Privacy-preserving Mutual Authentication Protocol for Drone Delivery Services

Rohini Poolat Parameswarath
Department of ECE
College of Design and Engineering
National University of Singapore
Singapore
rohini.p@nus.edu.sg

Biplab Sikdar
Department of ECE
College of Design and Engineering
National University of Singapore
Singapore
bsikdar@nus.edu.sg

***Abstract*—Drones are becoming popular in a variety of applications. One of them is to collect packages from sellers and deliver them to buyers who are connected through a marketplace platform. However, drones are also vulnerable to cyberattacks. Drone delivery service also brings up privacy concerns about the personal information of users of the marketplace. This paper addresses such security and privacy threats and proposes a privacy-preserving authentication protocol for drone delivery services. The proposed protocol is built on privacy-preserving Decentralized Identifiers (DIDs) and Verifiable credentials (VCs). The use of DID helps marketplace users to preserve their privacy rights and enables them to request drone services in a privacy-preserving manner. At the same time, their legitimacy can be verified by the drones using VCs. The proposed protocol also incorporates an efficient dynamic revocation mechanism to remove the marketplace users from the service subscription if required.**

***Index Terms*—Decentralized identifier, drones, mutual authentication, privacy, security, verifiable credential, unmanned aerial vehicles (UAVs).**

I. Introduction

Unmanned Aerial Vehicles (UAVs), often known as drones, can be deployed as aerial base stations or relays to provide wireless connectivity from the sky. They can also be used for surveillance or for delivering packages. Drone delivery of goods has become increasingly popular in recent years. Drone delivery offers key benefits such as increased speed and efficiency, improved access to remote areas, and lower carbon footprint. Overall, drone delivery has the potential to revolutionize the way goods are delivered to customers, making it faster, cheaper, and more efficient while also reducing our impact on the environment [1]. Amazon and UPS have major plans to employ drones to deliver items to their customers [1].

In a typical drone delivery service as given in [2], sellers and buyers are connected through a marketplace platform provider. A seller requests the marketplace platform provider to send a drone to pick up the package. The marketplace provider sends a drone to the docking station closest to the seller. The seller loads the package onto the arrived drone. However, there are security and privacy concerns associated with such a service. A malicious party may send a drone to impersonate a legitimate one to steal goods from the seller [2]. Hence, it is important to authenticate the drone before loading the package. Since all information is exchanged through an insecure medium (the Internet), an adversary may carry out various attacks on communication channels and gather personal information about the seller.

To provide security and privacy protection for sellers of drone delivery services, we propose a privacy-preserving mutual authentication protocol in this paper built on the concepts of Decentralized Identifier (DID) and Verifiable Credential (VC). DID is a verifiable, decentralized digital identity that was standardized by the World Wide Web Consortium (W3C) DID working group [3]. DID enables users to create and manage their identities without depending on a centralized authority. A VC is a digital equivalent of a physical credential that can be cryptographically verified [4]. It contains information related to certain attributes about the "Holder" of the VC. This information is stated by a trusted authority and can be cryptographically verified by another party.

A. Related Work

We now present the related work on authentication in drone services and on DIDs and VCs.

Authentication in Drone Services: Alladi et al. presented an authentication protocol for UAV to ground station and UAV to UAV communication in [5]. However, their protocol requires each UAV to be equipped with additional hardware (a Physical Unclonable Function (PUF)). The authors of [6] proposed an authenticated key exchange protocol for the Internet of Drones (IoD) environment. Hussain et al. also proposed an authentication scheme for the IoD environment in [7]. The protocol in [7] is based on elliptic curve cryptography. In addition to authentication in the IoD environment, researchers have been working on authentication in drone delivery services as well. Blockchain-based authentication mechanisms for drone delivery services were proposed in [8] and [9]. Researchers have also investigated drone fingerprinting as a means of drone delivery authentication. As an example, Ramesh et al. [2] proposed a delivery drone authentication mechanism using acoustic noise fingerprinting.

