Research Repository

An Improved Aggregation-Based Signcryption for Secure Drone to Ground Station Communication System

An Improved Aggregation-Based Signcryption for Secure Drone to Ground Station Communication System

Insaf Ullah[a], Syed Tariq Shah[b,*], Wali Ullah Khan[c], Mahmoud A. Shawky[d], Ahmad Almogren[e,*] and Haralambos Mouratidis[a]

[a] *Institute for Analytics and Data Science University of Essex, CO4 3SQ, Colchester, UK*
[b] *School of Computer Science and Electronic Engineering, University of Essex, CO4 3SQ, Colchester, UK*
[c] *School of Computing, Gachon University, Seongnam-si, 13120, Republic of Korea*
[d] *James Watt School of Engineering, University of Glasgow, G12 8QQ, Glasgow, UK*
[e] *Department of Computer Science, College of Computer and Information Sciences, King Saud University, 11633, Riyadh, Saudi Arabia*

ARTICLE INFO

Keywords:
Hyperelliptic Curve Cryptography (HECC)
Aggregation-based Signcryption
Drone Communication Security
Forgery and Impersonation Resistance
Lightweight Cryptographic Protocols.

ABSTRACT

Secure and efficient communication in drone-assisted networks is critical for maintaining the integrity of cyber-physical systems. This paper presents a rigorous cryptographic analysis of an existing aggregation-based signcryption scheme, revealing key vulnerabilities, including susceptibility to forgery, impersonation attacks, and inconsistencies in key generation and verification processes. To overcome these limitations, we propose a novel aggregation-based signcryption framework built upon hyperelliptic curve cryptography (HECC), offering enhanced security and lightweight computation. The proposed scheme is formally proven to satisfy fundamental cryptographic properties, including confidentiality, authentication, integrity, unforgeability, and impersonation resistance under the hardness assumption of the hyperelliptic curve discrete logarithm problem (HECDLP). Performance evaluation demonstrates that our approach achieves up to 36% reduction in computational cost and 12% lower communication overhead compared to the schemes of Verma et al. [1], Aithekar et al. [2], and Ali et al. [3]. These results confirm the practical applicability of our design for secure and efficient drone-to-ground station communication in resource-constrained environments.

1. Introduction

As the internet-of-Drones (IoD) emerges as a pivotal innovation within the internet-of-things (IoT) ecosystem, enabling real-time communication and collaboration between unmanned-aerial-vehicles (UAVs) and ground stations (GS), the challenge of ensuring secure, efficient data transmission in such systems becomes increasingly critical [4]. This burgeoning field extends UAV applications beyond military use into various industries such as logistics, environmental monitoring, smart agriculture, and disaster management [5, 6, 7]. In these IoD environments, UAVs collect data and transmit it to a central GS for processing, typically over open and vulnerable wireless channels [4, 8, 9]. The inherent vulnerability of these wireless communications exposes IoD systems to threats such as eavesdropping, impersonation, and data tampering, which can compromise data integrity and result in inaccurate decision-making at the GS [10] [3]. Moreover, since multiple UAVs simultaneously transmit data to a shared GS, ensuring both confidentiality and authentication of the transmitted data is crucial. More specifically, a highly efficient authentication verification mechanism is imperative to handle multiple inputs in real time without introducing significant computational delays [11].

Cryptographic methods, such as signcryption, which combines encryption and digital signatures into a single operation, have emerged as an efficient means to secure IoD communications [12, 13]. In particular, aggregated signcryption (AS), a pairing-based approach, has been introduced to provide simultaneous authentication and confidentiality while improving efficiency on the receiver side [14]. This approach enables the GS to verify multiple signcrypted messages from different UAVs in a single operation, thereby reducing the computational burden. A well-known implementation of AS is the identity-based aggregated signcryption (IDAS) scheme, which leverages identity-based cryptography to streamline the verification process [15].

However, relying on pairing-based operations in IDAS presents a challenge, as it introduces considerable computational overhead during the verification phase [16, 17]. In response to this issue, researchers have proposed several optimised IDAS variants that are designed to enhance verification efficiency by minimising the reliance on costly pairing operations [18, 19, 20, 21, 22]. These advancements aim to provide a more scalable and resource-efficient framework that addresses the security demands of increasingly complex IoD environments. Although AS has been utilised in various data aggregation (DA) protocols, its application in UAV-to-Ground Station (U2GS) communication remains underdeveloped. The AS techniques, though extensively utilised in DA schemes for wireless sensor networks (WSNs) such as IoT-enabled maritime transportation systems [22, 23], have not yet fully addressed the unique security requirements imposed by U2GS communication [24, 25].

insaf.ullah@essex.ac.uk (I. Ullah); syed.shah@essex.ac.uk (S.T. Shah); waliullahkhan30@gmail.com (W.U. Khan); mahmoud.shawky@glasgow.ac.uk (M.A. Shawky); Ahalmogren@ksu.edu.sa (A. Almogren); h.mouratidis@essex.ac.uk (Haralambos Mouratidis)

ORCID(s): 0009-0008-7095-3831 (I. Ullah); 0000-0003-4722-1786 (S.T. Shah); 0000-0003-1485-5141 (W.U. Khan); 0000-0003-3393-8460 (M.A. Shawky); 0000-0002-8253-9709 (A. Almogren); 0000-0002-2599-0712 (Haralambos Mouratidis)

Several existing studies have explored DA protocols in UAV-based communication, focusing on energy efficiency, routing, and trajectory optimisation. In [26], the benefits of a certificate-less cryptographic setting were highlighted to reduce the overhead associated with managing public key certificates. However, this approach comes with a high computational cost on the UAV side, impacting the overall performance in scenarios requiring real-time DA between U2GS. The authors in [27] optimised sensor clustering in WSNs by proposing the use of a UAV as a data mule to collect data from cluster heads (CHs), thereby significantly improving energy efficiency. Despite its advantages in energy savings, this design did not focus on the DA between U2GS, a critical element in IoD-based applications. Bera et al. [28] introduced a cost-effective DA protocol where UAVs served as aerial base stations, optimising their trajectory for IoT data collection. While this work focused on minimising travel time and optimising UAV routes, it similarly neglected the critical challenge of ensuring secure and efficient DA between multiple UAVs and a shared GS, leaving a gap in addressing the security and efficiency concerns in U2GS communication. As UAVs become crucial for data collection in monitoring applications such as environmental surveillance, disaster management, and healthcare, the need for a secure and efficient DA method tailored for U2GS communication is becoming increasingly imminent. Current solutions mainly focus on trajectory optimisation and clustering but fail to address the unique and pressing security and efficiency challenges of DA between U2GS [2]. A robust solution must minimise computational overhead and ensure confidentiality, integrity, and data authentication across inherently vulnerable wireless channels [3]. To address these challenges, this paper introduces a secure and lightweight signcryption protocol with aggregate verification. This solution represents the first efficient and secure DA (ES-DA) scheme specifically designed for IoD-based U2GS communication.

1.1. Related Works

Drones are increasingly utilised in various monitoring applications, including environmental surveillance, industrial inspections, and societal safety. According to a report by Global Market Insights, the global commercial drone market was valued at USD 14.6 billion in 2023 and is projected to grow at a compound annual growth rate (CAGR) of over 13.5% between 2024 and 2032, reaching approximately USD 44.5 billion by 2032 [29]. However, the reliance of IoD systems on wireless communication channels exposes them to risks such as data tampering and unauthorised access to sensitive information. Addressing these vulnerabilities, the authors in [30] have introduced a privacy-preserving authentication framework using physically unclonable functions (PUFs) to enhance security without requiring UAVs to store sensitive credentials, thus protecting against physical capture and reducing computational load [30].

In pursuit of securing communication within IoD networks, a certificateless signcryption approach was proposed in [31], facilitating key agreements between drones and GS. This approach, termed the certificateless-signcryption tag-key encapsulation mechanism (eCLSC-TKEM), aims to deliver efficient cryptographic protocols without the complexities of certificate management and mitigate the key escrow problem. Although eCLSC-TKEM reduces the computational load on drones, achieving efficiency gains, it introduces key overhead that challenges scalability, particularly within high-demand IoD settings.

To address these scalability issues, an enhanced suite of protocols was later introduced in [26], which supports a range of IoD communication types, including one-to-one key agreement, one-to-many multi-recipient encryption, and many-to-one data aggregation. This design enhances the flexibility of IoD applications, enabling secure communications with multiple smart objects and streamlined data aggregation. Nonetheless, the inherent limitations of certificateless systems, including specific security vulnerabilities, persisted, underscoring the need for further advancements to meet the security and efficiency requirements of expansive IoD deployments. To enhance the energy consumption of sensor nodes in a UAV-based network, a data aggregation scheme is proposed in [27]. In the proposed scheme, sensors collect data and offload it to a UAV, acting as a data mule, optimising energy usage across the network by handling the bulk of computation and communication. Although effective in conserving sensor energy, this protocol does not address security measures, leaving communication links potentially vulnerable to interception. In a similar effort to enhance IoD communication efficiency, Wazid et al. in [32] developed a lightweight user authentication and key agreement scheme that leverages bitwise XOR and hash functions to support secure, low-resource authentication. However, the scheme lacks efficient batch verification, which poses scalability limitations in larger IoD environments. To further strengthen security, Gope and Sikdar in [30] proposed an authentication scheme that integrates Physically Unclonable Functions (PUFs) with hash and XOR operations, improving protection against physical attacks without requiring drones to store sensitive credentials. Similarly, Alladi et al. in [33] presented an authentication framework using PUFs to secure drone-to-drone (D2D) and drone-to-infrastructure (D2I) communications. Despite its advantages, this approach remains susceptible to certain modelling attacks, highlighting the need for continued refinement.

To further improve the robustness of IoD systems, a key agreement protocol based on a fuzzy extractor was introduced in [34], enhancing security by deriving stable cryptographic keys from noisy biometric inputs to ensure adaptive and secure key generation. In parallel, a two-stage mutual authentication protocol tailored for SDN-supported IoD networks was proposed in [35], utilising PUFs to bolster resilience against physical and cloning attacks. Similarly, an identity-based cryptography approach is employed to develop a lightweight authentication mechanism optimised for reduced computational overhead in resource-constrained IoD environments in [36]. Building on lightweight methods, an authentication scheme incorporating a chaotic map for enhanced privacy preservation in IoD communication was designed in [37], dynamically adjusting security parameters to meet IoD communication needs. For intrusion detection, an AI-driven system for proactive threat identification in IoD-based networks is developed in [38], leveraging machine learning to anticipate and mitigate potential threats. For cost-efficient data aggregation, the CEDAN framework was proposed in [28], optimising UAV flight paths and processing tasks to minimise travel time and computational costs in drone-enabled IoT deployments. Furthermore, to address the security and efficiency challenges in U2GS communication, Verma et al. [1] developed an efficient and secure signcryption-based data aggregation scheme. This protocol incorporates aggregated verification to reduce computational overhead at the receiver end while ensuring data confidentiality and authentication in a single step. It is specifically tailored for IoD-based D2GS communication, making it a robust solution for securing data collection in high-demand IoD applications.

1.2. Contribution

Based on the above literature review and discussion, this paper presents a comprehensive analysis and enhancement of cryptographic schemes for secure drone-to-ground station communication. We begin by critically evaluating the scheme proposed by Verma et al. in [1], identifying significant vulnerabilities, including mathematical inconsistencies, susceptibility to forgery by outsider attackers, and lack of resistance to impersonation attacks. To address these shortcomings, we propose an improved aggregation-based signcryption scheme leveraging hyperelliptic curve cryptography (HECC), which ensures robust security properties such as confidentiality, authentication, integrity, unforgeability, and resistance to impersonation attacks. Furthermore, a comparative analysis demonstrates that our scheme outperforms existing approaches regarding computational efficiency, communication overhead, and adherence to critical security requirements. The Contributions of this paper can be summarised as:

- We present a detailed analysis of the cryptographic scheme proposed by Verma et al. in [1], identifying areas for improvement, including certain mathematical inconsistencies, vulnerabilities to forgery by outsider attackers, and susceptibility to impersonation attacks. Additionally, the reliance on elliptic curve cryptography renders this scheme unsuitable for resource-constrained devices due to the larger key and parameter sizes involved. This examination serves as a foundation for enhancing the security, robustness, and overall efficiency of cryptographic schemes, particularly in terms of computational cost and communication overhead.

- We propose an improved aggregation-based signcryption scheme designed for drone-to-ground station communication systems, utilising HECC's efficiency and security advantages. This approach addresses the limitations observed in prior work while maintaining computational efficiency.

- The proposed scheme is rigorously designed to meet key security requirements, including confidentiality, authentication, integrity, unforgeability, and resistance to impersonation attacks, ensuring a robust framework for secure communication.

