An Improvised Certificate-Based Proxy Signature Using Hyperelliptic Curve Cryptography for Secure UAV Communications

Muhammad Asghar Khan, *Senior Member, IEEE*, Insaf Ullah, Neeraj Kumar, *Senior Member, IEEE*, Adnan Akhunzada, *Senior Member, IEEE*, Mohammad Hossein Anisi, *Senior Member, IEEE*, Abdulmajeed Alqhatani, Fatemeh Afghah, *Senior Member, IEEE*, Gordana Barb, *Associate Member, IEEE*, and Abi Waqas, *Senior Member, IEEE*

***Abstract*— Unmanned aerial vehicles (UAVs) have enabled numerous inventive solutions to multiple problems, considerably facilitating our daily lives; however, UAVs frequently rely on an open wireless channel for communication, making them susceptible to cyber-physical threats. Also, UAVs cannot execute complicated cryptographic algorithms due to their limited onboard computing capabilities. Balancing high-security levels and minimum computation costs is imperative when developing a security solution for UAVs. Consequently, several proxy signature schemes have been proposed in the literature to fulfill these requirements. Nevertheless, many of these solutions face the issue of high computation costs, and some exhibit security vulnerabilities that could not be more feasible options for UAV communication. Considering these constraints in mind, in this article, we introduce an improvised certificate-based proxy signature scheme (ICPS), which leverages the concept of hyperelliptic curve cryptography (HECC) to meet the security and efficiency requirements of UAV networks. The proposed ICPS scheme offers a range of notable features, including its ability to address key escrow and secret key distribution issues. The proposed ICPS scheme's security hardness has been evaluated using the widely known security tool, the random oracle model (ROM), proving its resilience against known and unknown cybersecurity threats. Finally, this study conducts a performance comparison of the proposed scheme against existing schemes, emphasizing its outstanding cost-efficiency. Notably, the computation cost is measured at 5.3536 ms and the communication cost at 1120 bits, substantially lower than relevant existing schemes.**

***Index Terms*— Unmanned aerial vehicles, proxy signature, hyperelliptic curve, security, computation cost, random oracle model.**

Received 19 March 2024; revised 17 October 2024 and 29 November 2024; accepted 29 December 2024. Date of publication 15 January 2025; date of current version 31 March 2025. The Associate Editor for this article was X. Lei. *(Corresponding author: Muhammad Asghar Khan.)*

Muhammad Asghar Khan is with the Department of Electrical Engineering, Prince Mohammad Bin Fahd University, Al Khobar 31952, Saudi Arabia (e-mail: m.asghar@ieee.org; mkhan4@pmu.edu.sa).

Insaf Ullah is with the Institute for Analytics and Data Science, University of Essex, CO4 3SQ Colchester, U.K. (e-mail: insafktk@gmail.com).

Neeraj Kumar is with CSED, Thapar Institute of Engineering and Technology, Patiala 147004, India, and also with the Department of Networks and Communications, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, Dammam 31441, Saudi Arabia (e-mail: neeraj.kumar@thapar.edu).

Adnan Akhunzada is with the College of Computing and Information Technology, University of Doha for Science and Technology, Doha, Qatar (e-mail: adnan.adnan@udst.edu.qa).

Mohammad Hossein Anisi is with the School of Computer Science and Electronic Engineering, University of Essex, CO4 3SQ Colchester, U.K. (e-mail: m.anisi@essex.ac.uk).

Abdulmajeed Alqhatani is with the Department of Information Systems, Najran University, Najran 66454, Saudi Arabia (e-mail: aaalqhatni@nu.edu.sa).

Fatemeh Afghah is with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634 USA (e-mail: fafghah@clemson.edu).

Gordana Barb is with the Department of Communications, Politehnica University Timisoara, 300006 Timișoara, Romania (e-mail: gordana.barb@upt.ro).

Abi Waqas is with the Telecommunication Department, Mehran University of Engineering and Technology (MUET), Jamshoro 76062, Pakistan (e-mail: abi.waqas@faculty.muet.edu.pk).

Digital Object Identifier 10.1109/TITS.2024.3524575

I. Introduction

THE recent technological progress in embedded systems, sensors and wireless technologies has equipped unmanned aerial vehicles (UAVs) with a remarkable ability to carry out various tasks autonomously and collaboratively, eliminating human intervention [1]. Initially considered military tools, UAVs are used extensively in various civilian applications and primarily for exciting commercial purposes [2]. These UAV applications in civilian usage have a much broader spectrum, which includes but is not limited to agriculture, aerial photography, search and rescue, environmental monitoring, infrastructure inspection, surveying, delivery services, disaster response, scientific research, recreation, infrastructure development, telecommunications, wildlife conservation, mining, and resource exploration [3], [4], [5]. This is mainly because of the UAV's incredible flight capabilities, allowing it to fly at low altitudes and high elevation angles [6]. UAVs utilize navigation systems to accurately determine their position, communication systems to send and receive data, flight controllers to ensure stability and precise control of their movement, and cutting-edge sensors for a wide range of specific applications. In addition, the increasing use of UAVs has led to new approaches for various tasks and applications. Still, attackers are increasingly targeting both the UAVs and the intricate networks that enable their fast-paced operations and control [7].

UAVs and their networks are vulnerable to security breaches due to unique challenges, such as limited computing capacity and battery life, and they communicate via a wireless channel [8]. The main UAV components (hardware, sensor,

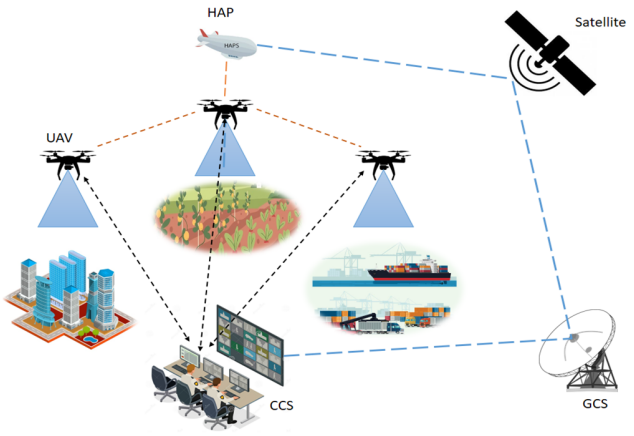

Fig. 1. A typical communication architecture for UAVs.

and software) and UAV communications could be exploited by attackers to launch cyber-attacks and leak critical data. The primary security solutions for cybersecurity challenges in UAV communications include strong data encryption to protect transmitted information, protection from signal jamming and interference, physical security to prevent unauthorized access and tampering, defence against GPS spoofing and hacking threats, protection from malware and cyber-attacks [9]. In addition, patching vulnerabilities requires timely deployment of firmware and software updates, and preventing vulnerabilities from entering the UAV ecosystem necessitates safe supply chain management. For safe UAV operation, it is essential to establish secure connections between UAVs or between a UAV and the infrastructure. The UAVs must ensure that only authorized users can access their resources and that all their internal components are verified. As the fundamental method for ensuring communication security, the authentication protocol between the UAV and the control station can verify the legitimacy of the communication systems.

Authentication is achieved through a digital signature mechanism to ensure data integrity, authenticity, and non-repudiation. Hence, this cryptographic method enhances trust, reduces the risk of unauthorized access or data manipulation, and promotes accountability for UAV-based aggregated data. However, UAVs usually perform remote tasks beyond the command control centre (CCC) range, allowing networks maximum flexibility and independence in their operations [10]. Thus, in such scenarios, the CCC generally designates a GCS on the role of an agent in the UAV's proximity, and the GCS confidently communicates commands directly to the UAVs. A typical UAV communication architecture is depicted in Fig. 1, which shows the main entities and their connectivity. A proxy signature [11] can be the most feasible option in these use cases to achieve confidentiality and authenticity of the transmitted commands. The proxy signature scheme can be constructed in various cryptographic settings, which include a certificateless cryptosystem (CLC), the identity-based cryptosystem (IBC), or the public key infrastructure (PKI) [12], [13], [14]. In certificateless cryptography [15], the key generation centre (KGC) is typically designated for generating the public and private key pairs, which may raise concerns about key escrow. This problem is because the KGC could retain copies of the private keys, compromising the system's security and privacy. Furthermore, key escrow may lead to a single point of failure problem, which compromises the scheme's viability. On the other hand, IBC [16] also has several drawbacks, including reliance on a trusted authority (TA), a limited revocation mechanism, key escrow concerns, and scalability issues. In addition, certificate-based cryptography [17] uses digital certificates to achieve authentication and non-repudiation and is a suitable option for constructing a proxy signature scheme.

From a security standpoint, a certificate-based proxy signature scheme is the most suitable method for UAV communication. It addresses key escrow issues by eliminating the need for a trusted third party to manage private keys. Additionally, the certificate-based approach ensures authentication and integrity of communications, while the proxy signature mechanism enables secure delegation of signing authority, providing both security in UAV networks. However, from an efficiency standpoint, UAVs' insufficient onboard computing capability can hamper such initiatives. Due to the inadequate hardware capacity for computation, UAVs cannot perform complex cryptographic operations, and developing efficient security mechanisms that incur less computation and communication costs is much needed. In the existing literature, the security and efficiency of the certificate-based proxy signature scheme relied on computationally challenging problems such as rivest-shamir-adleman (RSA) cryptography, bilinear pairing (BP), and elliptic curve cryptography (ECC) [18]. The RSA-based security methods are based on solving significant factorization problems with keys, parameters, certificates, and identities as long as 1024 bits. This cryptographic approach is not best suited for the resource-constrained nature of UAV networks. A typical UAV may have ample onboard processing resources to handle the RSA-based cryptographic operations. Similarly, BP is slower than RSA due to the extensive pairing and map-to-point function computations involved.