979-8-3503-1090-0/23/$31.00 ©2023 IEEE

They leveraged the fact that the manufacturing defects in the motors of drones make the acoustic noise unique to each drone. However, this method only supports the authentication of drones; it does not support mutual authentication. Also, it does not consider the privacy of the customers. Wu et al. developed a secure mutual authentication method in [10] based on the hand movement of the user. According to the authors of [11], the background might affect a phone's object tracking in [10] and they proposed an authentication scheme for drone delivery using face biometrics [11]. There are privacy concerns for users in the schemes proposed in [10] and [11],

DIDs and VCs: The concepts of DID and VC have been gaining popularity in a variety of fields as viable privacy preservation methods. The authors of [12] analyzed the resource requirements of using DIDs on resource-constrained IoT devices. Their research showed that the deployment of DID on IoT devices helps to improve the privacy of the users considerably. The usage of DID and VC in certifying individuals who have received Covid vaccination was proposed in [13]. Since the Covid vaccination certificate can be verified cryptographically, any interested party can confirm that a person has received a Covid vaccination while protecting the user's privacy. All these works in different domains ranging from IoT to healthcare show that DID and VC can improve the privacy of the users significantly.

B. Motivation and Contributions

Mutual authentication of drones and customers in drone delivery services is essential so that both parties can ensure that they are communicating with the right party. Also, an attacker should not be able to extract personal information about the user by eavesdropping the exchanged messages. Hence, we need a privacy-preserving mutual authentication protocol for drone delivery services. The marketplace users' identities should be used only with their approval, and they should decide who can use it for what purpose. This will ensure a high level of privacy. Though users do not reveal their real identities, the drone should be able to verify users' legitimacy. There should also be an option to remove users from the service if required.

Motivated by the above requirements, this paper makes the following major contributions:

1. A privacy-preserving mutual authentication protocol for drone delivery services based on DID and VC: We propose a new protocol for drone delivery services by combining the concepts of DID and VC. The proposed protocol enables marketplace platform users to create and manage their IDs without depending on any third party. By employing DID and VC, the users' identities can be used only with their approval, and they decide who can use it for what purpose, ensuring a high level of privacy. Though the users do not reveal their real identities, the drone can verify their legitimacy by verifying their VCs. Hence, the proposed protocol ensures the privacy of the users and mutual authentication between the users and drones along with several other security properties.

2. User revocation scheme: There may be marketplace platform users who do not need the service anymore and have left the service subscription. The marketplace platform provider may also want to remove those users who do not meet certain quality criteria. The proposed protocol incorporates a revocation scheme using a dynamic universal accumulator to remove such users. The users must prove that they are not members of the revocation list to access the service.

3. Security analysis: We provide informal security analysis to demonstrate that the proposed scheme ensures privacy and provides several security features.

4. Performance analysis: We provide a performance analysis of the proposed protocol to show that it is computationally efficient.

The rest of the paper is organized as follows. In Section II, we discuss the preliminaries. In Section III, the system and adversary models are presented. In Section IV, we present the proposed protocol. We discuss the security analysis in Section V and performance analysis in Section VI, respectively. In Section VII, conclusions are given.

II. Preliminaries

In this section, we discuss the building blocks of the proposed protocol.

A. Decentralized Identifier

DID is a type of identifier that is created and managed by its owner without relying on another party. The entity identified by the DID is called the 'DID subject'. DID enables the creation of decentralized digital identities [3]. A DID maps to a DID document that can be hosted on public ledgers such as blockchains. It acts like a key-value database, where DIDs are the keys and the DID documents are the values. DID resolution is the process of mapping a DID to the corresponding DID document. This is carried out by a component called DID resolver. The DID document contains information such as the public key that is required to authenticate the DID subject [3].

B. Verifiable Credential

A verifiable credential is a set of claims that can be verified cryptographically [4]. A trusted **Issuer** signs credentials about the **Holder** of the VC. Since a digital signature is used, VCs are tamper-resistant, credible, and can be verified digitally by others. The **Holder** presents the VC to another party, the **Verifier**, to prove that he/she possesses the required credentials. The **Verifier** can verify the statements about the **Holder** [4] cryptographically.

C. Revocation Using Accumulator

An accumulator scheme helps to hash many values into a single short value. This resultant value is called the accumulator. For values included in the accumulator, there exists a witness [14]. In a dynamic accumulator, the members can be added or removed dynamically [14]. The authors of [15] proposed a dynamic universal accumulator that supports non-membership witness. From time to time, certain marketplace platform users leave the service subscription due to various reasons. To remove the people who do not need the service, the proposed protocol incorporates a revocation mechanism using universal accumulators [14], [15]. Since the number of people joining a network is typically more than the number of people leaving, maintaining an accumulator for those who leave is more efficient. The accumulator stores the list of revoked users. The users must show a non-membership witness before requesting a service. The functionalities of the accumulator used in the proposed protocol are given below [15]:

Generation of an Accumulator: We denote the accumulator generation function as $Acc_{Gen}()$. It takes a secret key k_{acc} and the revocation list R as inputs and generates an accumulator Acc. Generation of accumulator can be expressed as $Acc \leftarrow Acc_{Gen}(k_{acc}, R)$.