- Through a comprehensive comparative analysis, we demonstrate that our scheme offers significant improvements in computational efficiency, communication overhead, and adherence to security requirements compared to existing schemes, making it a practical and effective solution for real-world applications.

2. Review of Verma et al.'s Scheme

This section outlines the mathematical steps involved in constructing Verma et al.'s scheme [1] and discusses its identified security limitations.

2.1. Construction of Verma et al.'s Scheme

The scheme comprises five phases: Initialization, AID-Gen, Key-Gen, Data-Aggregate, and Verify-Decryption. These phases are described in detail below.

1. *Initialisation*: The KGC executes this polynomial-time algorithm to generate a set of public parameters $\{M_{pb}, P, q, G, h_1, h_2, h_3, h_4, h_5\}$ and a master secret key (P_{ms}). Here, M_{pb} represents the public key of the KGC, P is the generator of a subgroup G of an elliptic curve, (q, p) are random prime numbers, and h_1, h_2, h_3, h_4, h_5 denote five cryptographic hash functions. The master secret key P_{ms} is chosen by the KGC as $P_{ms} \in \mathbb{Z}_p^*$, and M_{pb} is computed as $M_{pb} = P_{ms} \cdot P$.
2. *AID-Gen*: Upon receiving the actual identity of a drone (O_{id}), the KGC selects a random number $b_d \in \mathbb{Z}_p^*$, computes $B_d = b_d \cdot P$, and generates the anonymous identity A_{id} as $A_{id} = O_{id} \oplus h_5(b_d \cdot M_{pb}, M_{pb}, T_1)$, where T_1 is a time limit. The KGC stores (B_d, A_{id}) in its memory and transmits this tuple to the drone.
3. *Key-Gen*: Upon receiving the tuple $(O_{id}, S_d = \varrho_d \cdot P)$ through a secure channel, where $\varrho_d \in \mathbb{Z}_p^*$, the KGC selects $\varkappa_d$, computes $X_d = \varkappa_d \cdot P$, and calculates: $D_d = \varkappa_d + P_{ms} \cdot q_d$, where $q_d = h_1(O_{id}, S_d)$. The KGC then sends (D_d, X_d) to the drone. The drone sets (S_d, X_d) as its public key pair and $(D_d, \varkappa_d)$ as its private key pair. The key generation process for the GS is analogous: the GS sets $(S_{gs} = \varrho_{gs} \cdot P, X_{gs} = \varkappa_{gs} \cdot P)$ as its public key and $(D_{gs}, \varkappa_{gs})$ as its private key, where, $D_{gs} = \varkappa_{gs} + P_{ms} \cdot q_{gs}$, and $q_{gs} = h_1(O_{gs}, S_{gs})$.
4. *Data-Aggregate*: The drone selects $\sigma_i \in \mathbb{Z}_p^*$, computes $Q_i = \sigma_i \cdot P$, and calculates: $T_i = \varrho_d \cdot X_{gs}$, $H_{i2} = h_2(Q_i, T_i)$, $H_{i3} = h_3(X_{gs}, X_d, Q_i, T_i, O_{id}, O_{gs})$, $H_{i4} = h_4(X_{gs}, X_d, Q_i, T_i, O_{id}, O_{gs}, H_{i3})$. The drone

then computes: $j_i = (\sigma_i \cdot H_{i3} + D_d \cdot H_{i4}) \mod q$, $J_i = \upsilon_i \cdot P, C_i = (j_i, m_i) \oplus H_{i2}$, and sends $\phi_i = (Q_i, J_i, C_i)$ to the Mobile Edge Computing (MEC) server. Upon receiving data from multiple regions, the MEC server computes the aggregate signature as: $J = \sum_{i=0}^{n} J_i$, and transmits

$$\phi_{\text{agg}} = ((Q_1, Q_2, \ldots, Q_n), J, (C_1, C_2, \ldots, C_n)) \tag{1}$$

to the GS.

5. *Verify-Decryption*:For $1 \leq i \leq n$, the GS calculates: $T_i = S_d \cdot x_{gs}$, recovers $(j_i, m_i) = C_i \oplus h_2(Q_i, T_i)$, and verifies the aggregate signature by checking if: $J = \sum_{i=0}^{n} H_{i3} \cdot Q_i + \sum_{i=0}^{n} H_{i4} \cdot X_d + \left(\sum_{i=0}^{n} H_{i4} \cdot q_d\right) \cdot M_{pb}$. If the equation holds, the GS accepts the aggregate signature.

2.2. Security Flaws of Verma et al.'s Scheme

This subsection demonstrates that the scheme presented in [1] is vulnerable to the following security flaws.

2.2.1. *Impersonation Attack Vulnerability*

Proposition 1. *The identity-based key generation scheme proposed in [1] is susceptible to an impersonation attack by an outsider adversary due to the lack of identity verification and stateful key management at the key generation center (KGC).*

Proof. Let E_{ve} be an eavesdropper attempting to impersonate a legitimate drone. The adversary selects a fake identity $O_{E_{id_i}}$ and chooses a random secret $\varrho_{E_{id_i}} \in \mathbb{Z}_p^*$. He then computes:

$$S_{E_{id_i}} = \varrho_{E_{id_i}} \cdot P, \tag{2}$$

and sends the tuple $(O_{E_{id_i}}, S_{E_{id_i}})$ to the KGC. Upon receiving this tuple, the KGC, without validating the identity, randomly selects $\varkappa_{E_{id_i}} \in \mathbb{Z}_p^*$ and computes:

$$X_{E_{id_i}} = \varkappa_{E_{id_i}} \cdot P \tag{3}$$

$$q_{E_{id_i}} = h_1(O_{E_{id_i}}, S_{E_{id_i}}) \tag{4}$$

$$D_{E_{id_i}} = \varkappa_{E_{id_i}} + P_{ms} \cdot q_{E_{id_i}} \tag{5}$$

The KGC then transmits $(X_{E_{id_i}}, D_{E_{id_i}})$ to the adversary, who now holds a valid key pair. To forge a signature, E_{ve} chooses $\sigma_{E_{id_i}} \in \mathbb{Z}_p^*$ and computes:

$$Q_{E_{id_i}} = \sigma_{E_{id_i}} \cdot P, \tag{6}$$

$$T_{E_{id_i}} = \varrho_{E_{id_i}} \cdot X_{gs}, \tag{7}$$

$$H_{i2} = h_2(Q_{E_{id_i}}, T_{E_{id_i}}), \tag{8}$$

$$H_{i3} = h_3(X_{gs}, X_{E_{id_i}}, Q_{E_{id_i}}, T_{E_{id_i}}, O_{E_{id_i}}, O_{gs}), \tag{9}$$

$$H_{i4} = h_4(X_{gs}, X_{E_{id_i}}, Q_{E_{id_i}}, T_{E_{id_i}}, O_{E_{id_i}}, O_{gs}, H_{i3}), \tag{10}$$

The adversary then constructs the signature component:

$$j_{E_{id_i}} = (\sigma_{E_{id_i}} \cdot H_{i3} + D_{E_{id_i}} \cdot H_{i4}) \mod q, \tag{11}$$

$$J_{E_{id_i}} = j_{E_{id_i}} \cdot P, \tag{12}$$

$$C_{E_{id_i}} = (j_{E_{id_i}}, m_i) \oplus H_{i2}. \tag{13}$$

It then sends the forged tuple $\phi_i = (Q_{E_{id_i}}, J_{E_{id_i}}, C_{E_{id_i}})$ to the MEC server. The MEC aggregates signatures from all regions, that is $J = \sum_{i=0}^{n} J_{E_{id_i}}$ and forwards:

$$\phi_{\text{agg}} = \left((Q_{E_{id_i}1}, \ldots, Q_{E_{id_i}n}), J, (C_{E_{id_i}1}, \ldots, C_{E_{id_i}n})\right), \tag{14}$$

to the GS. At the GS, the signature verification procedure involves checking whether:

$$J = \sum_{i=0}^{n} H_{i3} \cdot Q_{E_{id_i}} + \sum_{i=0}^{n} H_{i4} \cdot X_{E_{id_i}} + \left(\sum_{i=0}^{n} H_{i4} \cdot q_{E_{id_i}}\right) \cdot M_{pb} \tag{15}$$

Substituting Equations (3) to (5), and simplifying:

$$\begin{aligned} J &= \sum_{i=0}^{n} H_{i3} \cdot Q_{E_{id_i}} + \sum_{i=0}^{n} H_{i4} \cdot \varkappa_{E_{id_i}} \cdot P + \sum_{i=0}^{n} H_{i4} \cdot q_{E_{id_i}} \cdot P_{ms} \cdot P \\ &= \sum_{i=0}^{n} \left(H_{i3} \cdot \sigma_{E_{id_i}} \cdot P + H_{i4} \cdot D_{E_{id_i}} \cdot P \right) \\ &= \sum_{i=0}^{n} \left(H_{i3} \cdot \sigma_{E_{id_i}} + H_{i4} \cdot D_{E_{id_i}} \right) \cdot P \\ &= \sum_{i=0}^{n} j_{E_{id_i}} \cdot P = \sum_{i=0}^{n} J_{E_{id_i}} = J \end{aligned} \tag{16}$$

Since the verification equation holds, the GS accepts the forged signature as valid.

This attack succeeds due to the following weaknesses in the scheme: 1) the KGC does not authenticate the identity $O_{E_{id_i}}$. 2) There is no stateful record or revocation mechanism for previously issued key pairs. 3) The public key construction allows valid outputs even from adversarial inputs. Therefore, based on the above analysis, the scheme in [1] is proven to be vulnerable to an impersonation attack by an outsider adversary. □

2.2.2. *Forgery Attack Vulnerability*

Proposition 2. *The identity-based authentication scheme proposed in [1] is vulnerable to a forgery attack by an outsider adversary. Given access to previously obtained key material, an adversary can generate a valid signature without interacting with the sender or possessing a legitimate message.*

Proof. Assume that the adversary E_{ve} has already acquired a valid key pair $(X_{E_{id_i}}, D_{E_{id_i}})$ from the KGC as shown in the Impersonation Attack (Proposition 1). The adversary selects $\sigma_{E_{id_i}} \in \mathbb{Z}_p^*$ and computes:

$$Q_{E_{id_i}} = \sigma_{E_{id_i}} \cdot P. \tag{17}$$

Given a target GS identity O_{gs} and fixed values for X_{gs} and $T_{E_{id_i}}$, the adversary computes the hash functions values of

H_{i3} and H_{i4} provided in Equations (9) and (10), respectively. Using these, the adversary constructs a forged signature as:

$$j_{E_{id_i}} = (\sigma_{E_{id_i}} \cdot H_{i3} + D_{E_{id_i}} \cdot H_{i4}) \mod q, \tag{18}$$

$$J_{E_{id_i}} = j_{E_{id_i}} \cdot P. \tag{19}$$

Then, E_{ve} sends the forged message tuple $\phi_i = (Q_{E_{id_i}}, J_{E_{id_i}})$ to the MEC server. Upon receiving data from multiple regions, the MEC aggregates all forged signatures as $J = \sum_{i=0}^{n} J_{E_{id_i}}$, and forwards the aggregated message:

$$\phi_{\text{agg}} = \left((Q_{E_{id_i}1}, Q_{E_{id_i}2}, \ldots, Q_{E_{id_i}n}), J \right), \tag{20}$$

to the GS. The GS verifies the aggregate signature by checking whether the following equation holds:

$$J = \sum_{i=0}^{n} H_{i3} \cdot Q_{E_{id_i}} + \sum_{i=0}^{n} H_{i4} \cdot X_{E_{id_i}} + \left(\sum_{i=0}^{n} H_{i4} \cdot q_{E_{id_i}} \right) \cdot M_{pb} \tag{21}$$

Now, since we know the $X_{E_{id_i}}$ and $D_{E_{id_i}}$ of the scheme, the final proof follows similar steps as mentioned in Section 2.2.1, i.e., substituting equations (3) and (5) into the verification equation (21) which then results in

$$\begin{aligned} J &= \sum_{i=0}^{n} H_{i3} \cdot Q_{E_{id_i}} + \sum_{i=0}^{n} H_{i4} \cdot \varkappa_{E_{id_i}} \cdot P + \sum_{i=0}^{n} H_{i4} \cdot q_{E_{id_i}} \cdot P_{ms} \cdot P \\ &= \sum_{i=0}^{n} \left(H_{i3} \cdot Q_{E_{id_i}} + H_{i4} \cdot (\varkappa_{E_{id_i}} + q_{E_{id_i}} \cdot P_{ms}) \cdot P \right) \\ &= \sum_{i=0}^{n} \left(H_{i3} \cdot \sigma_{E_{id_i}} \cdot P + H_{i4} \cdot D_{E_{id_i}} \cdot P \right) \\ &= \sum_{i=0}^{n} (H_{i3} \cdot \sigma_{E_{id_i}} + H_{i4} \cdot D_{E_{id_i}}) \cdot P \\ &= \sum_{i=0}^{n} j_{E_{id_i}} \cdot P = \sum_{i=0}^{n} J_{E_{id_i}} = J \end{aligned} \tag{22}$$