For UAV communication, ECC-based cryptographic schemes are suggested to address the shortcomings of RSA and BP. In ECC methods, critical components like parameters, public keys, private keys, identities, and certificates are meant to be as small as 160 bits, requiring much lower computation than RSA and BP. However, the new version of ECC, known as hyperelliptic curve cryptography (HECC), may further reduce this computation, which uses just an 80-bit key size to provide the same level of security as RSA, BP and ECC. This paper proposes a cryptographic solution that combines HECC and a certificate-based proxy signature. The key contributions of the proposed ICPS are as follows:

1) We develop an improvised certificate-based proxy signature (ICPS) scheme, effectively resolving the key escrow problem and eliminating the necessity for secret key distribution.
2) The proposed ICPS scheme utilizes the HECC concept, which offers a noteworthy compact advantage. It requires a much smaller key size of 80 bits for the same level of security as RSA, ECC, BP, and others. This represents a substantial reduction in key length, which in

turn helps reduce the computation and communication burdens on UAVs.

3) We conducted a comprehensive security analysis using the random oracle model (ROM) to confirm the proposed ICPS scheme's robustness against widely recognized security attacks. We also performed an efficiency analysis in terms of computation and communication costs with the relevant existing schemes. The results demonstrate the proposed scheme's effectiveness in both security and efficiency.

The organization of the paper is as follows: Section II presents a review of related work, Section III outlines the preliminaries, Section IV describes the system models, followed by the proposed scheme in Section V. Section VI discusses the security analysis, Section VII provides the performance evaluation, and Section VIII concludes with final remarks.

II. Related Work

Due to the rapid changes in speed and topology of the UAV network, it is essential to validate the commands and their authenticity in the shortest time possible. UAV's job is to verify signatures efficiently, particularly for location-based services. For example, when the user or ground station (GS) initiates a command and signature, the UAV must verify the signature. In addition, a UAV conducts remote tasks occasionally beyond the command control centre (CCC) range, preventing the CCC from communicating directly with the UAVs. In this scenario, the CCC identifies a GCS near the UAV as an agent, and the GCS transmits commands directly to the UAV. Given these constraints, there are more suitable options than conventional digital signature schemes for the security of a UAV-based network [19]. As a result, a proxy signature scheme can be adopted to authenticate transmitted commands and ensure that UAVs execute commands on time.

In 1996, Mambo et al. [20] introduced the idea of a proxy signature in which the primary signatory permits the proxy signatory the authority to sign on their behalf. Similarly, Verma et al. [21] proposed a short proxy signature approach using certificate-based cryptography for UAV networks. The proposed scheme [21] benefits signature length to address security and privacy challenges in the UAV network and mitigate cybersecurity threats. On the other hand, Verma et al.'s [21] scheme was based on BP operations, which involved heavy pairing operations and is unsuitable for UAVs due to their limited onboard computing resources. Performing computationally intensive BP operations is generally impractical for UAVs. To meet UAVs' security and efficiency requirements, He et al. [22] proposed a certificateless designated verifier proxy signature (CLDVPS) scheme. In this scheme [22], only the designated verifier can validate the signature. However, the proposed CLDVPS scheme could not withstand an impersonation attack. Xu et al. [23] proposed a solution to the impersonation attacks that hampered He et al. [22]'s scheme. Xu et al. [23] proposed a new CLDVPS protocol to address UAV networks' data security and privacy concerns. However, this scheme was also founded on computationally intensive bilinear operation, which is impractical for UAVs due to their limited resources.

Qiao et al. [24] demonstrated security concerns and design flaws in the previously published CBPS proposals. Three new CBPS schemes with enhanced security were introduced to address the deficiencies mentioned above. In the first two proposals, the validity of delegation is verifiable by the designated verifier (proxy signer). In contrast, the delegation of signature authority in the third proposal is publicly verifiable (by any signer). In addition, formal security proofs are provided using the forking Lemma in the random oracle under the assumption that the discrete logarithm problem is complex.

Nonetheless, most of the schemes mentioned above exhibit security vulnerabilities in the form of security flaws and operational inefficiencies in high computation and communication costs. For instance, Verma et al.'s [21] and Xu et al.'s [23] schemes rely on BP operations, rendering it less practical for UAVs due to the inherent limitations of these platforms regarding computational resources. Moreover, He et al. [22] introduced a certificateless designated verifier proxy signature (CLDVPS) scheme, wherein only the designated verifier can authenticate the signature and be vulnerable to impersonation attacks, raising further concerns about its security.

This article proposes an ICPS scheme for UAV communication to resolve existing schemes [21], [22], [23], [24] security and efficiency challenges, as shown in Tab. I. The security robustness of the proposed ICPS scheme is evaluated using the ROM, a formal security validation tool. The proposed scheme is constructed with HECC, an advanced form of elliptic curve cryptography (ECC), offering the same security level as ECC, RSA and BP. By comparing the performance of the proposed scheme to that of other existing schemes [21], [22], [23], particularly in computation and communication costs, this study's results demonstrate the proposed scheme's effectiveness. Additionally, the proposed ICPS scheme successfully addresses the key escrow issue and eliminates the need for secret key distribution.

III. Preliminaries

This section discusses fundamental explanations for the hyperelliptic curve, followed by the discrete logarithm problem for the hyperelliptic curve, the syntax of the proposed ICPS scheme for UAV networks, the threat model, and the network model. The topics mentioned above are explained in the subsections below.

A. Hyperelliptic Curve Cryptography

A class of algebraic curves known as the EC_H is sometimes called an elliptic curve (EC) in generalized form. EC_H points, however, cannot be acquired in a group. The additive Abelian group can be calculated using the EC_H or derived from a divisor. EC_H offers an advantage over RSA, EC, and BP since it requires fewer parameters to achieve the same level of security. The genus (gns) 2 EC_H can be defined as $w^2 + h(c)\,w = f(c)$, where $h(c)$ denotes the polynomial with degree $\leq gns$ defined over a finite field (EC_{f_n}) and $f(c)$ denotes the monic polynomial with degree $2gns + 1$, define over-finite field $\left(EC_{f_n}\right)$.

B. Hyper Elliptic Curve Discrete Logarithm Problem

Suppose $B = x.d_J$ is given, where $x \in [1, n-1]$ and $d_J \in J\left(EC_{f_n}\right)$; therefore, finding the value (x) from B is called a hyperelliptic curve discrete logarithm problem (HECDLP). Note that d_J represents the devisor, which belongs to the Jacobian group $[J\left(EC_{f_n}\right)]$ that is defined over the finite field (EC_{f_n}) of EC_H.

C. Syntax of the Proposed ICPS Scheme

The syntax of the proposed certificate-based proxy signature scheme for UAV networks contains the following sub-steps:

- *Setup:* CA executes this step, in which it sets his private key (SEK_{CA}), master public key (MBK_{CA}), and public parameter param (P_{CA}). Then, CA published P_{CA} for the UAV network.
- *Key Generation:* A user with identity (I_{ui}) selects(PK_{ui}) randomly as his private key and computes his public key (PBK_{ui}).
- *Certificate Generation:* When a user with an identity (I_{ui}) sends a request for a certificate to the CA, it generates the certificate (CT_{ui}) and sends it to that particular user.
- *Delegation Generation:* To generate the signature on the warrant message (w_m), CCC first takes his private and public keys (PK_{ccc}, PBK_{ccc}), identity (I_{ccc}), devisor (d_J), the public key (PBK_{gcs}) of GCS, and warrant message (w_m) as input. Then, it generates the delegation (DEL_{ccc}) and sends it to the GCS.
- *Delegation Verification:* When (DEL_{ccc}) is received by GCS, this step is done by checking the validity of DEL_{ccc}. For this purpose, GCStake his private and public keys (PK_{gcs}, PBK_{gcs}), the identity (I_{ccc}) and public key (PBK_{ccc}) of CCC, and the devisor (d_J), which is an input.
- *Proxy Signature Generation:* After successfully validating DEL_{ccc}, GCS performs this step, in which it first takes his private and public keys (PK_{gcs}, PBK_{gcs}), his identity (I_{gcs}), his certificate (CT_{gcs}), devisor (d_J), the public key (PBK_{ccc}) of CCC, and message (m) as input. Then, it generates the proxy signature (PS_{gcs}) and sends it to the UAV.
- *Proxy Signature Verification:* When (PS_{gcs}) is received by the UAV, it performs this step by checking the validity of PS_{gcs}. For this purpose, UAV takes the identity and public key (I_{gcs}, PBK_{gcs}) of GCS, the identity and the public key (I_{ccc}, PBK_{ccc}) of CCC, the master public key (MBK_{CA}) of CA, and the warrant and message (w_m, m), which are inputs.