Updating an Accumulator: The inputs are the current accumulator Acc, the secret key k_{acc}, and a new value r_{new} to be added to the accumulator. The output of this function $Acc_{upd}()$ is the updated accumulator Acc_{new}. This function can be expressed as $Acc_{new} \leftarrow Acc_{upd}(Acc, k_{acc}, r_{new})$.

Witness Generation: The inputs are current accumulator Acc, the secret key k_{acc}, the revocation list R, and a value v that is not in R. The output of this function $Gen_{witness}()$ is the non-membership witness w for v. This function can be expressed as $w \leftarrow Gen_{witness}(Acc, k_{acc}, R, v)$.

Witness Verification: The inputs for this function $Ver_{witness}()$ are the current accumulator Acc, a non-membership witness w for the value v, and the value v. The output is either 0 or 1. If v is not in the revocation list R, $Ver_{witness}$ outputs 1. If v is in R, $Ver_{witness}$ outputs 0. This function can be expressed as $0/1 \leftarrow Ver_{witness}(Acc, w, v)$.

III. System and Adversary Models

A. System Model

The system model considered in this paper is shown in Figure 1. We consider a drone-assisted delivery service supporting a consumer-to-consumer marketplace platform. On the package collection side of the model, there are three major participants: the marketplace platform provider ($\mathcal{MP}$), the drones ($\mathcal{D}$), and the sellers ($\mathcal{S}$). The $\mathcal{MP}$ is in charge of operating and maintaining the platform to buy and sell items and to provide drone delivery services to its users. It has a data centre to store information about transactions and deliveries. The seller drops the package at a nearby docking station. There are several drones to pick up packages from docking stations and carry them to warehouses. The sellers communicate with the drones and the $\mathcal{MP}$ using their mobile devices through the Internet. The $\mathcal{MP}$, the drones, and the sellers create their DIDs. The private key of the seller corresponding to his/her DID is stored on his/her mobile device. The DID documents corresponding to the DIDs are stored on the blockchain. The DID document stores the DID holder's public key.

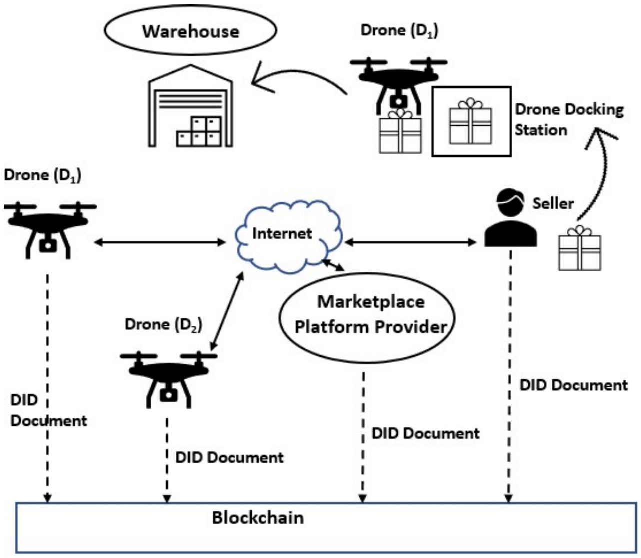

Fig. 1. System model.

B. Adversary Model

In our system model, the participants communicate with each other through an insecure channel, the Internet. In our adversary model, we consider the following threats:

Data Modification Threat: An adversary who is able to control the communication channels may eavesdrop and edit the exchanged messages. Hence, there is a threat of data modification.

Privacy Threat: The adversary may eavesdrop on exchanged messages and identify the real identity, location details, and trajectory of the sellers. This poses a privacy threat.

Service Access Threat: The attacker may capture the exchanged messages and send them later to get authenticated as a registered seller. Also, the attacker may generate messages to impersonate a registered seller. A seller who unsubscribes from the service also can be dishonest and may try to access the service with the previous credentials.