Since the forged aggregate signature satisfies the GS verification equation, the adversary succeeds in forging a valid signature without interaction or authorisation. This is possible due to the following design flaws: 1) Lack of message binding, i.e., the signature does not include or verify a bound message. 2) Deterministic and stateless behaviour of the KGC, enabling repeated generation of valid key pairs for arbitrary identities. 3) Public values (e.g., P, X_{gs}) are sufficient for computing valid verification responses. Thus, the scheme in [1] is proven to be insecure against forgery attacks by an outsider eavesdropper. □

2.2.3. Incorrectness in Signature Verification

Proposition 3. *The signature verification process in the scheme proposed by [1] is incorrect due to the dependency on the GS's identity O_{id} within the hash functions, which is not communicated during the signing process. As a result, the values of H_{i3} and H_{i4} differ at the sender and receiver sides, leading to failed verification.*

Proof. Assume a legitimate drone selects $\sigma_i \in \{1, 2, \ldots, n\}$ and computes the signature as follows. First, the drone computes the hash values:

$$H_{i3} = h_3(X_{gs}, X_d, Q_i, T_i, O_{id}, O_{gs}) \tag{23}$$

$$H_{i4} = h_4(X_{gs}, X_d, Q_i, T_i, O_{id}, O_{gs}, H_{i3}) \tag{24}$$

Then, the drone generates the signature using:

$$j_i = (\sigma_i \cdot H_{i3} + D_d \cdot H_{i4}) \mod n \tag{25}$$

Observe that both hash functions H_{i3} and H_{i4} (Equations (23) and (24)) include the identity O_{id} of the GS as an input. However, the scheme does not define any communication or transmission step through which the drone shares O_{id} with the GS. Therefore, the GS, during signature verification, computes the same hash values independently using its own identity O_{id}:

$$H'_{i3} = h_3(X_{gs}, X_d, Q_i, T_i, O_{id}, O_{gs}) \tag{26}$$

$$H'_{i4} = h_4(X_{gs}, X_d, Q_i, T_i, O_{id}, O_{gs}, H'_{i3}) \tag{27}$$

While the notations H'_{i3} and H'_{i4} match the structure of H_{i3} and H_{i4}, they are computed **without knowledge of the specific O_{id}** used by the drone during signing, since O_{id} was never explicitly transmitted or negotiated between the parties. Consequently, we have:

$$H'_{i3} \neq H_{i3}, \quad H'_{i4} \neq H_{i4} \tag{28}$$

This mismatch implies that the verification side reconstructs a different signature input:

$$j'_i = (\sigma_i \cdot H'_{i3} + D_d \cdot H'_{i4}) \mod n \tag{29}$$

Thus, $j'_i \neq j_i$, and the verification equation will fail, as:

$$j'_i \cdot P \neq j_i \cdot P \tag{30}$$

Therefore, even when the signature was computed correctly by the drone, the GS cannot validate it due to inconsistent inputs in the hash computations. This leads to a fundamental correctness flaw in the scheme. Specifically, the design requires both parties (signer and verifier) to include O_{id} in the hash inputs without a secure mechanism to ensure mutual agreement or synchronisation on this value. □

Building on this cryptanalysis and the identified weaknesses in Verma et al.'s [1] construction, the next section introduces the preliminaries and the IoD system model that underpin our improved aggregation-based signcryption scheme.

3. Preliminaries

This section provides the mathematical preliminaries, the system/network model, and the threat model that form the basis for the design and security analysis of the proposed scheme.

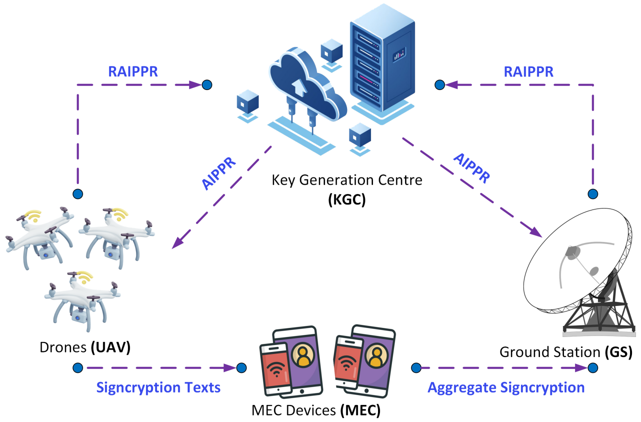

Figure 1: The system architecture of the proposed aggregation-based signcryption for a secure drone-to-ground station communication system.

3.1. System Model/Network Model

The network model for our improved scheme is illustrated in Fig. 1, which contains four entities, i.e., the *KGC*, *Drone Devices*, *MEC Devices*, and *GS* that are explained in the following steps [1].

1. *The KGC*: The main purpose of KGC is to set the public parameter, anonymous identity, and its public and private keys and produce the partial private key for the users (Drone and GS). Then, KGC generates and sends the partial private key in an encrypted form (AIPPR) to the users (Drone and GS) and stores it along with the user's identity in its memory for future use.
2. *Drone Devices*: Drone devices are responsible for generating their public and private keys when they receive AIPPR from KGC. Further, these devices generate Signcryption for the generated data and dispatch it to MEC devices over an open network.
3. *MEC Devices*: Upon reception of Signcryption texts from many Drone devices, MEC Devices are responsible for generating a single aggregate Signcryption and dispatch it to GS by using an open network. The MEC server is assumed to be trusted in the execution of its designated role. It honestly receives individual signcrypted messages from drones, performs aggregation without alteration, and forwards the aggregated signcryption to the ground station. It does not collude with any adversary nor attempt to learn the private content of the messages beyond what is necessary for aggregation.
4. *The GS*: GS is the first send request (RAIPPR) to KGC for an anonymous identity and partial private key. When it received a partial private key and an anonymous identity (AIPPR) from the KGC, then GS generated his public and private key. Further, upon reception of aggregate Signcryption from MEC devices, GS verify the aggregate signature and decrypts the ciphertexts.

3.2. Threat/Adversary Model

We evaluate our improved scheme against Confidentiality under adaptive chosen-ciphertext attack (IND-CCA2) and Unforgeability under adaptive chosen-message attack (EUF-CMA) by using the Random Oracle Model (ROM). We consider two types (Type 1 and Type 2) of adversary for our improved scheme that can be represented through symbols A_1 and A_2. Type-1 adversary (A_1) has the capability to replace the public keys of arbitrary participants and is restricted to accessing the master private key. Type-2 adversary (A_2) has no capability to replace public keys of arbitrary participants and can access the master private key. Both adversaries (A_1, A_2) can adaptively query the challenger (C_1, C_2) with the following:

- *Hash queries*: The adversaries (A_1, A_2) ask hash queries $(Q_(h_1), Q_(h_2), Q_(h_3), Q_(h_4), Q_(h_5))$ and challenger (C_1, C_2) respond accordingly.
- *Public/Private key queries*: The adversaries (A_1, A_2) ask Public Key queries $(Q_P BK)$ and challenger (C_1, C_2) respond accordingly.
- A_1 can ask Replaced Public Key queries $(Q_R PBK)$ and challenger (C_1) respond accordingly. The adversaries (A_1, A_2) ask Partial Private Key queries $(Q_P PK)$ and challenger (C_1, C_2) respond accordingly. Note that it is disallowed for the challenge identity.
- The adversaries (A_1, A_2) ask Full Private Key Queries $(Q_F PK)$ and challenger (C_1, C_2) respond accordingly. Note that it is disallowed for the challenge identity.
- The adversaries (A_1, A_2) ask Anonymous Identity Queries $(Q_A I)$ and challenger (C_1, C_2) respond accordingly.
- The adversaries (A_1, A_2) ask Signcryption Queries $(Q_S C)$ and Unsigncryption Queries $(Q_U SC)$, and challenger (C_1, C_2) respond accordingly.
- *Restrictions*: In confidentiality under IND-CCA2 (Corresponds to Theorem 1 and 2), the adversary (A_1, A_2) is not allowed to ask $Q_U SC$ on the challenge ciphertext. In EUF-CMA (Corresponds to Theorem 3 and 4), the adversary (A_1, A_2) must output a fresh forgery not returned by the $Q_S C$ queries.

IND-CCA2 Confidentiality (A_1) : This section corresponds to Theorem 1, and the following are the explanations:

- *Initialization*: C_1 published public parameters param.
- *Query Phase 1*: A_1 perform $Q_{PBK}, Q_{RPBK}, Q_{AI}, Q_{h_1}, Q_{h_2}, Q_{h_3}, Q_{h_4}, Q_{h_5}, Q_{PPK}, Q_{FPK}, Q_{USC}$ and Q_{SC}.
- *Challenge*: Here A_1 send two equal length plaintexts (m_1, m_2) and the target sender identity (${O_{id}}^*$). C_1 choose $\pi \in \{1, 2\}$, set $C^* = \left(j^*, m_\pi\right) \oplus {H_{i2}}^*$, and return $f^* = (Q^*, J^*, C^*)$ to A_1.

- *Query Phase 2*: Same as Phase 1, except: Q_{USC} on f^* are forbidden and the queries $\left(Q_{PPK}, Q_{FPK}\right)$ on O^* are forbidden if the public key was replaced.
- *Guess*: A_1 outputs π'. It succeeds if $\pi' = \pi$.

IND-CCA2 Confidentiality (A_2) : This section corresponds to Theorem 1 and the following are the explanations:

- *Initialization*: C_2 publishes the public parameters param and gives the master secret key to $\mathcal{A}_2$.
- *Query Phase 1*: $\mathcal{A}_2$ performs the queries in the same way as in IND-CCA2 confidentiality (by $\mathcal{A}_1$), except for Q_{RPBK}.
- *Challenge*: $\mathcal{A}_2$ sends two equal-length plaintexts (m_1, m_2) and the target sender identity $\mathcal{O}^*_{id}$. C_1 chooses $\pi \in \{1, 2\}$, sets $C^* = (j^*, m_\pi) \oplus H^*_{i2}$ and returns $f^* = (Q^*, J^*, C^*)$ to $\mathcal{A}_2$.
- *Query Phase 2*: Same as Phase 1, except: Q_{USC} on f^* and the other queries $(Q_{\mathsf{PPK}}, Q_{\mathsf{FPK}})$ on $\mathcal{O}^*$ are forbidden.
- *Guess*: A_2 outputs π'. It succeeds if $\pi' = \pi$.

EUF-CMA Unforgeability (A_1) : This section corresponds to Theorem 3, and the following are the explanations.

- *Initialization*: C_1 publishes the public parameters param.
- *Query Phase 1*: $\mathcal{A}_1$ performs the queries in the same way as in IND-CCA2 confidentiality (by $\mathcal{A}_1$).
- *Forgery*: $\mathcal{A}_1$ outputs a forged signcryption triple $f^* = (Q^*, J^*, C^*)$ on $(\mathcal{O}^*_{id}, m^*)$.
- *Winning Condition*: f^* verifies correctly under $\mathcal{O}^*_{id}$ and was not previously generated by the Q_{SC} oracle.

EUF-CMA Unforgeability (A_2) : This section corresponds to Theorem 4, and the following are the explanations:

- *Initialization*: C_2 publishes the public parameters param and gives the master secret key to $\mathcal{A}_2$.
- *Query Phase 1*: $\mathcal{A}_2$ performs the queries in the same way as in IND-CCA2 confidentiality (by $\mathcal{A}_2$).
- *Forgery*: $\mathcal{A}_2$ outputs a forged signcryption triple $f^* = (Q^*, J^*, C^*)$ on $(\mathcal{O}^*_{id}, m^*)$.
- *Winning Condition*: f^* verifies correctly under $\mathcal{O}^*_{id}$ and was not previously generated by the Q_{SC} oracle.