IV. System Model

The practicality of the proposed scheme has been demonstrated by utilizing two models: the network model and the threat model. The details of these models are provided as follows:

A. Network Model

The proposed network model for UAV networks, illustrated in Fig. 2, includes entities like the command control centre (CCC), a ground control station (GCS), a UAV, and a certificate authority (CA). The task of each entity is defined as follows:

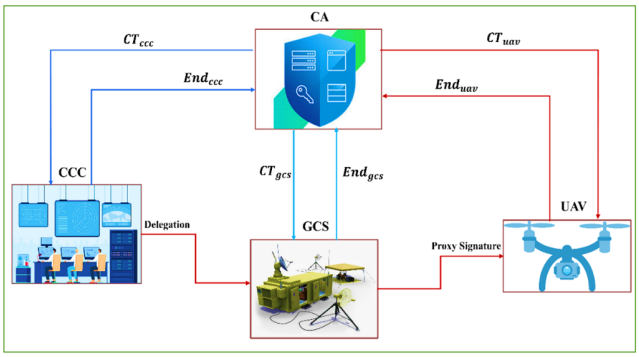

Fig. 2. Network model of the proposed ICPS scheme.

- *Command control centre (CCC):* This entity first sets its private and public key pairs (PK_{ccc}, PBK_{ccc}), then sends its public key along with an encrypted identity (PBK_{ccc}, End_{ccc}) to CA. Upon receiving (PBK_{ccc}, End_{ccc}), CA first recovers the secret key (K_{ccc}) and decrypts the real identity $(I_{ccc})$from the encrypted identity End_{ccc} by using the decryption function $(\oplus)$ through the secret key. After that, the certificate is generated by using that particular identity and public key, and then it delivers the certificate (CT_{ccc}) to CCC. By using his private key (PK_{ccc}) and (CT_{ccc}), it generates a delegation signature (DEL_{ccc}) on a warrant (w_m) and sends it to the GCS.
- *Ground control station (GCS):* Upon receiving the delegation signature (DEL_{ccc}), GCS first sets his private and public key pairs (PK_{gcs}, PBK_{gcs}), then sends his public key along with an encrypted identity (PBK_{ccc}, End_{ccc}) to CA. Upon receiving (PBK_{ccc}, End_{ccc}), CA first recovers the real identity $\left(I_{gcs}\right)$from the encrypted identity (End_{ccc}) by using the decryption function $(\oplus)$ through the secret key. Then, it generates the certificate by using that particular identity and public key, and then it delivers the certificate (CT_{gcs}) to CCC. After that, it performs the verification procedure to check the validity of the DEL_{ccc}. If it is valid, it generates the proxy signature (PS_{gcs}) on the message m and sends it to the UAV by using an insecure network.
- *Unmanned aerial vehicle (UAV):* Upon receiving the proxy signature (PS_{gcs}), UAV uses the verification procedure to check the validity of PS_{gcs}. If it is valid, it accepts the proxy signature (PS_{gcs}); otherwise, it generates an error message.
- *Certificate authority (CA):* The primary work of CA is to generate the master private key (SEK_{CA}), master public key (MBK_{CA}), and public parameter param (P_{CA}). When a user with an identity (I_{ui})wants a certificate from CA, it sends his public key along with an encrypted identity (PBK_{ui}, End_{ui}) to CA. Upon receiving (PBK_{ui}, End_{ui}), CA first recovers the real identity$(I_{ui})$from the encrypted identity (End_{ui}) by using

TABLE I
SUMMARY OF EXISTING WORK

Work	Description	Strengths	Weaknesses
Verma et al. [21]	Proposed a short proxy signature scheme using certificate-based cryptography.	The proposed scheme benefits from signature length in addressing security and privacy challenges in the UAV network and mitigating cybersecurity threats.	Performing BP is a mathematically complex task for UAVs when performing certificate-based signature ring operations.
He et al. [22]	Proposed a certificateless designated verifier proxy signature (CLDVPS) scheme, where only the designated verifier can validate the signature.	In this scheme, only the designated verifier is able to validate the signature.	Vulnerable to impersonation attacks.
Xu et al. [23]	Proposed a new CLDVPS protocol to address UAV networks' data security and privacy concerns.	The proposed scheme addressed the vulnerability of impersonation attacks of He et al.'s scheme [22].	BP operations incur high computation and communication costs, which are impractical for UAVs due to their limited computing capability.
Qiao et al. [24]	Proposed a secure and efficient certificate-based proxy signature scheme.	The proposed scheme demonstrated security concerns and design flaws in the previously published CBPS proposals. To address the deficiencies, three new CBPS schemes with enhanced security were introduced.	Incurs high computation and communication costs.
Our Scheme	Proposed an improvised certificate-based proxy signature scheme (ICPS), which leverages the concept of hyperelliptic curve cryptography (HECC) to meet the security and efficiency requirements of UAV networks.	The proposed ICPS scheme addresses key escrow and secret key distribution issues and uses HECC's smaller key size for higher security and efficiency.	Fails to address vulnerabilities posed by machine learning-based and quantum attacks.

the decryption function ($\oplus$) through the secret key. After that, CA generates a certificate (CT_{ui}) and sends it to the user with their identity (I_{ui}).

B. Threat Model

In this subsection, we will discuss two types of forgers: Type 1 (F_{CA1}) and Type 2 (F_{CA2}). The task of these forgers is to destroy the forgeability security property of our proposed certificate-based proxy signature scheme for UAV networks. The Type 1 (F_{CA1}) forger acts like an outsider opponent that can replace the user's public key but has no access to the master private key of CA. In contrast, the Type 2 (F_{CA2}) forger acts like an internal opponent (malicious CA) who has access to the master private key of the CA but cannot replace the user public key. The notions used in the proposed scheme are provided in Tab.I.

The opponent can request the following queries, and the challenger will respond accordingly. So, before responding to the opponent, the challenger runs the setup algorithm and gets the param and master secret keys.

- *Public Key Generation Query:* The opponent sends this query with an identity (I_{ui}), and the challenger returns the public and private keys for that particular identity.
- *Secret Key Generation Query:* The opponent sends this query with an identity (I_{ui}), and the challenger returns the private key for that particular identity.
- *Certificate Generation Query:* The opponent sends this query with an identity (I_{ui}), and the challenger returns the certificate for that particular identity.
- *Public Key Replaced Query:* The opponent sends this query with an identity (I_{ui}), and the challenger replaces the public key of that particular identity.
- *Corruption Query:* The opponent sends this query with an identity (I_{ui}), and the challenger returns the private key for that particular identity.
- *Delegation Generation Query:* The opponent sends this query with an identity (I_{ui}) and warrant (w_m), and then the challenger returns the delegation signature tuple for that particular identity.
- *Proxy Signature Generation Query:* The opponent sends this query with an identity (I_{ui}) and message (m), and then the challenger returns the proxy signature tuple for that particular identity.

The corresponding ICPS scheme is existential unforgeability against adaptive chosen message or warrant attacks ($\mathrm{EUF}-\mathrm{ICPS}-\mathrm{CMWA}$) if the probabilities $\Pr[Extl_{F_{CA1}}^{\mathrm{EUF}-\mathrm{ICPS}-\mathrm{CMWA}-\mathrm{I}}(x) = 1]$ and

$\Pr[Extl_{F_{CA2}}^{\text{EUF-ICPS-CMWA-II}}(x) = 1]$ are negligible for any two forgers F_{CA1} and F_{CA2}. The following are the definitions of (EUF − CMWA), EUF − ICPS − CMWA − I and EUF − ICPS − CMWA − II.

- Existential Unforgeability against Adaptive Chosen Message or Warrant Attacks (EUF − CMWA): The proxy signature scheme will be unforgeable for any new messages against F_{CA1} and F_{CA2} forgers.
- EUF − ICPS − CMWA − I: The proposed ICPS scheme is existentially unforgeable against adaptive chosen message or warrant attacks launched by F_{CA1} forger.
- EUF − ICPS − CMWA − II: The proposed ICPS scheme is existential unforgeability against adaptive chosen message or warrant attacks launched by F_{CA2} forger.

Note that in the security analysis phase, we will represent existential unforgeability against adaptive chosen message or warrant attacks (EUF − CMWA). For the resistance against F_{CA1},the ICPS scheme could be existential unforgeability against adaptive chosen message or warrant attacks (EUF−ICPS−CMWA−I), and for F_{CA2}, it could be existential unforgeability against adaptive chosen message or warrant attacks (EUF − ICPS − CMWA − II).

V. Proposed Scheme

The proposed ICPS for UAV networks contains the following sub-algorithms. The symbols used in the proposed ICPS scheme are provided in Tab.II.

1) *Setup:* Suppose $(h_{ca1}, h_{ca2}, h_{ca3})$ are three cryptographic secure collision-resistant functions, n is a large prime number, EC_H is a hyperelliptic curve defined over a finite field (EC_{f_n}), let $J(EC_{f_n})$ represent the Jacobian of EC_H. CA selects a reduced devisor (d_J), where $d_J \in J\left(EC_{f_n}\right)$, select $SEK_{CA} \in [1, n-1]$, compute $MBK_{CA} = SEK_{CA}.d_J$, generate the standard parameter param $P_{CA} = (h_{ca1}, h_{ca2}, h_{ca3}, n, EC_H, PBK_{CA}, EC_{f_n}, d_J, J\left(EC_{f_n}\right))$. CA sets SEK_{CA} as his private key and PBK_{CA} as his master public key. CA published P_{CA} to the UAV network.
2) *Key Generation:* A user with identity (I_{ui}), select $PK_{ui} \in [1, n-1]$ and compute $PBK_{ui} = PK_{ui}.d_J$. Then, a user with I_{ui} sets PK_{ui} as his private key and PBK_{ui} as his public key. When there is a desire for a certificate, the user with I_{ui} can select $G_{ui} \in [1, n-1]$, computes $U_{ui} = G_{ui}.d_J$, $K_{ui} = G_{ui}.PBK_{CA}$, $End_{ui} = I_{ui} \oplus K_{ui}$, and send $(End_{ui}, PBK_{ui}, U_{ui})$ to CA.
3) *Certificate Generation:* When $(End_{ui}, PBK_{ui}, U_{ui})$ received to CA, then it computes$K_{ui} = U_{ui}.SEK_{CA}$ and $I_{ui} = End_{ui} \oplus K_{ui}$. After that CA select $X_{ui} \in [1, n-1]$, compute $Y_{ui} = X_{ui}.d_J$, $CT_{ui} = X_{ui} + SEK_{CA}.h_{ca1}(I_{ui}, PBK_{ui})$, and send CT_{ui} to the user with identity (I_{ui}) by using an open network and store (CT_{ui}, I_{ui}) in his database to avoid an impersonation attack. Upon receivingCT_{ui}, the user with I_{ui} can check its validity if the equality of the following equation is satisfied: $CT_{ui}.d_J = Y_{ui} + MBK_{CA}.h_{ca1}(I_{ui}, PBK_{ui})$.