Drone Impersonation Threat: An attacker may send a drone impersonating an authorized drone to pick up the package from a seller.

IV. Proposed Authentication Protocol

We now present the protocol for drone delivery services. The proposed scheme consists of setup, registration, mutual authentication, and revocation phases.

A. Assumptions

We assume that the communication channels between the entities are secure during the registration phase.

B. Setup Phase

Step 1: The $\mathcal{MP}$ generates its private key $Pr_{\mathcal{MP}}$ and public key $Pu_{\mathcal{MP}}$.

Step 2: The $\mathcal{MP}$ generates a revocation list R and a secret key k_{acc}. Initially, there are no elements in R. As mentioned in Section II-C, the $\mathcal{MP}$ generates an accumulator Acc from k_{acc} and the revocation list R using the $Acc_{Gen}()$ function.

C. Registration Phase

In the registration phase, the sellers and the drones register with the $\mathcal{MP}$.

TABLE I
SELLER REGISTRATION PHASE

Seller	Marketplace Platform Provider
Generate: $did_{\mathcal{S}_i}$, $Pr_{\mathcal{S}_i}$, $Pu_{\mathcal{S}_i}$ $P = \{pid^1_{\mathcal{S}_i}, pid^2_{\mathcal{S}_i}, \ldots\ldots pid^n_{\mathcal{S}_i}\}$ $M_1 = \{Reg_{req}, did_{\mathcal{S}_i}, P, VC_{IDSi}\}$	
$\boxed{\xrightarrow{M_1}}$	
	Verify: VC_{IDSi} Generate: cred $[VC_{\mathcal{S}_i}] = sign[cred]_{Pr_{\mathcal{MP}}}$ w_i = $Gen_{witness}(R, Acc, \mathcal{S}_i, k_{acc})$ Store: $did_{\mathcal{S}_i}$, P, cred, $VC_{\mathcal{S}_i}$, w_i $M_2 = \{VC_{\mathcal{S}_i}, w_i\}$
Store: $VC_{\mathcal{S}_i}$, w_i	$\boxed{\xleftarrow{M_2}}$

1) Seller Registration: The steps involved in seller registration are given below:

Step 1: The seller $\mathcal{S}_i$ generates a DID, $did_{\mathcal{S}_i}$, and the corresponding pair of private ($Pr_{\mathcal{S}_i}$) and public ($Pu_{\mathcal{S}_i}$) keys. $\mathcal{S}_i$ also generates a set of n pseudo-DIDs $P = \{pid^1_{\mathcal{S}_i}, pid^2_{\mathcal{S}_i}, ..., pid^n_{\mathcal{S}_i}\}$. Then, $\mathcal{S}_i$ stores the DID documents corresponding to the DIDs on the blockchain. $\mathcal{S}_i$ stores the private keys corresponding to the DIDs in the digital wallet on his/her mobile device $MD_{\mathcal{S}_i}$. The public keys are stored in the DID documents corresponding to the DIDs on the blockchain. $\mathcal{S}_i$ holds VC_{IDSi}, the VC of his/her national identity document issued by an official authority. $\mathcal{S}_i$ composes a message M_1 with a registration request to join the marketplace platform, $did_{\mathcal{S}_i}$, the set of pseudo-DIDs P, and VC_{IDSi}. After that, $\mathcal{S}_i$ sends M_1 to the $\mathcal{MP}$.

Step 2: Upon receiving M_1, the $\mathcal{MP}$ verifies the signature on VC_{IDSi}. Then, the $\mathcal{MP}$ generates a credential $cred$ for $\mathcal{S}_i$. After that, the $\mathcal{MP}$ signs $cred$ with its private key $Pr_{\mathcal{MP}}$ to generate the VC, $VC_{\mathcal{S}_i}$, of the seller. The $\mathcal{MP}$ also generates a non-membership witness w_i for $\mathcal{S}_i$ by calling $Gen_{witness}()$ with inputs R, Acc, $\mathcal{S}_i$, and k_{acc} as mentioned in Section II-C. Then, the $\mathcal{MP}$ stores $did_{\mathcal{S}_i}$, P, $cred$, $VC_{\mathcal{S}_i}$, and w_i. Subsequently, the $\mathcal{MP}$ composes a message M_2 with $VC_{\mathcal{S}_i}$ and w_i and sends M_2 to $\mathcal{S}_i$.