4. Construction of the Improved Scheme

The improved scheme comprises five phases: *Initialization*, *AID-Gen*, *Key-Gen*, *Data-Aggregate*, and *Verify-Decryption*. Table 1 contains the symbols used in this paper. So, each phase is described as follows:

1. *Initialization:* KGC executes this polynomial time algorithm, in which it produces a set of public parameters $\left\{M_{pb}, D, n, h_1, h_2, h_3, h_4, h_5\right\}$ and master secret key $\left(P_{ms}\right)$, where $M_{pb}, D, n, h_1, h_2, h_3, h_4, h_5$, denotes the public key of KGC, devisor of hyperelliptic curve, a random prime number with have a size 80 bits, and five hash functions. Note that P_{ms} is chosen by KGC as $P_{ms} \dot{\in} {Z_n}^*$ and $M_{pb} = P_{ms}.D$.
2. *AID-Gen:* Upon receiving the actual identity of Drone $\left(O_{id}\right)$, KGC select a random number $\mathfrak{b}_d \dot{\in} \{1, 2, 3, \dots, n\}$, compute $B_d = \mathfrak{b}_d.D$, and $A_{id} = O_{id} \oplus h_5\left(\mathfrak{b}_d.M_{pb}, M_{pb}, T_1\right)$, where T_1 is the time limit. Then, it stores $\left(B_d, A_{id}, O_{id}\right)$ in its memory and sends it to the Drone.
3. *Key-Gen*: The user can pick his identity $\left(O_{id}\right)$, select a random number $\varsigma_d \dot{\in} \{1, 2, 3, \dots, n\}$, compute $S_d = \varsigma_d \cdot D, K = \varsigma_d \cdot M_{pb}$, encrypt the tuple $\left(O_{id}, S_d\right)$ as $E_{id} = \left(O_{id}, S_d\right) \oplus K$, and send E_{id} to KGC using insecure network. Upon receiving E_{id}, KGC compute $K = S_d.P_{ms}$, recover the tuple (O_{id}, S_d) as $(O_{id}, S_d) = E_{id} \oplus K$, select $\varkappa_d \dot{\in} \{1, 2, 3, \dots, n\}$, compute $X_d = \varkappa_d.D$, and $L_d = \left(\varkappa_d + P_{ms}.q_d\right)$, where $q_d = h_1\left(O_{id}, S_d\right)$. By using an insecure network, KGC dispatched $E_{ppt} = \left(L_d, X_d\right) \oplus K$ to Drone and stored (L_d, X_d, O_{id}) in his memory for future use. Then, Drone recover $\left(L_d, X_d\right) = E_{ppt} \oplus K$ and set (S_d, X_d) and $(L_d, \varkappa_d)$ as his public and private key pairs. The key generation process is same as above for GS: it set $(S_{gs} = \varsigma_{gs}.D, X_{gs} = \varkappa_{gs} \cdot D$) as his public key and his private key as $\left(L_{gs}, \varkappa_{gs}\right)$, where $L_{gs} = \left(\varkappa_{gs} + P_{ms}.q_{gs}\right)$ and $q_{gs} = h_1\left(O_{gs}, S_{gs}\right)$.
4. *Data-Aggregate:* Drone select $\sigma_i \dot{\in} \{1, 2, 3, \dots, n\}$, compute $Q_i = \sigma_i.D, T_i = \varsigma_d.X_{gs}, H_{i2} = h_2\left(Q_i, T_i\right)$, $H_{i3} = h_3\left(X_{gs}, X_d, Q_i, T_i, O_{gs}\right), H_{i4} = h_4(X_{gs}, X_d, Q_i, T_i, O_g, H_{i3})$. Then, Drone computes $j_i = (\sigma_i H_{i3} + L_d H_{i4}) \bmod n, J_i = \upsilon_i.D, C_i = \left(j_i, m_i, t_1\right) \oplus H_{i2}$, and send $f_i = \left(Q_i, J_i, C_i\right)$ to MEC. Upon receiving data from each region, MEC compute aggregate signature as $J = \sum_{i=0}^{n} j_i$ and send $\mathrm{f}_{\mathrm{agg}} = \left(\left(Q_1, Q_2, Q_3, \dots, Q_n\right), J, \left(C_1, C_2, C_3, \dots .C_n\right)\right)$ to GS.
5. *Verify-Decryption:* Drone select $\sigma_i \dot{\in} \{1, 2, 3, \dots, n\}$, compute $Q_i = \sigma_i.D, T_i = \varsigma_d.X_{gs}, H_{i2} = h_2\left(Q_i, T_i\right)$, $H_{i3} = h_3\left(X_{gs}, X_d, Q_i, T_i, O_{gs}\right), H_{i4} = h_4(X_{gs}, X_d, Q_i, T_i, O_g, H_{i3})$. Then, Drone computes $j_i = (\sigma_i H_{i3} + L_d H_{i4}) \bmod n, J_i = \upsilon_i.D, \left(j_i, m_i, t_1\right) = C_i \oplus H_{i2}$.
6. *New-Drone Joining:*The new Drone submit his identity $\left(NO_{id}\right)$, KGC select $\mathfrak{b}_N \dot{\in} \{1, 2, \dots, n\}$ (which is a random number), compute $B_N = \mathfrak{b}_N.D$, and $A_{Nid} = NO_{id} \oplus h_5\left(\mathfrak{b}_N.M_{pb}, M_{pb}, T_1\right)$, where T_1

Table 1
Notations Used in this Paper

No	Symbol	Description
1	M_{pb}	This symbol represents the master public key generated by KGC
2	D	It is used for representing a divisor on a hyper elliptic curve
3	h_1, h_2, h_3, h_4, h_5	These symbols represent one-way hash functions, which are collision-resistant
4	P_{ms}	This symbol represents the master private key generated by KGC
5	A_{id}	This symbol represents the encrypted identity of users
6	O_{id}	This symbol represents the identity of users
7	T_1	This symbol represents the fresh time stamp
8	$(L_d, \mathcal{H}_d)$	The private key pair for Drone
9	(S_d, X_d)	The public key pair for Drone
10	$(L_{gs}, \varkappa_g)$	The private key pair for GS
11	(S_{gs}, X_{gs})	The public key pair for GS
12	$\oplus$	This is XOR operation used for encryption and decryption
13	J	This symbol is used for aggregate signature
14	j_i	This symbol is used for single signature
15	C_i	This symbol is used for encrypted text

is the time limit. Then, it stores $\left(B_N, A_{Nid}, NO_{id}\right)$ in its memory and sends it to the Drone. For more explanation and clarity the new Drone can pick his identity $\left(NO_{id}\right)$, select a random number $\varsigma_N \dot{\in} \{1, 2, 3, \ldots, n\}$, compute $S_N = \varsigma_N \cdot D, K = \varsigma_N \cdot M_{pb}$, encrypt the tuple $\left(NO_{id}, S_N\right)$ as $E_{iN} = \left(NO_{id}, S_N\right) \oplus K$, and send E_{iN} to KGC using insecure network. Upon receiving E_{iN}, KGC compute $K = S_N.P_{ms}$, recover the tuple (NO_{id}, S_N) as $(NO_{id}, S_N) = E_{iN} \oplus K$, select $\varkappa_N \dot{\in} \{1, 2, \ldots, n\}$, compute $X_N = \varkappa_N.D$, and $L_N = \left(\varkappa_N + P_{ms}.q_N\right)$, where $q_N = h_1\left(NO_{id}, S_N\right)$. By using an insecure network, KGC dispatched $E_{ppt} = \left(L_N, X_N\right) \oplus K$ to the new Drone and stored (L_N, X_N, NO_{id}) in his memory for future use. Then, new Drone recover $\left(L_N, X_N\right) = E_{ppt} \oplus K$ and set (S_N, X_N) and $(L_N, \varkappa_N)$ as his public and private key pairs.

7. *Correctness:* GS calculate $T_i = S_d \cdot x_{gs} = S_d \cdot x_{gs} = \varsigma_d \cdot D \cdot x_{gs} = \varsigma_d \cdot x_{gs} \cdot D = \varsigma_d \cdot X_{gs} = T_i$ hence proved. Further, GS check if $J = \sum_{i=0}^{n} H_{i3}Q_i + \sum_{i=0}^{n} H_{i4}X_d + \left(\sum_{i=0}^{n} H_{i4}q_d\right) M_{pb}$ holds, then accept the aggregate signature: $J = \sum_{i=0}^{n} H_{i3}Q_i + \sum_{i=0}^{n} H_{i4}X_d + \left(\sum_{i=0}^{n} H_{i4}q_d\right) M_{pb} = \sum_{i=0}^{n} H_{i3}Q_i + \sum_{i=0}^{n} H_{i4}\varkappa_d.D + \left(\sum_{i=0}^{n} H_{i4}q_d\right) M_{pb} = \sum_{i=0}^{n} H_{i3}Q_i + \sum_{i=0}^{n} H_{i4}\varkappa_d.D + \sum_{i=0}^{n} H_{i4}q_d.P_{ms}.D = \sum_{i=0}^{n} H_{i3}Q_i + \sum_{i=0}^{n} \left(H_{i4}\varkappa_d + H_{i4} \cdot q_d \cdot P_{ms}\right) \cdot D = \sum_{i=0}^{n} H_{i3}Q_i + H_{i4}(\varkappa_d + q_d \cdot P_{ms}) \cdot D = \sum_{i=0}^{n} H_{i3}Q_i + \left(H_{i4}L_d\right) \cdot D = \sum_{i=0}^{n} H_{i3}\sigma_i \cdot D + \left(H_{i4}L_d\right) \cdot D = \sum_{i=0}^{n} \left(H_{i3}\sigma_i + H_{i4}L_d\right) \cdot D = \sum_{i=0}^{n} j_i = J$ hence proved.

5. Security Analysis

This section provides formal justification that the proposed scheme satisfies essential security requirements, including confidentiality, authentication, integrity, unforgeability, and resistance to impersonation attacks.

Theorem 1. *If a Type I adversary A_1 possesses an unignorable advantage ξ_{A_1} in breaching the IND-CCA2 confidentiality of our improved scheme in the Random Oracle Model, then there exists a challenger C_1 that can solve the Hyper Elliptic Curve Computational Diffie-Hellman problem (HECDH) with the following probability:*

$$Adv_{C_1}^{HECDH} \geq \left(1 - \frac{2Q_{PK}}{2^k}\right)^2 \cdot \left(\frac{\xi_{A_1}}{\varepsilon(Q_{SC} + 1)}\right),$$

where Q_{PK}, Q_{SC}, k, and ε represent private key queries, signcryption queries, the bit length of the group order, and Euler's number (natural logarithm base).

Proof. C_1 will interact with A_1 to solve the HECDH problem $(D, aD, bD) \rightarrow abD$ for our proposed scheme by following a series of steps.

Initialization: C_1 sets $P_{ms} = b \in \mathbb{Z}_n^*$ (unknown to C_1) and $M_{pb} = bD$, then publishes public parameters $param = \{M_{pb}, D, n, h_1, h_2, h_3, h_4, h_5\}$.

Public Key Queries (Q_{PBK}): If A_1 asks for a public key, C_1 checks O_{id} in L_{isK} and returns the stored tuple (S_d, X_d). Otherwise, it chooses $\varsigma_d, \varkappa_d \in \{1, 2, \ldots, n\}$, sets $S_d = \varsigma_d D$, $X_d = \varkappa_d D$, and returns (S_d, X_d).

Replaced Public Key Queries (Q_{RPBK}): If A_1 asks for a replaced public key, C_1 replaces (S_d, X_d) with $(S_d{}', X_d{}')$ and stores $(S_d{}', X_d{}', \perp, \perp)$ in L_{isK}.

Anonymous Identity Queries (Q_{AI}): If A_1 asks for an anonymous identity, C_1 checks O_{id} in L_{isAI}. If unseen, it selects $\mathfrak{b}_d \in \{1, 2, \ldots, n\}$, sets $B_d = \mathfrak{b}_d D$, and looks up $y = h_5(\mathfrak{b}_d \cdot M_{pb}, M_{pb}, T_1)$ in L_{ish5}. Then, C_1 sets $A_{id} = O_{id} \oplus y$, stores $(b_d, B_d, A_{id}, T_1, O_{id})$ in L_{ish5}, and returns (B_d, A_{id}).

Hash Queries:

- h_1 Queries Q_{h_1}: A_1 asks this query, C_1 checks q_i in L_{ish1}, if unseen, then chooses q_i randomly, stores q_i in L_{ish1} and return.

- h_2 Queries Q_{h_2}: A_1 asks this query, C_1 checks H_{i2} in L_{ish2}, if unseen, then chooses H_{i2} randomly, stores H_{i2} in L_{ish2} and return.

- h_3 Queries Q_{h_3}: A_1 asks this query, C_1 checks H_{i3} in L_{ish3}, if unseen, then chooses H_{i3} randomly, store H_{i3} in L_{ish3} and return.

- h_4 Queries Q_{h_4}: A_1 asks this query, C_1 checks H_{i4} in L_{ish4}, if unseen, then chooses H_{i4} randomly, stores H_{i4} in L_{ish4} and return.

- h_5 Queries Q_{h_5}: A_1 asks this query, C_1 checks y in L_{ishy}, if unseen, then chooses y randomly, stores y in L_{ishy} and return.