TABLE II
Notation Table

Notation	Descriptions
$h_{ca1}, h_{ca2}, h_{ca3}$	These are cryptographic functions with the property of collision-resistant parts.
EC_H	It represents a hyperelliptic curve defined over a finite field.(EC_{f_n}).
(EC_{f_n})	It represents a finite field over a large prime number n, *where* $n \geq 2^{80}$.
$J(EC_{f_n})$	It represents the Jacobian of EC_H.
d_J	It represents a reduced devisor, where $d_J \in J(EC_{f_n})$.
SEK_{CA}	It represents a secret key of CA.
MBK_{CA}	It is the master public key of CA.
P_{CA}	It represents the standard parameter param.
PK_{ui}	The private key is for a user with an identity (I_{ui}).
PBK_{ui}	The public key for a user with identity (I_{ui}).
I_{ui}	It indicates the participant user's identity, where $ui \in [CCC, GCS, UAV]$.
$\oplus$	This symbol encrypts and decrypts the user's original identity (I_{ui}).
EC_H	It represents a hyperelliptic Curve defined over a finite field(EC_{f_n}).
$J(EC_{f_n})$	It indicates Jacobian of Hyper elliptic Curve (EC_H).
K_{ui}	It represents the secret shared key between a user with identity (I_{ui}) and CA.
CT_{ui}	The certificate for a user with identity (I_{ui}).
d_J	It indicates the divisor of a hyperelliptic Curve (EC_H).
CT_{ccc}	It indicates the certificate of CCC.
CT_{gcs}	It indicates the certificate of GCS.
PK_{ccc}	It indicates the private key of CCC.
Id_{ccc}	It indicates the identity of CCC.
PBK_{ccc}	It indicates the public key of CCC.
PK_{gcs}	It indicates the private key of GCS.
PBK_{gcs}	It indicates the public key of GCS.
Id_{GCS}	It indicates the identity of GCS.
w_m	It represents the warrant message.
m	It represents the plaintext.

4) *Delegation Generation:* To generate the signature on the warrant message (w_m), CCC performs the following steps.
   - It selects$F_{ccc} \in [1, n-1]$ and compute $W_{ccc} = Y_{ccc} + F_{ccc}.d_J$,
   - Compute $SW_{ccc} = CT_{ccc} + F_{ccc} + PK_{ccc}.h_{ca2}(I_{ccc}, PBK_{ccc}, w_m, PK_{ccc}.PBK_{gcs})$
   - Finally, send $DEL_{ccc} = (SW_{ccc}, w_m, W_{ccc})$ to GCS using an open network.
5) *Delegation Verifications:* When (DEL_{ccc}) received to GCS, then by checking the equality of the equation $SW_{ccc}.d_J = W_{ccc} + PBK_{gcs}.h_{ca1}(I_{ccc}, PBK_{ccc}) + PBK_{ccc}.h_{ca2}(I_{ccc}, PBK_{ccc}, w_m, PBK_{ccc}.PK_{gcs})$, if it is satisfied then accept DEL_{ccc}.

TABLE III
COMPARISON OF SECURITY REQUIREMENTS

Schemes	Unforgeability	Resist Against Replay	Resist Against Impersonation	Resist Against Man in the Middle
Verma et al. [21]	Yes	*No*	*No*	*No*
He et al. [22]	Yes	*No*	*No*	*No*
Xu et al. [23]	Yes	*No*	*No*	*No*
Proposed Scheme	Yes	Yes	Yes	Yes

$PBK_{ccc}.PK_{gcs} = PK_{ccc}.PBK_{gcs} = PK_{ccc}.PK_{gcs}.d_J = PK_{ccc}.d_J.PK_{gcs} = PBK_{ccc}.PK_{gcs}$.

6) *Proxy Signature Generation:* After successful validation of DEL_{ccc}, GCS performs the following steps for the generation of the proxy signature:
   - It selects $F_{gcs} \in [1, n-1]$ and computes $R_{gcs} = Y_{gcs} + F_{gcs}.d_J$
   - Compute $SP_{gcs} = W_{ccc} + CT_{gcs} + F_{gcs} + PK_{gcs}.h_{ca3}(I_{gcs}, PBK_{gcs}, w_m, m, R_{gcs})$
   - Set $W_{gcs} = W_{ccc}$ and compute $K_{gcs} = PK_{gcs}.PBK_{ccc}$
   - Finally, send $PS_{gcs} = (SP_{gcs}, K_{gcs}, W_{gcs}, R_{gcs}, T_{gcs})$ to UAV using the open network.
7) *Proxy Signature Verifications:* When (PS_{gcs}, T_{gcs}) is received by UAV, check the validity of T_{gcs} and verify PS_{gcs} by using the following steps.
   - Compute $h_{ccc1} = h_{ca1}(I_{ccc}, PBK_{ccc})$ and $h_{gcs1} = h_{ca1}(I_{gcs}, PBK_{gcs})$,
   - Compute $h_{gc2} = h_{ca2}\left(I_{ccc}, PBK_{ccc}, w_m, K_{gcs}\right)$ and $h_{gc3} = h_{ca3}\left(I_{gcs}, PBK_{gcs}, w_m, m, R_{gcs}\right)$
   - Check the equality of the equation, $SP_{gcs}.d_J = R_{gcs} + W_{ccc} + MBK_{CA}\left(h_{ccc1} + h_{gcs1}\right) + h_{gc2}.PBK_{ccc} + h_{gc3}.PBK_{gcs}$, if it is satisfied, then accept PS_{gcs}.

A. Correctness

When (PS_{gcs}) is received by a UAV, then to check the validity of PS_{gcs}, it checks the equality of $SP_{gcs}.d_J = R_{gcs} + W_{ccc} + MBK_{CA}\left(h_{ccc1} + h_{gcs1}\right) + h_{gc2}.PBK_{ccc} + h_{gc3}.PBK_{gcs}$; if it is satisfied, then accept PS_{gcs}. The following are the computations for $SP_{gcs}.d_J = R_{gcs} + SW_{ccc} + MBK_{CA}\left(h_{ccc1} + h_{gcs1}\right) + h_{gc2}.PBK_{ccc} + h_{gc3}.PBK_{gcs}$.

$$\begin{aligned} SP_{gcs}.d_J &= \left(SW_{ccc} + CT_{gcs} + F_{gcs} + PK_{gcs}.h_{ca3} \right. \\ &\quad \left. \times \left(I_{gcs}, PBK_{gcs}, w_m, m, R_{gcs}\right)\right).d_J \\ &= \left(SW_{ccc} + CT_{gcs} + F_{gcs} + PK_{gcs}.h_{gc3}\right).d_J \\ &= \left(CT_{ccc} + F_{ccc} + PK_{ccc}.h_{gc2} \right. \\ &\quad \left. + CT_{gcs} + F_{gcs} + PK_{gcs}.h_{gc3}\right).d_J \\ &= W_{ccc} + h_{ccc1}.MBK_{CA} + PBK_{ccc}.h_{gc2} + Y_{gcs} \\ &\quad + h_{gcs1}.MBK_{CA} + + F_{gcs}.d_J + PBK_{gcs}.h_{gc3} \\ &= R_{gcs} + W_{ccc} + MBK_{CA}\left(h_{ccc1} + h_{gcs1}\right) \\ &\quad + h_{gc2}.PBK_{ccc} + h_{gc3}.PBK_{gcs} \end{aligned}$$

VI. SECURITY ANALYSIS

In this section, we will present the proposed scheme's security analysis, which encompasses both provable security analysis using the ROM oracle model and informal security analysis to evaluate its resilience against well-known attacks, as shown in Tab. III.

A. Provable Security Analysis Using ROM Model

In this subsection, we use the ROM, based on HECDLP assumptions and includes essential technologies like the forking Lemma, to perform a provable security analysis of our certificate-based proxy signature for UAV networks. In the following theorems (Theorem 1,2), we will prove that our scheme is unforgeable against Type 1 (F_{CA1}) and Type 2 (F_{CA2}) forger due to the hardiness of HECDLP.