Step 3: Upon receiving M_2, $\mathcal{S}_i$ stores $VC_{\mathcal{S}_i}$ and w_i in the digital wallet on $MD_{\mathcal{S}_i}$. The steps involved in the seller registration phase are given in Table I.

2) Drone Registration: The drones also register with the $\mathcal{MP}$. After registration, the drone $\mathcal{D}_j$ receives a VC, $VC_{\mathcal{D}_j}$, from the $\mathcal{MP}$.

TABLE II
MUTUAL AUTHENTICATION PHASE

Seller		Marketplace Platform Provider
$A_1 = \{pid^1_{\mathcal{S}_i}, L_i, Req\}$		
	$\boxed{\xrightarrow{A_1}}$	
		$A_2 = \{OK, L^*_i, \mathcal{D}_j\}$
	$\boxed{\xleftarrow{A_2}}$	
Store: L^*_i		
Seller		**Drone**
A_3= $\{pid^1_{\mathcal{S}_i}$, $VC_{Req}\}$		
	$\boxed{\xrightarrow{A_3}}$	
		Generate: N_j A_4= $\{[did_{\mathcal{D}_j}$, $VC_{\mathcal{D}_j}$, Req_{VC}, Req_w, $N_j]_{Pu_{pid^1_{\mathcal{S}_i}}}\}$
		$\boxed{\xleftarrow{A_4}}$
Decrypt: A_4 using $Pr_{pid^1_{\mathcal{S}_i}}$ Verify: $VC_{\mathcal{D}_j}$ using $Pu_{\mathcal{MP}}$ Generate: N_i Calculate: K_S= $did_{\mathcal{D}_j}$ $\|$ $pid^1_{\mathcal{S}_i}$ $\|$N$_i$ $\|$N$_j$ $A_5 = \{[VC_{\mathcal{S}_i}, w_i, N_i]_{Pu_{\mathcal{D}_j}}\}$		
	$\boxed{\xrightarrow{A_5}}$	Decrypt: A_5 using $Pr_{\mathcal{D}_j}$ Verify: $VC_{\mathcal{S}_i}$ using $Pu_{\mathcal{MP}}$ Ver$_{witness}$(Acc,w$_i$, $\mathcal{S}_i$)=?1 Calculate: K_S= $did_{\mathcal{D}_j}$ $\|$ $pid^1_{\mathcal{S}_i}$ $\|$N$_i$ $\|$N$_j$

D. Mutual Authentication Phase

Each time the seller wants a drone to pick up a package, the protocol requires the seller and the drone to go through the following authentication process.

Step 1: $\mathcal{S}_i$ composes a message A_1 with one of its pseudo-DIDs $pid^1_{\mathcal{S}_i}$ from P, his/her location identifier L_i, and a package pickup request. Then, $\mathcal{S}_i$ sends $A_1 = \{pid^1_{\mathcal{S}_i}, L_i, Req\}$ to the $\mathcal{MP}$. Upon receiving the message from the seller with the request for a drone, the $\mathcal{MP}$ sends a drone $\mathcal{D}_j$ to the docking station located at L^*_i closest to the location L_i of S_i. Then, $\mathcal{MP}$ sends a reply message $A_2 = \{OK, L^*_i, D_j\}$ to $\mathcal{S}_i$ with a response OK, the location identifier L^*_i, and D_j.

Step 2: After receiving A_2, $\mathcal{S}_i$ takes the package to the docking station located at L^*_i. Then, $\mathcal{S}_i$ composes a message A_3 with $pid^1_{\mathcal{S}_i}$ and a request for the VC of the drone D_j. After that, $\mathcal{S}_i$ sends A_3 to the drone D_j.

Step 3: Upon receiving A_3 from $\mathcal{S}_i$ with the request for the VC, $\mathcal{D}_j$ resolves the pseudo-DID of the seller, $pid^1_{\mathcal{S}_i}$, to the corresponding DID document on the blockchain and obtains the public key $Pu_{pid^1_{\mathcal{S}_i}}$ from the DID document. Then, $\mathcal{D}_j$ generates a nonce N_j. After that, $\mathcal{D}_j$ composes a message with its DID $did_{\mathcal{D}_j}$, its VC $VC_{\mathcal{D}_j}$, a request for the seller's VC and non-membership witness, and N_j. Then, D_j encrypts the message with the public key $Pu_{pid^1_{\mathcal{S}_i}}$ of $\mathcal{S}_i$ corresponding to the pseudo-DID, $pid^1_{\mathcal{S}_i}$, to generate the message $A_4 = \{[did_{\mathcal{D}_j}, VC_{\mathcal{D}_j}, Req_{VC}, Req_w, N_j]_{Pu_{pid^1_{\mathcal{S}_i}}}\}$ and send it to $\mathcal{S}_i$.

