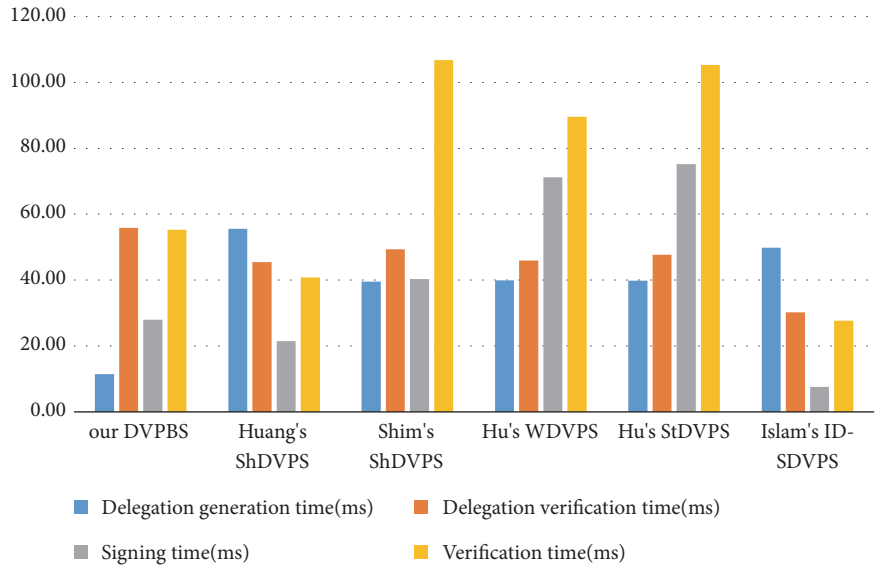

Figure 4: Experiment results of different signature schemes.

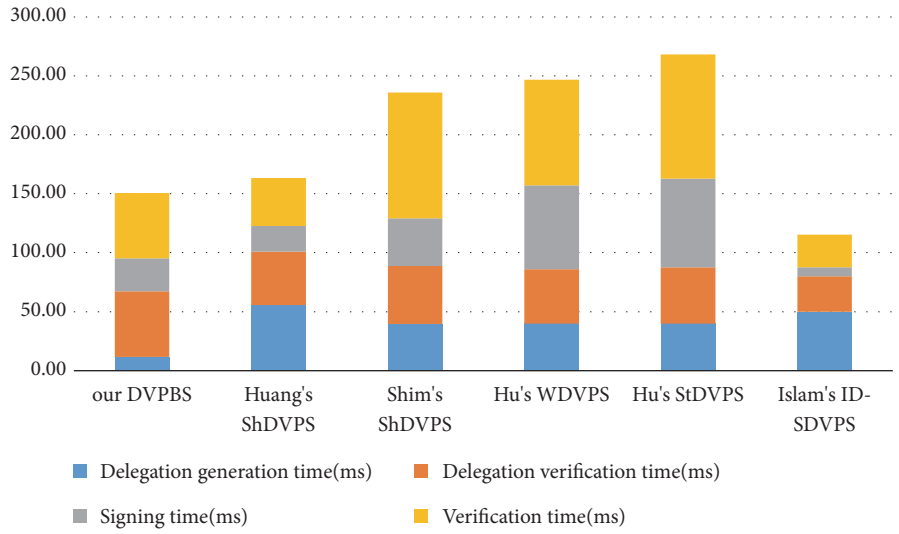

Figure 5: Total time of different signature schemes.

it is still short compared with the transmission capacity of UAV.

6. Conclusion

A UAV network consists of UAV, command center, and some command stations. It takes UAV application as its core. According to the characteristics of UAV, UAV network is sensitive to latency. Moreover, it is important to protect the integrity and authentication of commands which are sent by command center and some command stations to UAV. In this paper, we proposed a UAV network architecture based on MEC with low latency. We also proposed a DVPBS scheme for the scenario of UAV rental. In this scenario, a user rents UAV from a leasing company. We proved the security of our DVPBS scheme in the random oracle model. It is existentially unforgeable under an adaptive chosen message attack. Moreover, we think that our DVPBS scheme has distinguishability and blindness. We compared the efficiency of our DVPBS with some other signature schemes by simulation experiments and theoretical analysis. The experimental results indicate that our DVPBS scheme is efficient. Through theoretical analysis, the signature length of our DVPBS scheme is longer than those of other schemes, but it is still short compared with the transmission capacity of UAV. We will research how to further reduce the time spent in signature processing and shorten the length of signature in the future.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the Key Program of NSFC Grant (U1405255), Shaanxi Science & Technology Coordination & Innovation Project (2016TZC-G-6-3), and the Fundamental Research Funds for the Central Universities (SA-ZD161504).

References

[1] S. J. Kim and G. J. Lim, "Drone-aided border surveillance with an electrification line battery charging system," *Journal of Intelligent & Robotic Systems*, vol. 92, no. 3-4, pp. 657–670, 2018.

[2] Z. Wang, L. Liu, T. Long, and Y. Wen, "Multi-UAV reconnaissance task allocation for heterogeneous targets using an opposition-based genetic algorithm with double-chromosome encoding," *Chinese Journal of Aeronautics*, vol. 31, no. 2, pp. 339–350, 2018.

[3] G. Sona, D. Passoni, L. Pinto et al., "UAV multispectral survey to map soil and crop for precision farming applications," in *Proceedings of the 23rd International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences Congress, ISPRS 2016*, vol. 41, pp. 1023–1029, Czech Republic, July 2016.

[4] P. Nintanavongsa and I. Pitimon, "Impact of sensor mobility on UAV-based smart farm communications," in *Proceedings of the 2017 International Electrical Engineering Congress, iEECON 2017*, pp. 1–4, March 2017.