Theorem 1: Allowing a Type 1 (F_{CA1}) forger to break the EUF-CBPS-CMWA-I security of our certificate-based proxy signature for UAV networks with evident benefit ε_{CA1}, there exists a method C_{CA1} that can solve the hardness of the HECDLP with nonnegligible advantage $\left(1 - \frac{1}{z}\left(\frac{\varepsilon_{CA1}}{zQ_{h_{gc3}}(Q_{\mathrm{CG}}+Q_{\mathrm{CP}}+Q_{\mathrm{PSG}}+Q_{\mathrm{SK}}+1)}\right)\right.$, where z, Q_{CG}, Q_{CP}, Q_{PSG}, Q_{SK}, and $Q_{h_{gc3}}$ denotes the natural logarithm base, certificate generation query, corruption query, proxy signature generation query, secret key generation query, and several random oracle queries to h_{ca3}.

Proof: Suppose an algorithm C_{CA1} that could be used for solving the HECDLP instance like $(d_J, x.d_J)$. For this purpose, in the following sub-steps, we used two entities, i.e., C_{CA1} and $F_{CA1,}$ to correspond with each other to solve HECDLP.

Setup: C_{CA1} selects $SEK_{CA} \in [1, n-1]$, computes $MBK_{CA} = x.d_J$, and generates the standard parameter param P_{CA}.C_{CA1} sends P_{CA} to F_{CA1} and sets $SEK_{CA} = x$. Then, C_{CA1} will give a response to F_{CA1} in the following queries.

Public Key Generation Query: The triple $(I_{ui}, PK_{ui}, PBK_{ui})$ is used when F_{CA1} queries a user's public key generation. If $(I_{ui}, PK_{ui}, PBK_{ui}) \in L_{KG}$, then C_{CA1} sends

PBK_{ui} to F_{CA1}. Otherwise, C_{CA1} does the following computations.

- If $I_{ui} = I_u^*$, then set $PBK^* = x.d_J$ and $PK^* = x$, finally C_{CA1}send PBK^* to F_{CA1} and adds $(I_u^*, PBK^*, \perp)$ into L_{KG}.
- If $I_{ui}! = I_u^*$, then C_{CA1} select $PK_{ui} \in [1, n-1]$ and compute $PBK_{ui} = PK_{ui}.d_J$. Then C_{CA1} will send PBK_{ui} to F_{CA1} and add $(I_{ui}, PK_{ui}, PBK_{ui})$ into L_{KG}.

Secret Key Generation Query: The triple $(I_{ui}, PK_{ui}, PBK_{ui})$ is used when F_{CA1} queries a user secret key generation; if $I_{ui} = I_u^*$, then C_{CA1} stops the game. Otherwise, it does the following computations.

- If $(I_{ui}, PK_{ui}, PBK_{ui}) \in L_{KG}$, then C_{CA1} send PK_{ui} to F_{CA1}.
- Otherwise, it calls *Public Key Generation Query,*searches $(I_{ui}, PK_{ui}, PBK_{ui})$from$L_{KG}$ and sends PK_{ui} to F_{CA1}.

Certificate Generation Query: F_{CA1} sends a query with a tuple (I_{ui}, PBK_{ui}), if $I_{ui} = I_u^*$, then C_{CA1} stops the game. Otherwise, it selects $X_{ui} \in [1, n-1]$, computes $Y_{ui} = X_{ui}.d_J$, $CT_{ui} = X_{ui} + SEK_{CA}.h_{ca1}(I_{ui}, PBK_{ui})$, and send (CT_{ui}, Y_{ui}) to F_{CA1}.

Replaced Public Key Query: When C_{CA1} *received* $PBK^/$ from F_{CA1}, it replaced the triple $(I_{ui}, PK_{ui}, PBK_{ui})$ on $(I_{ui}, PK_{ui}, PBK^/)$ in the list L_{KG}.

Corruption Query: F_{CA1} sends the triple $(I_{ui}, PK_{ui}, PBK_{ui})$ as a corruption query; if $I_{ui} = I_u^*$, then C_{CA1} stops the game; otherwise, it does the following computations.

- If $(I_{ui}, PK_{ui}, PBK_{ui}) \in L_{KG}$, then C_{CA1} send PK_{ui} to F_{CA1}.
- Otherwise, it calls *Public Key Generation Query,*searches $(I_{ui}, PK_{ui}, PBK_{ui})$from$L_{KG}$ and sends PK_{ui} to F_{CA1}.

Random Oracle Queries: F_{CA1}send the random oracle query for h_{ca3}, C_{CA1} check if $\left(I_{gcs}, PBK_{gcs}, w_m, m, R_{gcs}, h_{gc3}\right) \in L_h$, then send h_{gc3} to F_{CA1}. Otherwise, choose h_{gc3} randomly and hand it to h_{ca3}. Finally, include $\left(I_{gcs}, PBK_{gcs}, w_m, m, R_{gcs}, h_{gc3}\right)$ into L_h.

Proxy Signature Query: F_{CA1}send the triple $(I_{ccc}, I_{gcs}, m_i, w_{m_i})$ as a proxy signature query, if $I_{ui} = I_u^*$, then C_{CA1} stop the game; otherwise it does the following computations.

- It calls the key generation algorithm for I_{gcs} and obtained (PK_{gcs}, PBK_{gcs})
- C_{CA1} calls certificate generation algorithm forI_{gcs} and obtained (CT_{gcs}, Y_{gcs})
- Calls delegation generation algorithm for I_{ccc} and obtained (SW_{ccc}, w_m, W_{ccc})
- It selects $F_{gcs} \in [1, n-1]$ and compute $R_{gcs} = Y_{gcs} + F_{gcs}.d_J$
- Compute $SP_{gcs} = SW_{ccc} + CT_{gcs} + F_{gcs} + PK_{gcs}.h_{gc3}$, where h_{gc3} is obtained through random oracle query
- Set $W_{gcs} = W_{ccc}$ and compute $K_{gcs} = PK_{gcs}.PBK_{ccc}$
- Finally, it sends$PS_{gcs} = (SP_{gcs}, K_{gcs}, W_{gcs}, R_{gcs})$ to $F_{CA1}$and adds $(I_{ccc}, I_{gcs}, m_i, w_{m_i})$ into L_{PS}

Forgery: For the tuple $(m^*, w_m^*, I_{ccc}^*, I_{gcs}^*)$,$F_{CA1}$ generate a forged signature by using the following steps.

- Compute $K^* = PK_{gcs}^*.PBK_{ccc}^*$ and set $\mathrm{W}^* = W_{ccc}^*$
- Compute $R^* = Y_{gcs}^* + F_{gcs}^*.d_J$ and $SP^* = SW_{ccc}^* + CT_{gcs}^* + F_{gcs}^* + PK_{gcs}^*.h_{gc3}$
- Finally, it generates the forge proxy signature as $PS_{gcs}^* = (R^*, K^*, \mathrm{W}^*, SP^*)$
- If $I_{ui}! = I_u^*$, then C_{CA1}stop the game; otherwise, by using the forking Lemma, it generates another tuple of forge proxy signature $PS_{gcs}^/ = (R^*, K^*, \mathrm{W}^*, SP^/)$ with $h_{gc3}^/$.
- We can get $SP^* = SW_{ccc}^* + CT_{gcs}^* + F_{gcs}^* + x.h_{gc3}$ and $SP^/ = SW_{ccc}^* + CT_{gcs}^* + F_{gcs}^* + x.h_{gc3}^/$
- If $PS_{gcs}^/$ and PS_{gcs}^*, F_{CA1} give the solution of HECDLP as $x = \frac{SP^* - SP^/}{h_{gc3} - h_{gc3}^/}$.

We define the following events and their probability.

- E^a : For the challenged identity I_{gcs}^*, F_{CA1} requests for Secret Key Generation Query, Corruption Query, Certificate Generation Query, and Proxy Signature Query have not been submitted.The probability of $E^a$is denoted as $\Pr(E^a) = (1 - \frac{1}{Q_{\mathrm{CG}} + Q_{\mathrm{CP}} + Q_{\mathrm{PSG}} + Q_{\mathrm{SK}} + 1})^{Q_{\mathrm{CG}} + Q_{\mathrm{CP}} + Q_{\mathrm{PSG}} + Q_{\mathrm{SK}}}$
- E^b : In the forgery stage, C_{CA1} will not stop the game. The probability of E^b is denoted as$\Pr\left(E^b\right) = \frac{1}{Q_{\mathrm{CG}} + Q_{\mathrm{CP}} + Q_{\mathrm{PSG}} + Q_{\mathrm{SK}} + 1}$
- E^c : In the forgery stage, F_{CA1} must generate two valid forged proxy signatures. The probability of E^c is denoted as$\Pr(E^c) = (1 - \frac{1}{\mathrm{z}}) \frac{\varepsilon_{CA1}}{Q_{h_{gc3}}}$.

We do the following computations:

$$
\begin{aligned}
&\Pr\left[E^a \wedge E^b \wedge E^c\right] \\
&\geq \left(1 - \frac{1}{Q_{\mathrm{CG}} + Q_{\mathrm{CP}} + Q_{\mathrm{PSG}} + Q_{\mathrm{SK}} + 1}\right)^{Q_{\mathrm{CG}} + Q_{\mathrm{CP}} + Q_{\mathrm{PSG}} + Q_{\mathrm{SK}}} \\
&\left(\frac{1}{Q_{\mathrm{CG}} + Q_{\mathrm{CP}} + Q_{\mathrm{PSG}} + Q_{\mathrm{SK}} + 1}\right)\left(1 - \frac{1}{\mathrm{z}}\right)\frac{\varepsilon_{CA1}}{Q_{h_{gc3}}} \\
&\geq \left(1 - \frac{1}{\mathrm{z}}\right)\frac{\varepsilon_{CA1}}{Q_{h_{gc3}}}\left(\frac{1}{Q_{\mathrm{CG}} + Q_{\mathrm{CP}} + Q_{\mathrm{PSG}} + Q_{\mathrm{SK}} + 1}\right) \\
&= \left(1 - \frac{1}{\mathrm{z}}\right)\left(\frac{\varepsilon_{CA1}}{Q_{h_{gc3}}(Q_{\mathrm{CG}} + Q_{\mathrm{CP}} + Q_{\mathrm{PSG}} + Q_{\mathrm{SK}} + 1)}\right)
\end{aligned}
$$

From the above analysis, we say our scheme is unforgeable against F_{CA1}.