Step 4: When $\mathcal{S}_i$ receives A_4, he/she decrypts it with the private key $Pr_{pid^1_{\mathcal{S}_i}}$. Then, $\mathcal{S}_i$ verifies the signature on $VC_{\mathcal{D}_j}$ using the $\mathcal{MP}$'s public key $Pu_{\mathcal{MP}}$. After that, $\mathcal{S}_i$ generates a nonce N_i and calculates the session key as

$K_s = did_{\mathcal{D}_j} \parallel pid^1_{\mathcal{S}_i} \parallel N_i \parallel N_j$. Then, $\mathcal{S}_i$ composes $\{VC_{\mathcal{S}_i}, w_i, N_i\}$, encrypts it with the public key of D_j, $Pu_{\mathcal{D}_j}$, to generate the message $A_5 = \{[VC_{\mathcal{S}_i}, w_i, N_i]_{Pu_{\mathcal{D}_j}}\}$ and sends it to D_j.

Step 5: D_j decrypts A_5 with its private key $Pr_{\mathcal{D}_j}$. After that, D_j verifies the signature on $VC_{\mathcal{S}_i}$ with the $\mathcal{MP}$'s public key $Pu_{\mathcal{MP}}$. Then, D_j recovers the accumulator version Acc and verifies that $\mathcal{S}_i$ is not revoked by checking if $Ver_{witness}(Acc, w_i, \mathcal{S}_i)$ is 1. If either verification fails, the execution of the protocol is terminated. Otherwise, D_j calculates the session key as $K_s = did_{\mathcal{D}_j} \parallel pid^1_{\mathcal{S}_i} \parallel N_i \parallel N_j$. Thus, a session key is established between the seller and the drone. The steps involved in the mutual authentication phase are given in Table II.

E. Revocation Phase

When $\mathcal{S}_i$ sends a request to the $\mathcal{MP}$ to unsubscribe from the service, the $\mathcal{MP}$ removes the seller from the accumulator by calling the function $Acc_{upd}()$. Then, the $\mathcal{MP}$ removes the non-membership witness w_i of $\mathcal{S}_i$. The $\mathcal{MP}$ can also initiate the process of unsubscribing a seller from the service if the goods sold by the seller do not meet certain quality criteria.

V. Security Analysis

We provide a security analysis of the proposed protocol in this section.

Privacy of User: In the proposed protocol, the seller's real identity is not revealed while using the service. Further, the seller uses a different pseudo-DID during each authentication event. Hence, the proposed protocol makes it difficult for an adversary to link multiple usage requests of the same person to track his/her activities and usage patterns.

Protection from Eavesdropping and Man-In-The-Middle Attacks: The messages A_4 and A_5 are encrypted with the public keys of the seller and the drone, respectively. Hence, an adversary cannot eavesdrop and modify it as he/she does not have the corresponding private key to decrypt it. Thus, the proposed protocol ensures resilience against eavesdropping and Man-In-The-Middle Attacks.

Protection Against Replay Attacks: In the messages $A_1 = \{pid^1_{\mathcal{S}_i}, L_i, Req\}$ and $A_3 = \{pid^1_{\mathcal{S}_i}, VC_{Req}\}$, a different pseudo-DID from the set P is used in each authentication session. Further, the adversary can't reuse A_4 since the parameter N_j in A_4 changes in each authentication session. Similarly, the parameter N_i in A_5 changes in each authentication session. As a result, the attacker will not be able to replay the messages to launch a replay attack.

Revoked Users Cannot Access Service: To access the service, the users must prove that they are not members of the revocation list by showing a non-membership witness issued by the $\mathcal{MP}$. Hence, revoked users cannot access the service.

Mutual Authentication: The sellers and the drones show their VCs to each other. Only the marketplace platform users and the drones registered with the $\mathcal{MP}$ have valid VCs signed with the private key of the $\mathcal{MP}$. Both the seller and the drone verify the VC presented to them and authenticate each other.