[5] M. Pircher, J. Geipel, K. Kusnierek et al., "Development of a hybrid UAV sensor platform suitable for farm-scale applications in precision agriculture," in *Proceedings of the International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, p. 297, 2017.

[6] T. Ahn, J. Seok, I. Lee, and J. Han, "Reliable flying IoT networks for UAV disaster rescue operations," *Mobile Information Systems*, vol. 2018, Article ID 2572460, 12 pages, 2018.

[7] A. M. Hayajneh, S. A. R. Zaidi, D. C. McLernon, M. Di Renzo, and M. Ghogho, "Performance analysis of UAV enabled disaster recovery networks: a stochastic geometric framework based on cluster processes," *IEEE Access*, vol. 6, pp. 26215–26230, 2018.

[8] M. Deruyck, J. Wyckmans, W. Joseph, and L. Martens, "Designing UAV-aided emergency networks for large-scale disaster scenarios," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, p. 79, 2018.

[9] J. Lyu, Y. Zeng, and R. Zhang, "UAV-aided offloading for cellular hotspot," *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 3988–4001, 2018.

[10] Y. Wu, Y. Qin, Z. Wang, and L. Jia, "A UAV-based visual inspection method for rail surface defects," *Applied Sciences*, vol. 8, no. 7, p. 1028, 2018.

[11] P. Addabbo, A. Angrisano, M. L. Bernardi et al., "UAV system for photovoltaic plant inspection," *IEEE Aerospace and Electronic Systems Magazine*, vol. 33, no. 8, pp. 58–67, 2018.

[12] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1123–1152, 2016.

[13] M. ETSI, "Mobile edge computing (mec); framework and reference architecture," *ETSI, DGS MEC*, vol. 3, 2016.

[14] S. Garg, A. Singh, S. Batra, N. Kumar, and L. T. Yang, "UAV-empowered edge computing environment for cyber-threat detection in smart vehicles," *IEEE Network*, vol. 32, no. 3, pp. 42–51, 2018.

[15] J. Lee and J. Lee, "Hierarchical mobile edge computing architecture based on context awareness," *Applied Sciences*, vol. 8, no. 7, p. 1160, 2018.

[16] K. Intharawijitr, K. Iida, and H. Koga, "Simulation study of low latency network architecture using mobile edge computing," *IEICE Transaction on Information and Systems*, vol. E100D, no. 5, pp. 963–972, 2017.

[17] M.-A. Messous, H. Sedjelmaci, N. Houari, and S.-M. Senouci, "Computation offloading game for an UAV network in mobile edge computing," in *Proceedings of the 2017 IEEE International Conference on Communications, ICC 2017*, pp. 1–6, May 2017.

[18] N. Ansari and X. Sun, "Mobile edge computing empowers internet of things," *IEICE Transactions on Communications*, vol. E101B, no. 3, pp. 604–619, 2018.

[19] K. Zhang, S. Leng, Y. He, S. Maharjan, and Y. Zhang, "Mobile edge computing and networking for green and low-latency internet of things," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 39–45, 2018.

[20] D. Chaum, "Blind signatures for untraceable payments," *Advances in Cryptology*, pp. 199–203, 1983.

[21] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures for delegating signing operation," in *Proceedings of the 3rd ACM Conference on Computer and Communications Security*, pp. 48–56, ACM Press, March 1996.

[22] Z. Tan, Z. Liu, and C. Tang, "Digital proxy blind signature schemes based on DLP and ECDLP," *MM Research Preprints*, vol. 21, pp. 212–217, 2002.

[23] Z. Tan, "Efficient pairing-free provably secure identity-based proxy blind signature scheme," *Security and Communication Networks*, vol. 6, no. 5, pp. 593–601, 2013.

[24] F.-Y. Yang and L.-R. Liang, "A proxy partially blind signature scheme with proxy revocation," *Journal of Ambient Intelligence and Humanized Computing*, vol. 4, no. 2, pp. 255–263, 2013.

[25] G. K. Verma and B. B. Singh, "Efficient message recovery proxy blind signature scheme from pairings," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 11, Article ID e3167, 2017.

[26] H. Zhu, Y.-A. Tan, L. Zhu, Q. Zhang, and Y. Li, "An efficient identity-based proxy blind signature for semioffline services," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 5401890, 9 pages, 2018.

[27] M. Jakobsson, K. Sako, and R. Impagliazzo, "Designated verifier proofs and their applications," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 143–154, 1996.

[28] J. Z. Dai, X. H. Yang, and J. X. Dong, "Designated-receiver proxy signature scheme for electronic commerce," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 384–389, 2003.

[29] X. Huang, Y. Mu, W. Susilo, and F. Zhang, "Short designated verifier proxy signature from pairings," in *Proceedings of the International Conference on Embedded and Ubiquitous Computing*, pp. 835–844, 2005.

[30] K.-A. Shim, "Short designated verifier proxy signatures," *Computers and Electrical Engineering*, vol. 37, no. 2, pp. 180–186, 2011.

[31] S. H. Islam and G. Biswas, "A provably secure identity-based strong designated verifier proxy signature scheme from bilinear pairings," *Journal of King Saud University—Computer and Information Sciences*, vol. 26, no. 1, pp. 55–67, 2014.

[32] X. Hu, W. Tan, H. Xu, and J. Wang, "Short and provably secure designated verifier proxy signature scheme," *IET Information Security*, vol. 10, no. 2, pp. 69–79, 2016.

[33] A. de Caro and V. Iovino, "jPBC: Java pairing based cryptography," in *Proceedings of the 16th IEEE Symposium on Computers and Communications (ISCC '11)*, pp. 850–855, July 2011.