Theorem 2: Allowing a Type 2 (F_{CA2}) forger to break the EUF-CBPS-CMWA-II security of our certificate-based proxy signature for UAV networks with evident benefit ε_{CA2}, there exists a method C_{CA2} that can solve the hardness of the HECDLP with nonnegligible advantage $(1 - \frac{1}{z})(\frac{\varepsilon_{CA2}}{z Q_{h_{gc3}}(Q_{\mathrm{CG}} + Q_{\mathrm{CP}} + Q_{\mathrm{PSG}} + Q_{\mathrm{SK}} + 1)})$, where z, Q_{CG}, Q_{CP}, Q_{PSG}, Q_{SK}, and $Q_{h_{gc3}}$ denotes the natural logarithm base, certificate generation query, corruption query, proxy signature generation query, secret key generation query, and several random oracle queries to h_{ca3}.

Proof: Suppose an algorithm C_{CA2} that could be used for solving the HECDLP instance is like $(d_J, x.d_J)$. For this purpose, in the following sub-steps, we used two entities, i.e., C_{CA2} and $F_{CA2,}$ to correspond with each other to solve HECDLP.

Setup: C_{CA2} selects $SEK_{CA} \in [1, n-1]$, computes $MBK_{CA} = SEK_{CA}.d_J$, generates, and sends the standard

parameter P_{CA} along with SEK_{CA} to F_{CA2}. Then, C_{CA2} will give a response to F_{CA2}in the following queries:

Public Key Generation Query: The triple $(I_{ui}, PK_{ui}, PBK_{ui})$ is used when F_{CA2} querying a user's public key generation; if $(I_{ui}, PK_{ui}, PBK_{ui}) \in L_{KG}$, then C_{CA2} sends PBK_{ui} to F_{CA1}. Otherwise, C_{CA2} does the following computations.

- If $I_{ui} = I_u^*$, then set $PBK^* = x.d_J$ and $PK^* = x$, finally C_{CA2}send $PBK^*$to F_{CA2} and adds $(I_u^*, PBK^*, \perp)$ into L_{KG}.
- If $I_{ui} != I_u^*$, then C_{CA2}select$PK_{ui} \in [1, n-1]$ and compute $PBK_{ui} = PK_{ui}.d_J$. Then C_{CA2} will send PBK_{ui} to F_{CA2} and adds $(I_{ui}, PK_{ui}, PBK_{ui})$ into L_{KG}.

Secret Key Generation Query: The triple $(I_{ui}, PK_{ui}, PBK_{ui})$ is used when F_{CA2} queries a user secret key generation; if $I_{ui} = I_u^*$, then C_{CA2} stops the game; otherwise, it does the following computations.

- If $(I_{ui}, PK_{ui}, PBK_{ui}) \in L_{KG}$, then C_{CA2} send PK_{ui} to F_{CA2}.
- Otherwise, it calls *Public Key Generation Query,*searches $(I_{ui}, PK_{ui}, PBK_{ui})$from$L_{KG}$ and sends PK_{ui} to F_{CA2}.

Corruption Query: F_{CA2} sends the triple $(I_{ui}, PK_{ui}, PBK_{ui})$ as a corruption query; if $I_{ui} = I_u^*$, then C_{CA2} stops the game; otherwise, it does the following computations.

- If $(I_{ui}, PK_{ui}, PBK_{ui}) \in L_{KG}$, then C_{CA2} send PK_{ui} to F_{CA2}.
- Otherwise, it calls *Public Key Generation Query,*searches $(I_{ui}, PK_{ui}, PBK_{ui})$from$L_{KG}$ and sends PK_{ui} to F_{CA2}.

Random Oracle Queries: F_{CA2} send the random oracle query for h_{ca3}, C_{CA1} check if $(I_{gcs}, PBK_{gcs}, w_m, m, R_{gcs}, h_{gc3}) \in L_h$, then send h_{gc3} to F_{CA2}. Otherwise, choose h_{gc3} randomly and hand it to h_{ca3}. Finally, include $(I_{gcs}, PBK_{gcs}, w_m, m, R_{gcs}, h_{gc3})$ into L_h.

Proxy Signature Query: F_{CA2} send the triple $(I_{ccc}, I_{gcs}, m_i, w_{m_i})$ as a proxy signature query; if $I_{ui} = I_u^*$, then C_{CA1} stop the game; otherwise, it does the following computations.

- It calls the key generation algorithm for I_{gcs} and obtained (PK_{gcs}, PBK_{gcs})
- C_{CA2} calls certificate generation algorithm for I_{gcs} and obtained (CT_{gcs}, Y_{gcs})
- Calls delegation generation algorithm for I_{ccc} and obtained (SW_{ccc}, w_m, W_{ccc})
- It selects $F_{gcs} \in [1, n-1]$ and compute $R_{gcs} = Y_{gcs} + F_{gcs}.d_J$
- Compute $SP_{gcs} = SW_{ccc} + CT_{gcs} + F_{gcs} + PK_{gcs}.h_{gc3}$, where h_{gc3} is obtained through random oracle query
- Set $W_{gcs} = W_{ccc}$ and compute $K_{gcs} = PK_{gcs}.PBK_{ccc}$
- Finally, it sends $PS_{gcs} = (SP_{gcs}, K_{gcs}, W_{gcs}, R_{gcs})$ to $F_{CA2}$and add $(I_{ccc}, I_{gcs}, m_i, w_{m_i})$ into L_{PS}

Forgery: For the tuple $(m^*, w_m^*, I_{ccc}^*, I_{gcs}^*)$,$F_{CA2}$ generates a forged signature using the following steps.

- Compute $K^* = PK_{gcs}^*.PBK_{ccc}^*$ and set $W^* = W_{ccc}^*$
- Compute $R^* = Y_{gcs}^* + F_{gcs}^*.d_J$ and $SP^* = SW_{ccc}^* + CT_{gcs}^* + F_{gcs}^* + PK_{gcs}^*.h_{gc3}$
- Finally, it generates the forge proxy signature as $PS_{gcs}^* = (R^*, K^*, W^*, SP^*)$
- If $I_{ui} != I_u^*$, then C_{CA2}stop the game, otherwise by using forking Lemma, it generates another tuple of forge proxy signature $PS_{gcs}^{/} = (R^*, K^*, W^*, SP^{/})$ with $h_{gc3}^{/}$.
- So, we can get $SP^* = SW_{ccc}^* + CT_{gcs}^* + F_{gcs}^* + x.h_{gc3}$ and $SP^{/} = SW_{ccc}^* + CT_{gcs}^* + F_{gcs}^* + x.h_{gc3}^{/}$
- If $PS_{gcs}^{/}$and PS_{gcs}^*, F_{CA2} give the solution of HECDLP as $x = \frac{SP^* - SP^{/}}{h_{gc3} - h_{gc3}^{/}}$.

We define the following events and their probability.

- E^a :For the challenged identity I_{gcs}^*, F_{CA2} does not submit a request for a *Secret Key Generation Query, Corruption Query, Certificate Generation Query,* and *Proxy Signature Query.*The probability of$E^a$is denoted as$\Pr(E^a) = (1 - \frac{1}{Q_{CG}+Q_{CP}+Q_{PSG}+Q_{SK}+1})^{Q_{CG}+Q_{CP}+Q_{PSG}+Q_{SK}}$
- E^b :In the forgery stage,C_{CA2}will not stop the game. The probability of$E^b$is denoted as$\Pr(E^b) = \frac{1}{Q_{CG}+Q_{CP}+Q_{PSG}+Q_{SK}+1}$
- E^c :In the forgery stage,F_{CA2} must generate two valid forge proxy signatures. The probability of$E^c$is denoted as$Pr(E^c) = (1 - \frac{1}{z})\frac{\varepsilon_{CA2}}{Q_{h_{gc3}}}$.

We do the following computations:

$$
\begin{aligned}
&\Pr\left[E^a \wedge E^b \wedge E^c\right] \\
&\geq \left(1 - \frac{1}{Q_{CG}+Q_{CP}+Q_{PSG}+Q_{SK}+1}\right)^{Q_{CG}+Q_{CP}+Q_{PSG}+Q_{SK}} \\
&\left(\frac{1}{Q_{CG}+Q_{CP}+Q_{PSG}+Q_{SK}+1}\right)\left(1-\frac{1}{z}\right)\frac{\varepsilon_{CA2}}{Q_{h_{gc3}}} \\
&\geq \left(1-\frac{1}{z}\right)\frac{\varepsilon_{CA1}}{Q_{h_{gc3}}}\left(\frac{1}{Q_{CG}+Q_{CP}+Q_{PSG}+Q_{SK}+1}\right) \\
&= \left(1-\frac{1}{z}\right)\left(\frac{\varepsilon_{CA2}}{Q_{h_{gc3}}(Q_{CG}+Q_{CP}+Q_{PSG}+Q_{SK}+1)}\right)
\end{aligned}
$$

Based on the above analysis, we conclude that the proposed scheme is unforgeable against F_{CA2}.