Protection Against User Impersonation Attack: S_i uses a different pseudo-DID $pid^1_{\mathcal{S}_i}$ from P in the messages A_1 and A_3 during each authentication event. The adversary does not have access to P. Hence, the adversary cannot compose A_1 and A_3. The adversary does not have access to the VC, $VC_{\mathcal{S}_i}$, and the revocation witness w_i of the seller as well. As a result, the adversary cannot compose a valid message $A_5 = \{[VC_{\mathcal{S}_i}, w_i, N_i]_{Pu_{\mathcal{D}_j}}\}$ as well to impersonate a legitimate user to get authenticated.

Protection Against Drone Impersonation: An adversary does not have access to the VC, $VC_{\mathcal{D}_j}$, of the drone. As a result, the adversary cannot compose a valid message $A_4 = \{[did_{\mathcal{D}_j}, VC_{\mathcal{D}_j}, Req_{VC}, Req_w, N_j]_{Pu_{pid^1_{\mathcal{S}_i}}}\}$ to impersonate a legitimate drone to get authenticated.

Session Key Agreement: After verifying the VC and revocation status, a session key is established between the seller and the drone at the end of each authentication session as $K_s = did_{\mathcal{D}_j} \parallel pid^1_{\mathcal{S}_i} \parallel N_i \parallel N_j$. Thus, the proposed protocol ensures session key agreement.

VI. Performance Analysis

In this section, we analyze the computation cost of the proposed protocol. Since the seller needs to register with the $\mathcal{MP}$ only once, the registration phase of the proposed protocol is executed only once. Therefore, the performance of the protocol primarily depends on the computation cost during the authentication phase. Hence, we evaluate the computation cost incurred during the authentication phase to analyze the performance of the proposed protocol.

We evaluated the protocol with a Python implementation. The protocol was simulated on a personal computer with Intel (R) Core (TM) i5-11320H @3.20 GHz and 8 GB of RAM memory. We used the ECDSA algorithm for signing the VC with a key length of 256 bits. The concatenation, encryption, decryption, and ECDSA signature verification take 0.08 ms, 2.14 ms, 2.35 ms, and 27.14 ms, respectively. The time taken by various cryptographic operations and the corresponding total execution time taken by the seller's device and the drone during authentication are given in Tables III and IV, respectively.

TABLE III
Execution time of various cryptographic operations

Operation	**Time Taken (ms)**
$Concatenation$	0.08
$Encrypt$	2.14
$Decrypt$	2.35
$Signature\ Verification$	27.14

Next, we provide a comparison of the computation cost of the proposed protocol with other existing authentication methods for drone delivery services. We consider two recent papers for the comparison. The authentication method proposed in [10] which is based on the hand movement of

TABLE IV
EXECUTION TIME DURING AUTHENTICATION

Seller's Device	**Drone**
$3T_C+T_E+T_D+T_{SV} = 31.87$ ms	$3T_C + T_E + T_D + T_{SV} =$ 31.87 ms
Total Time required for authentication: 63.74 ms =1.06 s	
T_C**:** Time required for a concatenation operation;	
T_E/T_D**:** Time required for an encryption/decryption operation;	
T_{SV}**:** Time required for signature verification	

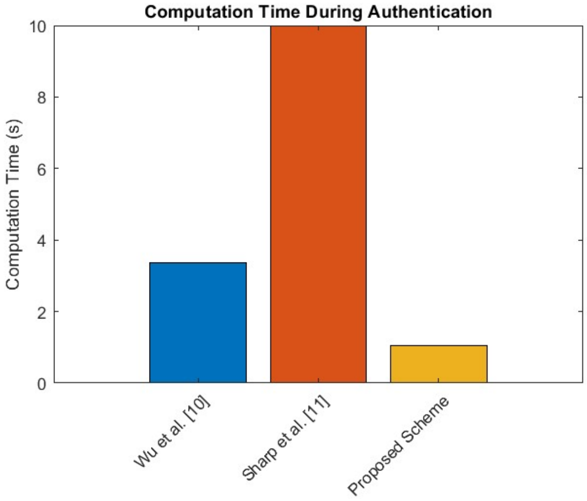

Fig. 2. Comparison of computation cost.

the customer takes 3.36 s on average. The authentication method using face biometrics proposed in [11] takes 10 s as it involves time spent in recording videos. We have plotted the graph for computation cost in Figure 2. From the graph, the computation cost of the proposed protocol is less than that of other existing schemes.