Partial Private Key Queries (Q_{PPK}): A_1 asks for partial private key, C_1 checks if $O_{id} \neq O^*_{id}$, call $Q_A I$, and return (B_d, A_{id}).

Full Private Key Queries (Q_{FPK}): $A_1 4$ asks for full private key, C_1 check if $O_{id} = O^*_{id}$ and abort. Otherwise, check for q_i in L_{ish1}, if $q_i \neq 0$, then set $q_i = 0$, proceeds, and return $(D_d, \varkappa_d)$.

Signcryption Queries (Q_{SC}): In this query, C_1 checks (S_d, X_d) in L_{isK}, if unseen, then it creates or uses the replaced public key. Also, it ensures the GS public key (S_{gs}, X_{gs}) and the secret x_{gs}; note that if GS is honest, then C_1 knows x_{gs}. C calculates $T = \varsigma_d \cdot X_{gs}$ and returns $q_i, H_{i2}, H_{i3}, H_{i4}, y$ from $L_{isqi}, L_{isH_{i2}}, L_{isH_{i3}}, L_{isH_{i4}}$, and L_{isy}. Defines $J = H_{i3}Q + H_{i4}X_d + H_{i4}q_i M_{pb}$, $C_i = (j_i, m_i) \oplus H_{i2}$, where $H_{i2} = h_2(Q, T)$, and sends $f = (Q, J, C)$ to A_1.

Unsigncryption Queries (Q_{USC}): In this query, C_1 checks the sender public key (S_d, X_d) in L_{isK}. Also, it ensures the GS public key (S_{gs}, X_{gs}) and the secret x_{gs}; note that if GS is honest, then C_1 knows $x_g s$. C_1 calculated $T = S_d.x_{gs}$ and queries H_{i2} on the exact pair (Q, T). Then, parse $(j_i, m_i) = C_i \oplus H_{i2}$, returns $q_i, H_{i2}, H_{i3}, H_{i4}$ from $L_{isqi}, L_{isH_{i2}}, L_{isH_{i3}}$, and $L_{isH_{i4}}$. It checks, if $J = \sum_{i=0}^{n} H_{i3}Q + \sum_{i=0}^{n} H_{i4}X_d + \left(\sum_{i=0}^{n} H_{i4}q_d M_{pb}\right)$, holds, then returns m, otherwise returns with error.

Challenge: A_1 sends two equal length plaintexts (m_1, m_2) and the target sender identity O^*_{id}. C_1 chooses $\pi \in \{1, 2\}$, compute $Q^* = a \cdot D, T^* = S^*_d x_{gs}$, returns $q^*_i, H^*_{i2}, H^*_{i3}, H^*_{i4}$ from $L_{isqi}, L_{isH_{i2}}, L_{isH_{i3}}$, and $L_{isH_{i4}}$. C_1 then defines $J^* = H^*_{i3}Q + H^*_{i4}X^*_d + H^*_{i4}q^*_i M_{pb}$, chooses any j^*, sets $C^* = (j^*, m_\pi) \oplus H^*_{i2}$, and return $f^* = (Q^*, J^*, C^*)$ to A_1.

Phase 2: In this phase, A_1 perform all the queries as asked in Phase 1 and restricted to ask full private key on O^*_{id} and no decryption on the exact $f^* = (Q^*, J^*, C^*)$ for sender (O^*_{id}) and receiver (O_{gs}).

Guess: A_1 outputs π'. suppose $\pi' = \pi$ with benefit ξ_{A_1}, then the programmed relation in J^* with $Q^* = a.D$ and $M_{pb} = b.D$ gives the standard ROM extractor (e.g. by forking on H_{i3}/H_{i4}) the chance to extract abD, solving HECDH.

Probability Analysis:

- The probability that C_1 must program q_i to a given value (0) after A_1 has already set it to another random value is $\leq Q_{PK}/2^k$. In two functions of protection, this is given away.

$$\Pr[\text{Event 1}] \geq \left(1 - \frac{Q_{PK}}{2^k}\right)^2$$

- **Event 2:** With $\phi = \frac{1}{Q_{SC}+1}$, and

$$\Pr[\text{Event 2}] = (1-\phi)^{Q_{SC}} = \left(\frac{Q_{SC}}{Q_{SC}+1}\right)^{Q_{SC}} \approx \varepsilon^{-1}$$

- **Event 3:** There is a probability of the simulator randomly selecting a slot equal to the challenge slot, and this probability is precisely

$$\Pr[\text{Event 3}] = \phi = \frac{1}{Q_{SC}+1}$$

Combining:

$$\Pr[\text{Event 1} \wedge \text{Event 2} \wedge \text{Event 3}] \geq \left(1 - \frac{Q_{PK}}{2^k}\right)^2 \cdot \frac{1}{\varepsilon(Q_{SC}+1)}.$$

Multiplying by ξ_{A_1} gives:

$$\mathrm{Adv}_{C_1}^{HECDH} \geq \left(1 - \frac{2Q_{PK}}{2^k}\right)^2 \cdot \frac{\xi_{A_1}}{\varepsilon(Q_{SC}+1)}.$$

□

Theorem 2. *If a Type II adversary A_2 possesses an unignorable advantage ξ_{A_2} in breaching the IND-CCA2 confidentiality of our improved scheme in the Random Oracle Model, then there exists a challenger C_2 that can solve the Hyper Elliptic Curve Computational Diffie-Hellman problem (HECDH) with the following probability:*

$$Adv_{C_2}^{HECDH} \geq \frac{\xi_{A_2}}{\varepsilon(Q_{SC}+1)}.$$

Proof. C_2 will interact with A_2 to solve the HECDH problem $(D, aD, bD) \rightarrow abD$ for our proposed scheme by following a series of steps.

Initialization: Since A_2 knows P_{ms}, C_2 computes $M_{pb} = P_{ms} \cdot D$, then provides P_{ms} and public parameters $param = \{M_{pb}, D, n, h_1, h_2, h_3, h_4, h_5\}$ to A_2.

Public Key Queries (Q_{PBK}): Since A_2 cannot replace public keys, C_2 answers as follows:

- For all non-challenge identities $O_{id} \neq O^*_{id}$: choose $\varsigma_d, \varkappa_d \in \{1, 2, \dots, n\}$, set $S_d = \varsigma_d D$, $X_d = \varkappa_d D$, and return (S_d, X_d) to A_2.

- For the challenge identity O^*_{id}: sample $\varsigma^*_d, \varkappa^*_d$, set $S^*_d = \varsigma^*_d \cdot D$, $X^*_d = b \cdot D$, and return (S^*_d, X^*_d) to A_2.

Anonymous Identity Queries (Q_{AI}): If A_2 asks for an anonymous identity, C_2 selects $b_d \in \{1, 2, \dots, n\}$, sets $B_d = b_d \cdot D$, and looks up $y = h_5(b_d.M_{pb}, M_{pb}, T_1)$ in L_{ish5}. Then, it sets $A_{id} = O_{id} \oplus y$, stores $(b_d, B_d, A_{id}, T_1, O_{id})$ in L_{ish5}, and returns (B_d, A_{id}).

Hash Queries: These are handled identically as in Theorem 1.

Partial Private Key Queries (Q_{PPK}): If A_2 requests a partial private key, C_2 checks $O_{id} \neq O^*_{id}$, calls Q_{AI}, and returns (B_d, A_{id}).

Full Private Key Queries (Q_{FPK}): If A_2 asks for a full private key, C_2 checks if $O_{id} = O^*_{id}$ and aborts. Otherwise, it checks q_i in L_{ish1}, sets $q_i = 0$ if $q_i \neq 0$, then returns $(D_d, \varkappa_d)$.

Signcryption Queries (Q_{SC}): C_2 ensures (S_d, X_d) and (S_{gs}, X_{gs}) exist in L_{isK}, and that x_{gs} is known if GS is honest. It selects $\sigma \in \mathbb{Z}^*_p$, sets $Q = \sigma \cdot D$, computes $T = S_d \cdot x_{gs}$, and queries H_{i2}, H_{i3}, H_{i4}. Then sets $q_i = h_1(O_{id}, S_d)$, defines $J = H_{i3}Q + H_{i4}X_d + H_{i4}q_i M_{pb}$, selects j, computes $C_i = (j, m) \oplus H_{i2}$, and returns $f = (Q, J, C)$ to A_2.

Unsigncryption Queries (Q_{USC}): C_2 computes $T = S_d \cdot x_{gs}$, queries H_{i2} on (Q, T), parses $(j, m) = C_i \oplus H_{i2}$, retrieves q_i, H_{i3}, H_{i4}, and checks whether

$$J = \sum_{i=0}^{n} H_{i3}Q + \sum_{i=0}^{n} H_{i4}X_d + \left(\sum_{i=0}^{n} H_{i4}q_d M_{pb} \right)$$

holds. If valid, it returns m; otherwise, it returns error.

Challenge: A_2 sends two equal-length plaintexts (m_1, m_2) and the target sender identity O^*_{id}. C_2 chooses $\pi \in \{1, 2\}$, computes $Q^* = aD, T^* = S^*_d x_{gs}$, retrieves $q^*_i, H^*_{i2}, H^*_{i3}, H^*_{i4}$ from $L_{isqi}, L_{isH_{i2}}, L_{isH_{i3}}, L_{isH_{i4}}$, defines

$$J^* = H^*_{i3}Q + H^*_{i4}X^*_d + H^*_{i4}q^*_i M_{pb},$$

chooses any j^*, sets $C^* = (j^*, m_\pi) \oplus H^*_{i2}$, and returns $f^* = (Q^*, J^*, C^*)$ to A_2.

Phase 2: A_2 may perform the same queries as in Phase 1, except it cannot request the full private key for O^*_{id} nor decrypt the exact challenge $f^* = (Q^*, J^*, C^*)$.

Guess: A_2 outputs π'. If $\pi' = \pi$ with advantage ξ_{A_2}, then the programmed relation in J^* with $Q^* = aD$ and $M_{pb} = bD$ allows the ROM extractor (e.g., via forking on H_{i3}/H_{i4}) to extract abD, solving HECDH.

Probability Analysis:

- **Event 1:** Since A_2 knows P_{ms} and all non-challenge $\varkappa_d$, it can correctly answer all allowed Q_{PK} queries. Thus,

$$\Pr[\text{Event 1}] = 1, \ \left(\text{no } (1 - (\tfrac{Q_{PK}}{2^k}))^2 \text{ loss}\right).$$

- **Event 2:** With $\phi = \frac{1}{Q_{SC}+1}$,

$$\Pr[\text{Event 2}] = (1-\phi)^{Q_{SC}} = \left(\frac{Q_{SC}}{Q_{SC}+1} \right)^{Q_{SC}} \approx \varepsilon^{-1}.$$

- **Event 3:** The probability that the simulator selects the correct challenge slot is

$$\Pr[\text{Event 3}] = \phi = \frac{1}{Q_{SC}+1}.$$

Combining:

$$\Pr[\text{Event 1} \wedge \text{Event 2} \wedge \text{Event 3}] \geq \frac{1}{\varepsilon(Q_{SC}+1)}.$$

Multiplying by ξ_{A_2} gives:

$$\text{Adv}^{HECDH}_{C_2} \geq \frac{\xi_{A_2}}{\varepsilon(Q_{SC}+1)}.$$

□

Theorem 3. *If a Type I adversary A_1 possesses an unignorable advantage ξ_{sig1} in breaching the EUF-CMA unforgeability of our improved scheme in the Random Oracle Model, then there exists a challenger C_1 that can solve the HECDH with the following probability:*

$$Adv^{HECDH}_{C_1} \geq \left(1 - \frac{2Q_{PK}}{2^k}\right)^2 \cdot \frac{\xi_{sig1}}{\varepsilon(Q_{SC}+1)},$$

where Q_{PK} and Q_{SC} denote the number of private key and signcryption queries respectively, k is the bit length of the group order, and ε is Euler's number (the base of the natural logarithm).

Proof. C_1 interacts with A_1 to solve the HECDH problem $(D, aD, bD) \to abD$ for our proposed scheme by following a series of steps.

Initialization: C_1 sets $P_{ms} = b \in \mathbb{Z}^*_n$ (unknown to C_1) and $M_{pb} = bD$, then publishes the public parameters

$$param = \{M_{pb}, D, n, h_1, h_2, h_3, h_4, h_5\}.$$

Queries: A_1 may issue $Q_{PBK}, Q_{RPBK}, Q_{AI}, Q_{h_1}, Q_{h_2}, Q_{h_3}, Q_{h_4}, Q_{h_5}, Q_{PPK}, Q_{FPK}$, and Q_{SC}, which are handled identically as in Theorem 1.