B. Informal Security Analysis

The proposed scheme ensures the fulfillment of the following security requirements.

1) Unforgeability: The proposed scheme provides both the unforgeability of delegation and proxy signature. In an event when internal or external attackers want to generate a forge signature $SW_{ccc} = CT_{ccc} + F_{ccc} + PK_{ccc}.h_{ca2}(I_{ccc}, PBK_{ccc}, w_m, PK_{ccc}.PBK_{gcs})$ in *Delegation Generation*phase, for this, he/she needs two unknown values PK_{ccc} and F_{ccc} which is only known to CCC. It is mathematically not feasible to solve the equation, which has two unknown values. Similarly, if an internal or external attacker wants to generate a forged signature $SP_{gcs} = W_{ccc} + CT_{gcs} + F_{gcs} + PK_{gcs}.h_{ca3}(I_{gcs}, PBK_{gcs}, w_m, m, R_{gcs})$in *Proxy Signature Generation*phase, he/she needs two unknown values PK_{gcs} and F_{gcs}, which is only known to GCS, and it is mathematically not feasible to solve the equation which has two unknown values. From the above discussions, it is proved that the proposed scheme

TABLE IV
COMPARISON OF COMPUTATION COST WITH MAJOR OPERATIONS

Schemes	Delegation	Delegation Verifications	Proxy Signature	Proxy Signature Verifications	Total
Verma et al. [21]	ICPS_{PBM}	2ICPS_{P}	ICPS_{PBM}	$2\text{ICPS}_{PBM} + 2\text{ICPS}_{P}$	$4\text{ICPS}_{PBM} + 4\text{ICPS}_{P}$
He et al. [22]	2ICPS_{PBM}	2ICPS_{PBM}	$4\text{ICPS}_{PBM} + 2\text{ICPS}_{P}$	$4\text{ICPS}_{PBM} + \text{ICPS}_{P}$	$8\text{ICPS}_{PBM} + 3\text{ICPS}_{P}$
Xu et al. [23]	3ICPS_{PBM}	4ICPS_{PBM}	$3\text{ICPS}_{PBM} + \text{ICPS}_{P}$	$\text{ICPS}_{PBM} + \text{ICPS}_{P}$	$11\text{ICPS}_{PBM} + 2\text{ICPS}_{P}$
Proposed Scheme	3ICPS_{HEDM}	4ICPS_{HEDM}	3ICPS_{HEDM}	4ICPS_{HEDM}	14ICPS_{HEDM}

TABLE V
COMPARISON OF COMPUTATION COST (IN MS)

Schemes	Delegation	Delegation Verification	Proxy Signature	Proxy Signature Verification	Total
Verma et al. [21]	2.2560	9.2056	2.2560	13.7176	27.4352
He et al. [22]	4.512	4.512	18.4112	13.6268	41.062
Xu et al. [23]	6.768	9.024	11.3708	6.8588	34.0216
Proposed Scheme	1.1472	1.5296	1.1472	1.5296	5.3536

TABLE VI
COMPARISON OF COMMUNICATION COST WITH MAJOR OPERATIONS

Schemes	Delegation size	Proxy Signature Size	Total
Verma et al. [21]	$\text{ICPS}_{G} + \text{ICPS}_{w_m}$	$\text{ICPS}_{G} + \text{ICPS}_{w_m} + \text{ICPS}_{m}$	$2\text{ICPS}_{G} + 2\text{ICPS}_{w_m} + \text{ICPS}_{m}$
He et al. [22]	$2\text{ICPS}_{G} + \text{ICPS}_{w_m}$	$2\text{ICPS}_{G} + \text{ICPS}_{w_m} + \text{ICPS}_{m}$	$4\text{ICPS}_{G} + 2\text{ICPS}_{w_m} + \text{ICPS}_{m}$
Xu et al. [23]	$2\text{ICPS}_{G} + \text{ICPS}_{w_m}$	$2\text{ICPS}_{G} + \text{ICPS}_{w_m} + \text{ICPS}_{m}$	$4\text{ICPS}_{G} + 2\text{ICPS}_{w_m} + \text{ICPS}_{m}$
Proposed Scheme	$2\text{CBPS}_{n} + \text{ICPS}_{w_m}$	$4\text{ICPS}_{n} + \text{ICPS}_{w_m} + \text{ICPS}_{m}$	$6\text{ICPS}_{n} + 2\text{ICPS}_{w_m} + \text{ICPS}_{m}$

provides both the unforgeability of delegation and proxy signature.

2) Resist against Reply Attack: The proposed scheme is safeguarded from reply attacks. In communicating proxy signatures, the GCS sends $PS_{gcs} = (SP_{gcs}, K_{gcs}, W_{gcs}, R_{gcs}, T_{gcs})$ as a signature tuple, where T_{gcs} represents a time stamp. When $PS_{gcs} = (SP_{gcs}, K_{gcs}, W_{gcs}, R_{gcs}, T_{gcs})$ is received by the UAV, it checks the validity of T_{gcs}; if it is valid, UAV verifies the proxy signature.
3) Resist against Impersonation Attacks: The proposed scheme is safeguarded from impersonation attacks. In our proposed proxy signature scheme, the attacker will not be able to impersonate the legitimate user because in the certificate generation phase, CA selects $X_{ui} \in [1, n-1]$, computes $Y_{ui} = X_{ui}.d_J$, $CT_{ui} = X_{ui} + SEK_{CA}.h_{ca1}(I_{ui}, PBK_{ui})$, and send CT_{ui} to user with identity (I_{ui}) by using an open network and stores (CT_{ui}, I_{ui}) in his database to avoid an impersonation attack.
4) Man-In-the-Middle Attack: In our proposed proxy signature scheme, the attacker will not be able to impersonate the legitimate user because in the certificate generation phase, CA selects $X_{ui} \in [1, n-1]$, compute $Y_{ui} = X_{ui}.d_J$, $CT_{ui} = X_{ui} + SEK_{CA}.h_{ca1}(I_{ui}, PBK_{ui})$, and send CT_{ui} to the user with identity (I_{ui}) by using an open network and store (CT_{ui}, I_{ui}) in his database to avoid an impersonation attack. Further, In an event when an internal or external attacker wants to generate a forge signature $SW_{ccc} = CT_{ccc} + F_{ccc} + PK_{ccc}.h_{ca2}(I_{ccc}, PBK_{ccc}, w_m, PK_{ccc}.PBK_{gcs})$ in *Delegation Generation*phase, they need two unknown values PK_{ccc} and F_{ccc} which is only known to CC. It is mathematically impossible to solve an equation with two unknown values. Similarly, if an internal or external attacker wants to generate a forged signature $SP_{gcs} = W_{ccc} + CT_{gcs} + F_{gcs} + PK_{gcs}.h_{ca3}(I_{gcs}, PBK_{gcs}, w_m, m, R_{gcs})$in *Proxy Signature Generation*phase, they need two unknown values PK_{gcs} and F_{gcs}, which is only known to GCS, and it is

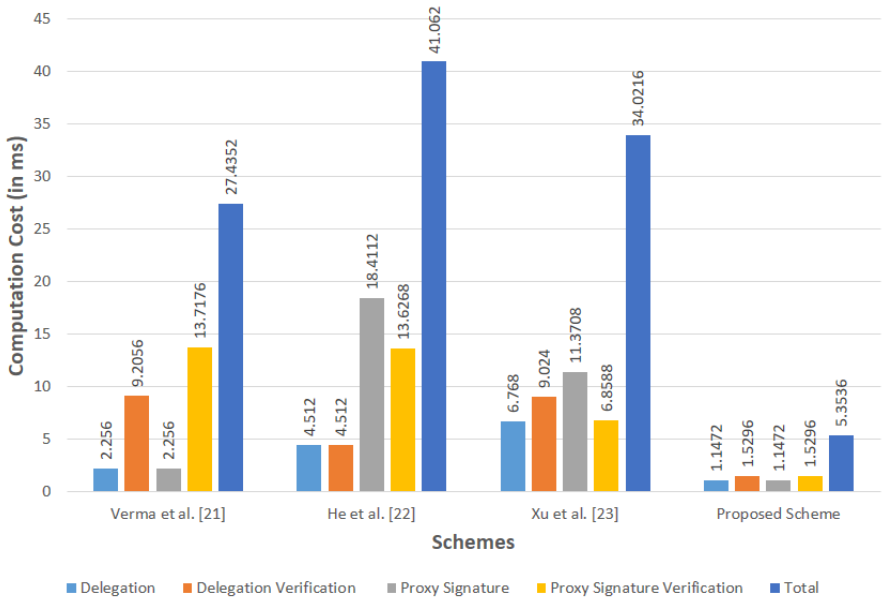

Fig. 3. Comparative analysis based on computation cost (in ms).

TABLE VII
COMPARISON OF COMMUNICATION COST (IN BITS)

Schemes	Delegation Size	Proxy Signature Size	Total
Verma et al. [21]	1184	1344	2528
He et al. [22]	2208	2528	4736
Xu et al. [23]	2208	2528	4736
Proposed Scheme	320	800	1120

Note: We assume that $\text{ICPS}_G = 1024\ bits$,$\text{ICPS}_{w_m} = 160\ bits$, $\text{ICPS}_m = 160 bits$, and $\text{ICPS}_n = 80\ bits$.

mathematically not feasible to solve the equation, which has two unknown values. From the above discussions, it is proved that the proposed scheme provides both the unforgeability of delegation and proxy signature.