VII. Conclusion

In this paper, we proposed a privacy-preserving mutual authentication protocol for drone delivery services. The proposed protocol enables marketplace platform users to generate and control their IDs without depending on an external party. The protocol provides protection against several attacks. Hence, by using this protocol, the sellers can request drone delivery services in a secure and privacy-preserving manner. The protocol also incorporates efficient membership revocation using an accumulator scheme with non-membership witnesses. We compared the proposed protocol with two other protocols in terms of the computation cost. The comparison shows that the computation cost of the proposed protocol is less than the other protocols.

VIII. Acknowledgement

This work is supported in part by Ministry of Education, Singapore by Tier 2 grant MOE-T2EP20121-0011.

References

[1] C. Chen, S. Leon, and P. Ractham, "Will customers adopt last-mile drone delivery services? an analysis of drone delivery in the emerging market economy," *Cogent Business & Management*, vol. 9, no. 1, p. 2074340, 2022.

[2] S. Ramesh, T. Pathier, and J. Han, "Sounduav: Towards delivery drone authentication via acoustic noise fingerprinting," in *Proceedings of the 5th Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*, 2019, pp. 27–32.

[3] "Decentralized Identifiers (DIDs)," Online, https://www.w3.org/TR/did-core/, [Accessed: Aug 2022].

[4] "Verifiable Credentials Data Model 1.0," Online, https://www.w3.org/TR/vc-data-model/, [Accessed: Aug 2022].

[5] T. Alladi, G. Bansal, V. Chamola, M. Guizani *et al.*, "Secauthuav: A novel authentication scheme for uav-ground station and uav-uav communication," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15 068–15 077, 2020.

[6] M. Tanveer, A. H. Zahid, M. Ahmad, A. Baz, and H. Alhakami, "Lake-iod: Lightweight authenticated key exchange protocol for the internet of drone environment," *IEEE Access*, vol. 8, pp. 155 645–155 659, 2020.

[7] S. Hussain, S. A. Chaudhry, O. A. Alomari, M. H. Alsharif, M. K. Khan, and N. Kumar, "Amassing the security: An ecc-based authentication scheme for internet of drones," *IEEE Systems Journal*, vol. 15, no. 3, pp. 4431–4438, 2021.

[8] M. A. Cheema, R. I. Ansari, N. Ashraf, S. A. Hassan, H. K. Qureshi, A. K. Bashir, and C. Politis, "Blockchain-based secure delivery of medical supplies using drones," *Computer Networks*, vol. 204, p. 108706, 2022.

[9] M. Singh, G. S. Aujla, R. S. Bali, S. Vashisht, A. Singh, and A. Jindal, "Blockchain-enabled secure communication for drone delivery: a case study in covid-like scenarios," in *Proceedings of the 2nd ACM MobiCom Workshop on Drone Assisted Wireless Communications for 5G and beyond*, 2020, pp. 25–30.

[10] C. Wu, X. Li, L. Luo, and Q. Zeng, "G2auth: secure mutual authentication for drone delivery without special user-side hardware," in *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*, 2022, pp. 84–98.

[11] J. Sharp, C. Wu, and Q. Zeng, "Authentication for drone delivery through a novel way of using face biometrics," in *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, 2022, pp. 609–622.

[12] Y. Kortesniemi, D. Lagutin, T. Elo, and N. Fotiou, "Improving the privacy of iot with decentralised identifiers (dids)," *Journal of Computer Networks and Communications*, vol. 2019, p. 8706760 (10 pp.), 2019.

[13] M. Eisenstadt, M. Ramachandran, N. Chowdhury, A. Third, and J. Domingue, "Covid-19 antibody test/vaccination certification: There's an app for that," *IEEE Open Journal of Engineering in Medicine and Biology*, vol. 1, pp. 148–155, July 2020.

[14] J. Camenisch and A. Lysyanskaya, "Dynamic accumulators and application to efficient revocation of anonymous credentials," in *Annual international cryptology conference*. Springer, 2002, pp. 61–76.

[15] J. Li, N. Li, and R. Xue, "Universal accumulators with efficient nonmembership proofs," in *International Conference on Applied Cryptography and Network Security*. Springer, 2007, pp. 253–269.