Forgery: A_1 succeeds only if it produces a new signcryption triple $f^* = (Q^*, J^*, C^*)$ for the target (O^*_{id}, O_{gs}, m^*). If $O^*_{id} \neq O^*$ or f^* is invalid, C_1 aborts. Otherwise, by applying the *Forking Lemma*, A_1 can generate another valid triple $f^{**} = (Q^{**}, J^{**}, C^{**})$ by modifying the programmed value of H_{i4} on the challenge input. This yields two accepting equations $J^* = H^*_{i3}Q^* + H^*_{i4}X^*_d + H^*_{i4}q^*_i M_{pb}$, $J^{**} = H^{**}_{i3}Q^{**} + H^{**}_{i4}X^{**}_d + H^{**}_{i4}q^{**}_i M_{pb}$. Subtracting gives $\Delta J = (H^*_{i3} - H^{**}_{i3})Q^* + (H^*_{i4} - H^{**}_{i4})(X^*_d + q^*_i M_{pb})$. If we fork on H_{i4} while keeping H_{i3} fixed (so $H^*_{i3} = H^{**}_{i3}$ and $H^*_{i4} \neq H^{**}_{i4}$), then $X^*_d + q^*_i M_{pb} = \frac{\Delta J}{H^*_{i4} - H^{**}_{i4}}, \quad q^*_i M_{pb} = \frac{\Delta J}{H^*_{i4} - H^{**}_{i4}} - X^*_d$. Under the challenge embedding $Q^* = aD$ and $M_{pb} = bD$, these relations allow the extractor to isolate the target point abD, thereby solving HECDH.

Probability Analysis:

- **Event 1:** The probability that C_1 must program q_i to a fixed value after A_1 has already set it to another random value is at most $Q_{PK}/2^k$. Thus,

$$\Pr[\text{Event 1}] \geq \left(1 - \frac{Q_{PK}}{2^k}\right)^2.$$

- **Event 2:** With $\phi = \frac{1}{Q_{SC}+1}$,

$$\Pr[\text{Event 2}] = (1-\phi)^{Q_{SC}} = \left(\frac{Q_{SC}}{Q_{SC}+1}\right)^{Q_{SC}} \approx \varepsilon^{-1}.$$

- **Event 3:** The probability that the simulator selects the correct challenge slot is

$$\Pr[\text{Event 3}] = \phi = \frac{1}{Q_{SC}+1}.$$

Combining:

$$\Pr[\text{Event 1}\wedge\text{Event 2}\wedge\text{Event 3}] \geq \left(1 - \frac{Q_{PK}}{2^k}\right)^2 \cdot \frac{1}{\varepsilon(Q_{SC}+1)}.$$

Multiplying by ξ_{sig1} gives the reduction advantage:

$$\text{Adv}_{C_1}^{HECDH} \geq \left(1 - \frac{2Q_{PK}}{2^k}\right)^2 \cdot \frac{\xi_{\text{sig1}}}{\varepsilon(Q_{SC}+1)}.$$

□

Theorem 4. *If a Type I adversary A_2 possesses an unignorable advantage ξ_{sig2} in breaching the EUF-CMA unforgeability of our improved scheme in the Random Oracle Model, then there exists a challenger C_2 that can solve the HECDH with the following probability:*

$$Adv_{C_2}^{HECDH} \geq \frac{\xi_{sig2}}{\varepsilon(Q_{SC}+1)}.$$

Proof. C_2 interacts with A_2 to solve the HECDH problem $(D, aD, bD) \to abD$ for the proposed scheme as follows.

Initialization: Since A_2 already knows P_{ms}, C_2 computes $M_{pb} = P_{ms} \cdot D$ and provides P_{ms} together with public parameters $param = \{M_{pb}, D, n, h_1, h_2, h_3, h_4, h_5\}$ to A_2.

Queries: A_2 performs $Q_{PBK}, Q_{RPBK}, Q_{AI}, Q_{h_1}, Q_{h_2}, Q_{h_3}, Q_{h_4}, Q_{h_5}, Q_{PPK}, Q_{FPK}$, and Q_{SC} queries identically as described in Theorem 2.

Forgery: A_2 succeeds if it outputs a new valid signcryption triple $f^* = (Q^*, J^*, C^*)$ for the target (O_{id}^*, O_{gs}, m^*). If $O_{id}^* \neq O^*$ or f^* is invalid, C_2 aborts. Otherwise, applying the Forking Lemma, A_2 produces a second valid signcryption $f^{**} = (Q^{**}, J^{**}, C^{**})$ with a modified response from H_{i4}. This yields two accepting equations: $J^* = H_{i3}^* Q^* + H_{i4}^* X_d^* + H_{i4}^* q_i^* M_{pb}, J^{**} = H_{i3}^{**} Q^{**} + H_{i4}^{**} X_d^{**} + H_{i4}^{**} q_i^{**} M_{pb}$. Subtracting, we obtain:$\Delta J = (H_{i3}^* - H_{i3}^{**})Q^* + (H_{i4}^* - H_{i4}^{**})(X_d^* + q_i^* M_{pb})$. If the fork is applied on H_{i4} while keeping H_{i3} fixed (so $H_{i3}^* = H_{i3}^{**}$ and $H_{i4}^* \neq H_{i4}^{**}$), then $X_d^* + q_i^* M_{pb} = \frac{\Delta J}{H_{i4}^* - H_{i4}^{**}}$. Under the challenge embedding $Q^* = aD$ and $M_{pb} = bD$, this allows the extractor to compute abD, thus solving HECDH.

Probability Analysis:

- **Event 1:** Since A_2 knows P_{ms} and all non-challenge $\varkappa_d$, it correctly answers all Q_{PK} queries. No programming loss occurs, hence

$$\Pr[\text{Event 1}] = 1, \text{ (no } (1 - (Q_{PK}/2^k))^2 \text{ loss).}$$

- **Event 2:** With $\phi = \frac{1}{Q_{SC}+1}$, and

$$\Pr[\text{Event 2}] = (1-\phi)^{Q_{SC}} = \left(\frac{Q_{SC}}{Q_{SC}+1}\right)^{Q_{SC}} \approx \varepsilon^{-1}.$$

- **Event 3:** The probability that the simulator selects the correct challenge slot is

$$\Pr[\text{Event 3}] = \phi = \frac{1}{Q_{SC}+1}.$$

Combining all events:

$$\Pr[\text{Event 1} \wedge \text{Event 2} \wedge \text{Event 3}] \geq \frac{1}{\varepsilon(Q_{SC}+1)}.$$

Multiplying by the adversary's advantage ξ_{sig2} gives:

$$\text{Adv}_{C_2}^{HECDH} \geq \frac{\xi_{sig2}}{\varepsilon(Q_{SC}+1)}.$$

□

Theorem 5 (Authentication)**.** *The proposed scheme ensures message origin authentication by verifying the validity of aggregate signatures at the GS.*

Proof. Each drone computes the signature as:

$$j_i = (\sigma_i H_{i3} + D_d H_{i4}) \mod n$$

and sends:

$$f_i = (Q_i, J_i, C_i), \quad J_i = v_i \cdot D$$

Upon receiving all f_i, MEC aggregates the signatures, $J = \sum_{i=0}^{n} j_i$ and sends f_{agg} to GS. GS verifies authenticity by checking:

$$J \overset{?}{=} \sum_{i=0}^{n} H_{i3} Q_i + \sum_{i=0}^{n} H_{i4} X_d + \left(\sum_{i=0}^{n} H_{i4} q_d\right) M_{pb}$$

Expanding using $X_d = \varkappa_d \cdot D$, $D_d = \varkappa_d + P_{ms} \cdot q_d$, we get:

$$\begin{aligned} J &= \sum_{i=0}^{n} H_{i3} Q_i + \sum_{i=0}^{n} H_{i4} \varkappa_d \cdot D + \sum_{i=0}^{n} H_{i4} q_d \cdot P_{ms} \cdot D \\ &= \sum_{i=0}^{n} (H_{i3} \cdot \sigma_i + H_{i4} \cdot D_d) \cdot D \\ &= \sum_{i=0}^{n} j_i = J \quad \text{(proven)} \end{aligned} \tag{31}$$

This confirms the validity of the drone's origin and message authenticity. □

Theorem 6 (Integrity)**.** *The proposed scheme guarantees message integrity, assuming the collision resistance and determinism of the hash functions h_2, h_3, and h_4, and that the same inputs are used by both the sender (Drone) and the receiver (GS).*

Proof. In the data aggregation phase, the drone computes the following:

$$H_{i2} = h_2(Q_i, T_i)$$
$$H_{i3} = h_3(X_{gs}, X_d, Q_i, T_i, O_{gs})$$
$$H_{i4} = h_4(X_{gs}, X_d, Q_i, T_i, O_g, H_{i3})$$

and generates the signature and ciphertext as:

$$j_i = (\sigma_i \cdot H_{i3} + D_d \cdot H_{i4}) \bmod n, \quad C_i = (j_i, m_i) \oplus H_{i2}$$

The tuple (Q_i, J_i, C_i) is sent to the GS. Upon receiving this tuple, the GS computes $T_i = S_d \cdot x_{gs}$ and recomputes H_{i2}, H_{i3}, H_{i4} using the same inputs as the Drone. Using these, it recovers (j_i, m_i) from C_i and verifies the correctness of the aggregate signature by checking:

$$J = \sum_{i=0}^{n} H_{i3} \cdot Q_i + \sum_{i=0}^{n} H_{i4} \cdot X_d + \left(\sum_{i=0}^{n} H_{i4} \cdot q_d \right) \cdot M_{pb}$$

As the hash inputs are identical at both ends, and the hash functions are assumed to be collision-resistant and deterministic, the outputs must be the same. Any modification in C_i, Q_i, or any input affecting the hash values will result in a mismatch during verification, leading to the rejection of the signature. Hence, the proposed scheme satisfies the integrity requirement. □

Theorem 7 (Impersonation Resistance)**.** *The proposed scheme resists impersonation attacks by securely binding the user's identity during the AID-Gen and checking for duplication during key issuance.*

Proof. In the AID-Gen phase, the KGC receives the actual identity O_{id} from the drone and computes the anonymous identity as:

$$A_{id} = O_{id} \oplus h_5(\mathfrak{d}_d \cdot M_{pb}, M_{pb}, T_1)$$

where $\mathfrak{d}_d \in \{1, 2, \dots, n\}$ is a randomly selected scalar and T_1 denotes the time validity. The KGC stores the tuple (B_d, A_{id}, O_{id}) in its memory. Later, during Key-Gen, if an entity claiming to be a drone reuses the same identity O_{id}, the KGC searches its records. If a match is found, it denies the request and does not issue a new key. This identity binding and record checking prevent replay and reuse of registered identities by an outsider adversary. As a result, impersonation attempts by entities such as E_{ve} are detected and blocked, ensuring resistance to impersonation attacks. □

Theorem 8. *(Replay Resistance). The proposed scheme resists replay attacks by padding the time stamp with a message at the time of encryption and checking the session expiry for a specific time.*

Proof. In the Data-Aggregate phase, the Drone select a time stamp (t_1) and the pad with massage and generate a ciphertext as: $C_i = \left(j_i, m_i, t_1\right) \oplus H_{i2}$, then send to GS, the GS can decrypt as $\left(j_i, m_i, t_1\right) = C_i \oplus H_{i2}$, and check the freshness of (t_1), if it is fresh the consider it as a valid/new message from original source, otherwise return with error message. □

6. Performance Analysis

This section compares the performance of the proposed scheme with the schemes presented by Verma et al. [1], Aithekar et al. [2], and Ali et al. [3] in terms of computational cost and communication overhead.

6.1. Computational Cost

The computational cost of the proposed scheme is evaluated based on a set of standard cryptographic operations commonly used in the literature for performance benchmarking. These operations include single multiplication in bilinear pairing (SM_{BP}), single bilinear pairing operation (SP_{BP}), single point addition in bilinear pairing (SA_{BP}), single point addition on elliptic curves (SA_{EC}), and single point multiplication on elliptic curves (SM_{EC}). Additionally, the analysis includes single exponential operations (SE), one-way hash functions (H_{OH}), and operations specific to hyperelliptic curves, namely single divisor addition (SA_{HEC}) and single divisor multiplication (SM_{HEC}).