VII. PERFORMANCE EVALUATION

In this section, we analyze the performance of the proposed scheme in terms of security, computation and communication costs with the schemes offered by Verma et al. [21], He et al. [22] and Xu et al. [23]. In terms of security, we have compared our scheme in terms of unforgeability, replay attack, impersonation attack, and man-in-the-middle attack, which is presented in Tab. III. So, according to Tab. III, our scheme provides all the above security requirements and existing schemes only meet the criteria for unforgeability.

For computation cost, we used the MIRACL library to simulate cryptographic operations on a Windows 10 laptop with an Intel i7–1195G7 @2.9 GHz processor and 8 GB of memory [25]. The run time of bilinear pairing operation (ICPS_P) is 4.6028, bilinear pairing-based multiplication ICPS_{PBM} is 2.2560 and elliptic curve multiplication ICPS_{ECM} is 0.7648 [25]. We assume hyperelliptic curve divisor multiplication $\text{ICPS}_{HEDM} = 0.3824$[26]. As shown in Tab. IV, V, and Fig. 3, it has been demonstrated that the proposed scheme had a lower computation cost in comparison to the extant schemes proposed by Verma et al. [21], He et al. [22], and Xu et al. [23]. Similarly, as detailed in Tab. VI, VII and depicted in Fig. 4, the proposed scheme has lower communication costs than its counterparts. These findings verify the usefulness of the proposed scheme for UAVs with limited resources.

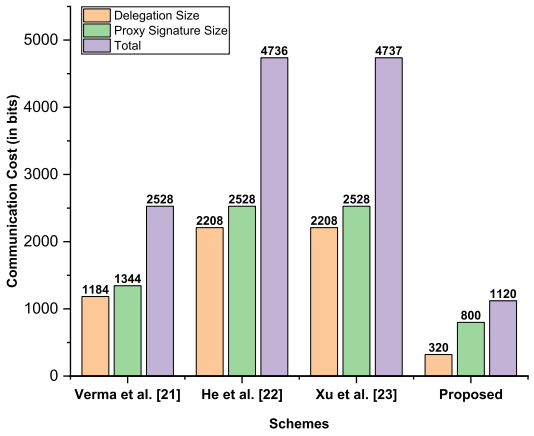

Fig. 4. Comparative analysis based on communication cost (in bits).

VIII. CONCLUSION

In this article, we proposed an improvised certificate-based proxy signature (ICPS), which used the hyperelliptic curve cryptography (HECC) concept to mitigate security concerns

in UAV communication. We utilized the random oracle model (ROM) in the proposed scheme to conduct provable security assessments, demonstrating its resilience against well-known cyber-security threats. The performance analysis of the proposed ICPS scheme is performed with similar existing schemes, which reveals the proposed scheme's efficiency in computation and communication costs. The computation cost analysis of the proposed ICPS scheme in this study observed a significantly lower computation cost of 5.3536 ms. On the other hand, the computation cost of the existing schemes was observed as 27.4352 ms, 41.062 ms, and 34.0216 ms, as presented by Verma et al. [21], He et al. [22], and Xu et al. [23] respectively. Moreover, while considering communication cost, the proposed ICPS scheme demonstrated remarkable efficiency with a communication cost of only 1120 bits, in comparison to the significantly higher communication costs of the existing schemes as 2528 bits, 4736 bits, and 4736 bits associated with the schemes of Verma et al. [21], He et al. [22], and Xu et al. [23]. This comprehensive analysis of computation and communication costs explicitly supports the proposed scheme's better efficiency than its counterparts.

REFERENCES

[1] S. A. H. Mohsan, N. Q. H. Othman, Y. Li, M. H. Alsharif, and M. A. Khan, "Unmanned aerial vehicles (UAVs): Practical aspects, applications, open challenges, security issues, and future trends," *Intell. Service Robot.*, vol. 16, pp. 109–137, Jan. 2023.

[2] M. Lyu, Y. Zhao, C. Huang, and H. Huang, "Unmanned aerial vehicles for search and rescue: A survey," *Remote Sens.*, vol. 15, no. 13, p. 3266, Jun. 2023.

[3] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2624–2661, 4th Quart., 2016.

[4] K. AL-Dosari, Z. Hunaiti, and W. Balachandran, "Systematic review on civilian drones in safety and security applications," *Drones*, vol. 7, no. 3, p. 210, 2023.

[5] H. Shakhatreh et al., "Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges," *IEEE Access*, vol. 7, pp. 48572–48634, 2019.

[6] M. Adil, M. A. Jan, Y. Liu, H. Abulkasim, A. Farouk, and H. Song, "A systematic survey: Security threats to UAV-aided IoT applications, taxonomy, current challenges and requirements with future research directions," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 1437–1455, Feb. 2023.

[7] V. Cichella et al., "Safe coordinated maneuvering of teams of multirotor unmanned aerial vehicles: A cooperative control framework for multi-vehicle, time-critical missions," *IEEE Control Syst. Mag.*, vol. 36, no. 4, pp. 59–82, Aug. 2016.

[8] M. A. Khan et al., "Swarm of UAVs for network management in 6G: A technical review," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 1, pp. 741–761, Mar. 2023.

[9] M. A. Khan, B. A. Alzahrani, A. Barnawi, A. Al-Barakati, A. Irshad, and S. A. Chaudhry, "A resource friendly authentication scheme for space–air–ground–sea integrated maritime communication network," *Ocean Eng.*, vol. 250, Apr. 2022, Art. no. 110894.

[10] H. Wang, H. Zhao, J. Zhang, D. Ma, J. Li, and J. Wei, "Survey on unmanned aerial vehicle networks: A cyber physical system perspective," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 1027–1070, 2nd Quart., 2020.

[11] S. Kim, S. Park, and D. Won, "Proxy signatures, revisited," in *Proc. Int. Conf. Inf. Commun. Secur.* Berlin, Germany: Springer, 1997, pp. 223–232.

[12] F. Zhang and K. Kim, "Efficient ID-based blind signature and proxy signature from bilinear pairings," in *Proc. Australas. Conf. Inf. Secur. Privacy*, Perth, WA, Australia. Berlin, Germany: Springer, Jan. 2003, pp. 312–323.

[13] G. K. Verma, N. Kumar, P. Gope, B. B. Singh, and H. Singh, "SCBS: A short certificate-based signature scheme with efficient aggregation for Industrial-Internet-of-Things environment," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 9305–9316, Jun. 2021.

[14] R. Elhabob, Y. Zhao, I. Sella, and H. Xiong, "Efficient certificateless public key cryptography with equality test for Internet of Vehicles," *IEEE Access*, vol. 7, pp. 68957–68969, 2019.

[15] S. S. Al-Riyami and K. G. Paterson, "Certificate-less public key cryptography," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, Taipei, Taiwan. Berlin, Germany: Springer, 2003, pp. 452–473.

[16] A. Shamir, "Identity-based cryptosystem and signatures schemes," in *Advances in Cryptology*, G. R. Blakley and D. Chaum, Eds., Berlin, Germany: Springer, 1984, pp. 47–53.

[17] M. L. Pura and D. Buchs, "A self-organized key management scheme for ad hoc networks based on identity-based cryptography," in *Proc. 10th Int. Conf. Commun. (COMM)*, Bucharest, Romania, May 2014, pp. 1–4.

[18] I. Ullah, N. U. Amin, A. Almogren, M. A. Khan, M. I. Uddin, and Q. Hua, "A lightweight and secured certificate-based proxy signcryption (CB-PS) scheme for E-Prescription systems," *IEEE Access*, vol. 8, pp. 199197–199212, 2020, doi: 10.1109/ACCESS.2020.3033758.

[19] L. He, J. Ma, R. Mo, and D. Wei, "Designated verifier proxy blind signature scheme for unmanned aerial vehicle network based on mobile edge computing," *Secur. Commun. Netw.*, vol. 2019, pp. 1–12, Apr. 2019.

[20] M. Mambo, K. Usuda, and K. Okamoto, "Proxy signatures: Delegation of the power to sign messages," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. 79, pp. 1338–1354, Sep. 1996.

[21] G. K. Verma, B. B. Singh, N. Kumar, and D. He, "CB-PS: An efficient Short-Certificate-Based proxy signature scheme for UAVs," *IEEE Syst. J.*, vol. 14, no. 1, pp. 621–632, Mar. 2020.

[22] L. He, J. Ma, L. Shen, and D. Wei, "Certificateless designated verifier proxy signature scheme for unmanned aerial vehicle networks," *Sci. China Inf. Sci.*, vol. 64, no. 1, Jan. 2021, Art. no. 112101.

[23] Z. Xu, M. Luo, P. Vijayakumar, C. Peng, and L. Wang, "Efficient certificateless designated verifier proxy signature scheme using UAV network for sustainable smart city," *Sustain. Cities Soc.*, vol. 80, May 2022, Art. no. 103771.

[24] Z. Qiao, Y. Zhou, B. Yang, M. Zhang, T. Wang, and Z. Xia, "Secure and efficient certificate-based proxy signature schemes for Industrial Internet of Things," *IEEE Syst. J.*, vol. 16, no. 3, pp. 4719–4730, Sep. 2022.

[25] L. Zhou and X. Yin, "An improved pairing-free certificateless aggregate signature scheme for healthcare wireless medical sensor networks," *PLoS ONE*, vol. 17, no. 7, Jul. 2022, Art. no. e0268484.