To evaluate the performance of our proposed scheme, we implemented the necessary cryptographic operations using the MIRACL library, which supports elliptic, pairing-based, and hyperelliptic curve arithmetic. The experiments were carried out on a Windows 7 Home Basic 64-bit operating system, running on an Intel Core $i7-4510U$ CPU at 2.0 GHz with 8 GB of RAM [39]. Each runtime was averaged over 1,000 executions to reduce measurement bias. For pairing-based cryptography, we used the super singular elliptic curve $EC/F_p : y^2 = x^3 - 3x$ with an embedding degree of 2, where q is a 160-bit Solinas prime ($q = 2^{159} + 2^{17} + 1$) and p is a 512-bit prime satisfying $p+1 = 2_{qh}+1$, offering an 80-bit AES-equivalent security level. For elliptic curve cryptography (ECC), we adopted the standard curve secp160r1, recommended by Certicom, to ensure the same 80-bit security. For HECC, we considered a genus-2 curve of the form HEC: $y^2 + h(x)y = f(x)$ defined over F_q, where $deg(h(x))$ and $deg(f(x)) = 5$. The Jacobian group order was chosen to be about 160 bits, providing an equivalent 80-bit security level. Group operations in this setting were performed on divisors in the Jacobian of the genus-2 curve using Cantor's algorithm, with efficient coordinate representation provided by MIRACL's classic implementation. The measured runtimes were as follows: SM_{BP} took 4.31 ms, SP_{BP} took 14.90 ms, SM_{EC} took 0.97 ms, and $SM_H EC$ took 0.48 ms. The computational cost using the primary operations of the proposed scheme, Verma et al. [4], Aithekar et al. [5], and Ali et al. [7], is given in Table 2. These foundational metrics are used to analyse the proposed scheme's overall computational overhead compared to existing schemes, as discussed in the

Table 2
Computation Cost Comparison (in milliseconds)

Scheme	Drone (ms)	GS (ms)	Total (ms)
Proposed (HECC)	$3SM_{HEC} = \mathbf{1.44}$	$2SM_{HEC} = \mathbf{0.96}$	$5SM_{HEC} = \mathbf{2.40}$
Verma et al. [1]	$2SM_{EC} = \mathbf{1.94}$	$2SM_{EC} = \mathbf{1.94}$	$5SM_{EC} = \mathbf{4.85}$
Aithekar et al. [2]	$2SM_{EC} = \mathbf{1.94}$	$1SM_{EC} = \mathbf{0.97}$	$3SM_{EC} = \mathbf{2.91}$
Ali et al. [3]	$3SM_{EC} = \mathbf{2.91}$	$3SM_{EC} = \mathbf{2.91}$	$6SM_{EC} = \mathbf{5.82}$

Table 3
Verification Scalability Summary (Total Verification Time at GS in ms)

n (messages)	Proposed (HECC)	Verma et al. [1]	Aithekar et al. [2]	Ali et al. [3]
10	**9.6**	19.4	9.7	29.1
50	**48.0**	97.0	48.5	145.5
100	**96.0**	194.0	97.0	291.0
200	**192.0**	388.0	194.0	582.0

subsequent analysis and Table 2: in our proposed scheme, the Drone requires $3SM_{HEC} = 3 \times 0.48 = 1.44$ ms, the GS requires $2SM_{HEC} = 2 \times 0.48 = 0.96$ ms, and the total is $5SM_{HEC} = 5 \times 0.48 = 2.40$ ms. Verma et al. [4] incurs $2SM_{EC} = 2 \times 0.97 = 1.94$ ms at the Drone, $2SM_{EC} = 2 \times 0.97 = 1.94$ ms at the GS, and $5SM_{EC} = 5 \times 0.97 = 4.85$ ms in total. Aithekar et al. [5] requires $2SM_{EC} = 2 \times 0.97 = 1.94$ ms at the Drone, $1SM_{EC} = 1 \times 0.97 = 0.97$ ms at the GS, and $3SM_{EC} = 3 \times 0.97 = 2.91$ ms in total. Ali et al. [7] incurs $3SM_{EC} = 3 \times 0.97 = 2.91$ ms at the Drone, $3SM_{EC} = 3 \times 0.97 = 2.91$ ms at the GS, and $6SM_{EC} = 6 \times 0.97 = 5.82$ ms in total. Further we add scalability analysis, in which we show how the total verification time at the GS side grows with the number of signcrypted messages n. Using single message verification times from Table 2, we model the total as $T(n) = t_{verify} \times n$ and generate Table 3. We Start from $n = 10$ matches realistic batch sizes and avoids small-n noise.

The analysis shows (also depicted in Fig. 2 and 3) that the proposed scheme offers a significantly reduced computational cost due to the use of hyperelliptic curve operations, which are more lightweight than elliptic curve operations. This makes it well-suited for resource-constrained devices such as drones.

6.2. Communication Cost

This subsection presents a comparative analysis of the communication cost between the proposed aggregated authentication and signature scheme and the schemes introduced by Verma et al. [1], Aithekar et al. [2], and Ali et al. [3].

To perform this comparison, the following parameter sizes (in bits) are considered: bilinear pairing parameter size ($BG = 1024$), elliptic curve parameter size ($eq = 160$), hyperelliptic curve parameter size ($hn = 80$), ciphertext size ($CIP = 1024$), timestamp size in EC-based schemes ($T_{eq} = 160$), and timestamp size in HEC-based schemes ($T_{hn} = 80$).

The communication cost is computed as the size of the ciphertext along with the associated parameters transmitted during the authentication process. In the proposed scheme, the cost includes the ciphertext and two additional hyperelliptic curve parameters:

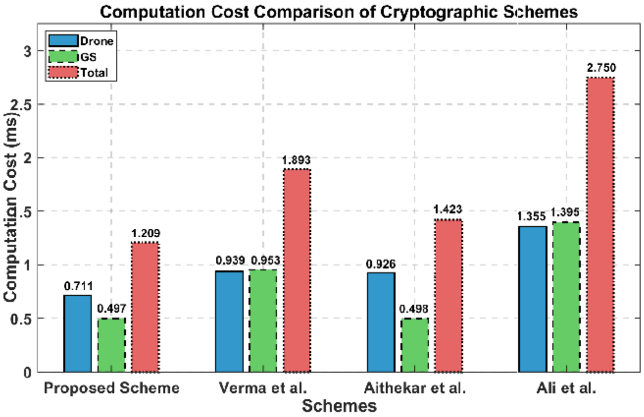

Figure 2: Computation cost comparison of cryptographic schemes at the Drone, GS, and overall levels. The proposed scheme demonstrates the lowest total computational cost.

$$\text{Comm}_{\text{Proposed}} = |CIP| + 2 \cdot |hn| = 1024 + 2 \cdot 80 = 1184 \text{ bits} \tag{32}$$

In comparison, the communication overhead for the scheme by Verma et al. [1] is:

$$\text{Comm}_{\text{Verma}} = |CIP| + 2 \cdot |eq| = 1024 + 2 \cdot 160 = 1344 \text{ bits} \tag{33}$$

Similarly, Aithekar et al. [2] incurs the same communication cost:

$$\text{Comm}_{\text{Aithekar}} = |CIP| + 2 \cdot |eq| = 1024 + 2 \cdot 160 = 1344 \text{ bits} \tag{34}$$

Ali et al. [3] introduce an additional EC parameter, leading to a higher communication cost:

$$\text{Comm}_{\text{Ali}} = |CIP| + 3 \cdot |eq| = 1024 + 3 \cdot 160 = 1504 \text{ bits}$$

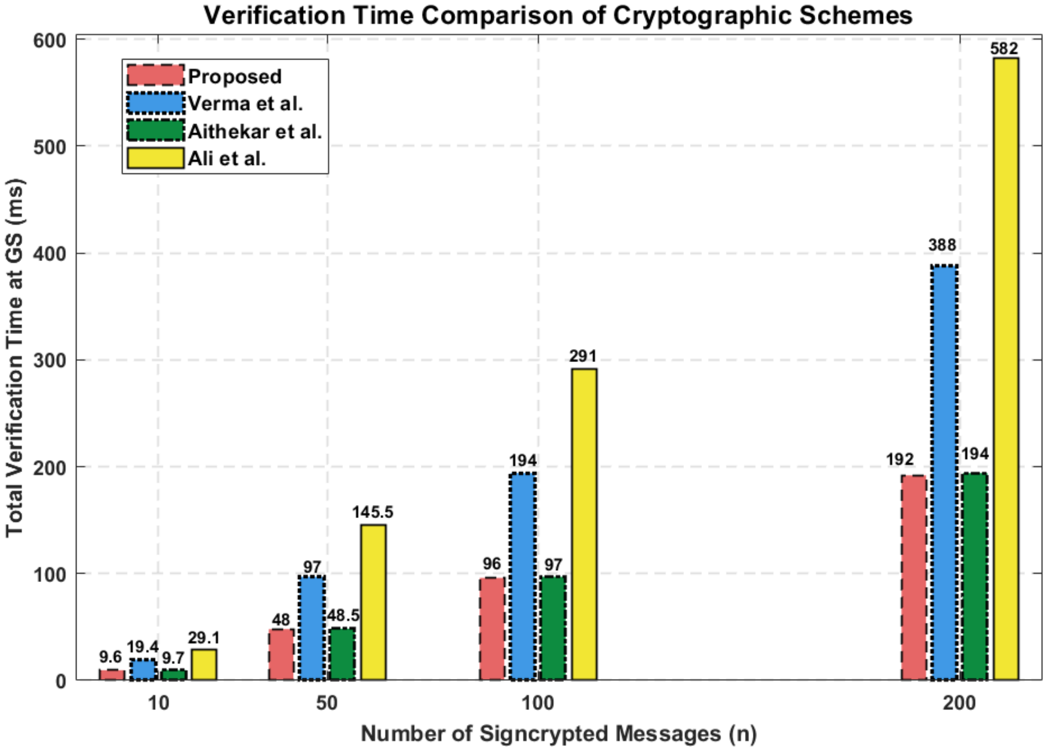

Figure 3: Total verification time vs. number of signcrypted messages n (starting at $n = 10$).

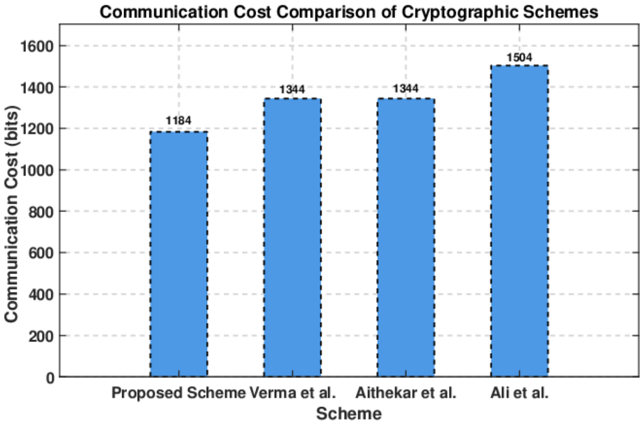

Figure 4: Communication cost comparison among cryptographic schemes based on ciphertext size and related parameters. The proposed scheme achieves the lowest overall communication overhead.

(35)

From this analysis, it is evident that the proposed scheme achieves a lower communication overhead compared to existing works. This improvement is further illustrated in Fig. 4 and supported by the values reported in Table 4. The overall performance including security requirements, computational cost, and communication cost of the proposed scheme with that of recently published works by Verma et al. [1], Aithekar et al. [2], and Ali et al. [3] is presented in Table 5. The results show that our proposed scheme consistently outperforms prior work.

Table 4
Communication Cost Comparison (in bits)

Scheme	Communication Parameters	Total Cost
Proposed Scheme	$\|CIP\| + 2\|hn\| = 1024 + 2 \cdot 80$	1184
Verma et al. [1]	$\|CIP\| + 2\|eq\| = 1024 + 2 \cdot 160$	1344
Aithekar et al. [2]	$\|CIP\| + 2\|eq\| = 1024 + 2 \cdot 160$	1344
Ali et al. [3]	$\|CIP\| + 3\|eq\| = 1024 + 3 \cdot 160$	1504

6.3. Discussion and Limitations

The above analysis demonstrates that the proposed HECC-based aggregation signcryption scheme achieves strong security guarantees and significantly reduces computational and communication overhead compared with recent alternatives. Nevertheless, several limitations should be acknowledged. First, the scheme assumes a fully trusted and always available KGC, which may become a single point of failure if compromised or unreachable. This setup brings the key escrow problem. The KGC can decrypt any signcrypted message because it knows all partial private keys. For privacy, this means one point of compromise. If the KGC is hacked, an attacker can get all session keys and plaintext data. For deployment, centralized key management limits scalability in large or distributed IoD networks. It may also conflict with

Table 5
Differences between our proposed scheme and Existing schemes, highlighting efficiency gains and key security properties

Scheme	Comm. Cost Reduction vs Proposed (%)	Comp. Cost Reduction vs Proposed (%)	Confidentiality	Unforgeability	Authentication	Impersonation Resistance
Proposed Scheme	–	–	✓	✓	✓	✓
Verma et al. [1]	11.90	50.52	✓	×	×	×
Aithekar et al. [2]	11.90	17.53	✓	×	×	×
Ali et al. [3]	21.28	58.76	✓	×	×	×
Average Reduction (Existing vs Proposed)	**15.03**	**42.27**	–	–	–	–

Notes: Reductions are computed as $\frac{\text{Existing} - \text{Proposed}}{\text{Existing}} \times 100$. ✓ = provided, × = not provided.

privacy rules like GDPR that limit single entities with full decryption power. Options like certificateless or threshold schemes can fix this, but they are outside this work's scope. Second, the current design targets a single ground station architecture; extending the protocol to multi-GS or multi-recipient scenarios would require additional mechanisms for key management and aggregation consistency. Third, our analysis focuses on the cryptographic layer and abstracts from lower-layer networking aspects such as routing, interference, and packet loss, which may affect performance in real deployments. Fourth, while the reduced computational cost implies lower energy consumption for battery-limited UAVs, as fewer operations mean less processor usage and heat, we did not measure actual power draw. This is because energy profiling needs specific hardware like drone boards, which is beyond this paper's scope. Future work can include energy tests on real UAVs to confirm these benefits. Fifth, the centralised KGC raises ethical concerns in civilian drone deployments, such as potential misuse of key escrow for surveillance, eroding user privacy. Regulatory implications include conflicts with laws like FAA rules on drone data security or GDPR requirements for data minimisation and decentralised control. Future work could explore ethical guidelines and compliant designs, like distributed key management, to better suit civilian applications. Finally, the construction relies on classical HECC assumptions and therefore does not provide post-quantum security. Addressing these limitations by distributing trust, generalising the network topology, incorporating realistic communication models, and designing post-quantum variants will be an important direction for future work.

7. Conclusion

This paper critically examines the signcryption scheme proposed by Verma et al. and identifies multiple security flaws, including its vulnerability to forgery and impersonation attacks and internal inconsistencies within its mathematical formulation. To overcome these shortcomings, we present an improved aggregation-based signcryption scheme leveraging the security and efficiency advantages of hyperelliptic curve cryptography. The proposed construction integrates identity-based key generation, robust signcryption, and aggregate verification while ensuring provable security properties under the HECDLP hardness assumption. Through formal security proofs and detailed performance evaluations, we show that the proposed scheme satisfies critical cryptographic guarantees, confidentiality, integrity, authentication, unforgeability, and impersonation resistance, without compromising efficiency. A comparative analysis against recent schemes, including those by Verma et al. [1], Aithekar et al. [2], and Ali et al. [3], demonstrates the superiority of our approach in terms of both computational and communication costs. The scheme's practical performance, validated through implementation benchmarks, makes it well-suited for secure communication in drone-to-ground station systems, particularly in resource-constrained edge computing environments. Future work will explore extending the scheme toward multi-recipient aggregation and post-quantum cryptographic primitives to further enhance resilience and scalability [40]. In addition, we plan to integrate the protocol into realistic IoD/UAV simulation environments or hardware testbeds to evaluate end-to-end performance under dynamic channel conditions and mobility patterns.

Acknowledgement

This work was supported by King Saud University, Riyadh, Saudi Arabia, through the Ongoing Research Funding program (ORF-2025-184).

Declaration

Declaration of generative AI and AI-assisted technologies in the writing process: During the preparation of this work, the author(s) used Grammarly in order to improve the language and readability. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

References

[1] G. K. Verma, V. Chamola, N. Kumar, A. K. Das, D. Mishra, Efficient and secure signcryption-based data aggregation for internet of drone-based drone-to-ground station communication, Ad Hoc Netw. 159 (2024) 103502. doi:10.1016/j.adhoc.2024.103502.

[2] A. Aithekar, P. Gupta, D. Chaudhary, A secure and efficient heterogeneous id-based signcryption for unmanned aerial vehicular networking system, Secur. Privacy 7 (5) (2024) e389. doi:10.1002/spy2.389.

[3] I. Ali, J. Li, J. Chen, Y. Chen, S. Ullah, S. Khan, Ioosc-u2g: An identity-based online/offline signcryption scheme for unmanned aerial vehicle to ground station communication, IEEE Internet Things J. 11 (18) (2024) 29941–29955. doi:10.1109/JIOT.2024.3407767.

[4] M. Yahuza, M. Y. I. Idris, I. B. Ahmedy, A. W. A. Wahab, T. Nandy, N. M. Noor, A. Bala, Internet of drones security and privacy issues: taxonomy and open challenges, IEEE Access 9 (2021) 57243–57270.

[5] L. Abualigah, A. Diabat, P. Sumari, A. H. Gandomi, Applications, deployments, and integration of internet of drones (iod): a review, IEEE Sensors J. 21 (22) (2021) 25532–25546.

[6] M. Shin, S. T. Shah, M. Y. Chung, S. F. Hasan, B.-C. Seet, P. H. J. Chong, Moving small cells in public safety networks, in: 2017 International Conference on Information Networking (ICOIN), 2017, pp. 564–568. doi:10.1109/ICOIN.2017.7899559.

[7] A. Heidari, N. J. Navimipour, M. Unal, G. Zhang, Machine learning applications in internet-of-drones: systematic review, recent deployments, and open issues, ACM Comput. Surv. 55 (12) (2023) 1–45, art. 247, Mar.

[8] M. A. Shawky, S. T. Shah, M. Abdrabou, M. Usman, Q. H. Abbasi, D. Flynn, M. A. Imran, S. Ansari, A. Taha, How secure are our roads? an in-depth review of authentication in vehicular communications, Veh. Commun. 47 (2024) 100784. doi:10.1016/j.vehcom.2024.100784.

[9] K. Mahmood, Z. Ghaffar, M. Farooq, K. Yahya, A. K. Das, S. A. Chaudhry, A security enhanced chaotic-map-based authentication protocol for internet of drones, IEEE Internet Things J. 11 (12) (2024) 22301–22309.

[10] B. Branco, J. S. S. Silva, M. Correia, Cyber attacks on commercial drones: a review, IEEE Access 13 (2025) 9566–9577.

[11] I. Bhattarai, C. Pu, K.-K. R. Choo, A lightweight aggregate authentication protocol for internet of drones, in: Proc. IEEE 21st Consumer Commun. Netw. Conf. (CCNC), 2024, pp. 143–151.

[12] Y. Zheng, Digital signcryption or how to achieve cost (signature & encryption) $\ll$ cost (signature) + cost (encryption), in: Proc. Annu. Int. Cryptology Conf. (CRYPTO), Santa Barbara, CA, USA, 1997, pp. 165–179.

[13] M. Barbosa, P. Farshim, Certificateless signcryption, in: Proc. ACM Symp. Inf., Comput. Commun. Secur. (ASIACCS), Tokyo, Japan, 2008, pp. 369–372.

[14] H. Yu, R. Ren, Certificateless elliptic curve aggregate signcryption scheme, IEEE Syst. J. 16 (2) (2021) 2347–2354.

[15] S. S. D. Selvi, S. S. Vivek, J. Shriram, S. Kalaivani, C. P. Rangan, Identity-based aggregate signcryption schemes, in: Proc. Int. Conf. Cryptology in India (INDOCRYPT), Vol. 5922 of LNCS, Springer, New Delhi, India, 2009, pp. 378–397.

[16] G. K. Verma, B. B. Singh, N. Kumar, V. Chamola, Cb-cas: Certificate-based efficient signature scheme with compact aggregation for industrial internet of things environment, IEEE Internet Things J. 7 (4) (2020) 2563–2572.

[17] B. Cao, M. Li, X. Liu, J. Zhao, W. Cao, Z. Lv, Many-objective deployment optimization for a drone-assisted camera network, IEEE Trans. Netw. Sci. Eng. 8 (4) (2021) 2756–2764.

[18] H. Wang, Z. Liu, Z. Liu, D. S. Wong, Identity-based aggregate signcryption in the standard model from multilinear maps, Front. Comput. Sci. 10 (2016) 741–754.

[19] G. Swapna, P. V. Reddy, Efficient identity based aggregate signcryption scheme using bilinear pairings over elliptic curves, J. Phys.: Conf. Ser. 1344 (1) (2019) 012010.

[20] E. Abouelkheir, S. El-sherbiny, Pairing free identity based aggregate signcryption scheme, IET Inf. Secur. 14 (6) (2020) 625–632.

[21] H. Yu, R. Ren, Certificateless elliptic curve aggregate signcryption scheme, IEEE Syst. J. 16 (2) (2021) 2347–2354.

[22] Y. Yang, D. He, P. Vijayakumar, B. B. Gupta, Q. Xie, An efficient identity-based aggregate signcryption scheme with blockchain for iot-enabled maritime transportation system, IEEE Trans. Green Commun. Netw. 6 (3) (2022) 1520–1531.

[23] M. A. Shawky, S. T. Shah, Q. H. Abbasi, M. Hussein, M. A. Imran, S. F. Hasan, S. Ansari, A. Taha, Ris-enabled secret key generation for secured vehicular communication in the presence of denial-of-service attacks, Sensors 23 (8) (2023). doi:10.3390/s23084104.
URL https://www.mdpi.com/1424-8220/23/8/4104

[24] M. Wazid, A. K. Das, V. Chamola, Y. Park, Uniting cyber security and machine learning: Advantages, challenges and future research, ICT Express 8 (3) (2022) 313–321.

[25] K. Cao, H. Ding, W. Li, L. Lv, M. Gao, F. Gong, B. Wang, On the ergodic secrecy capacity of intelligent reflecting surface aided wireless powered communication systems, IEEE Wireless Commun. Lett. 11 (11) (2022) 2275–2279.

[26] J. Won, S.-H. Seo, E. Bertino, Certificateless cryptographic protocols for efficient drone-based smart city applications, IEEE Access 5 (2017) 3721–3749.

[27] Q. Wu, P. Sun, A. Boukerche, An energy-efficient uav-based data aggregation protocol in wireless sensor networks, in: Proc. 8th ACM Symp. Design Anal. Intell. Veh. Netw. Appl. (DIVANet), 2018, pp. 34–40.

[28] A. Bera, S. Misra, C. Chatterjee, S. Mao, Cedan: Cost-effective data aggregation for uav-enabled iot networks, IEEE Trans. Mobile Comput. 22 (9) (2023) 5053–5063.

[29] Global Market Insights, Military drone market size – by motors, application, platform type, range, propulsion, size/weight class, mode of operation, end use, take-off, analysis, share, growth forecast, 2024–2032, GMI Industry Report, online. Available: https://www.gminsights.com/industry-analysis/military-drone-market (Accessed: Apr. 3, 2025) (Oct. 2024).

[30] P. Gope, B. Sikdar, An efficient privacy-preserving authenticated key agreement scheme for edge-assisted internet of drones, IEEE Trans. Veh. Technol. 69 (11) (2020) 13621–13630.

[31] J. Won, S.-H. Seo, E. Bertino, A secure communication protocol for drones and smart objects, in: Proc. 10th ACM Symp. Inf., Comput. Commun. Secur. (ASIACCS), 2015, pp. 249–260.

[32] M. Wazid, A. K. Das, N. Kumar, A. V. Vasilakos, J. J. P. C. Rodrigues, Design and analysis of secure lightweight remote user authentication and key agreement scheme in internet of drones deployment, IEEE Internet Things J. 6 (2) (2019) 3572–3584.

[33] T. Alladi, N. Naren, G. Bansal, V. Chamola, M. Guizani, Secauthuav: A novel authentication scheme for uav-ground station and uav-uav communication, IEEE Trans. Veh. Technol. 69 (12) (2020) 15068–15077.

[34] M. Tanveer, N. Kumar, M. M. Hassan, Ramp-iod: A robust authenticated key management protocol for the internet of drones, IEEE Internet Things J. 9 (2) (2022) 1339–1353.

[35] T. Alladi, V. Chamola, Naren, N. Kumar, Parth: A two-stage lightweight mutual authentication protocol for uav surveillance networks, Comput. Commun. 160 (2020) 81–90.

[36] Y. Zhang, D. He, L. Li, B. Chen, A lightweight authentication and key agreement scheme for internet of drones, Comput. Commun. 154 (2020) 455–464.

[37] B. D. Deebak, F. Al-Turjman, A smart lightweight privacy preservation scheme for iot-based uav communication systems, Comput. Commun. 162 (2020) 102–117.

[38] J. Whelan, A. Almehmadi, K. El-Khatib, Artificial intelligence for intrusion detection systems in unmanned aerial vehicles, Comput. Electr. Eng. 99 (2022) 107784.

[39] C. Zhou, Z. Zhao, W. Zhou, Y. Mei, Certificateless key-insulated generalized signcryption scheme without bilinear pairings, Secur. Commun. Netw. 2017 (1) (2017) 8405879. doi:10.1155/2017/8405879.

[40] S. M. Gilani, A. Anjum, A. Khan, M. H. Syed, S. A. Moqurrab, G. Srivastava, A robust internet of drones security surveillance communication network based on iota, Internet Things 25 (2024) 101066.