





Article

An Efficient and Conditional Privacy-Preserving Heterogeneous Signcryption Scheme for the Internet of Drones

Muhammad Asghar Khan ^{1,*}, Insaf Ullah ¹, Ako Muhammad Abdullah ^{2,3}, Syed Agha Hassnain Mohsan ⁴
and Fazal Noor ⁵

¹ Department of Electrical Engineering, Hamdard University, Islamabad 44000, Pakistan

² Computer Science Department, College of Basic Education, University of Sulaimani, Sulaimaniyah 00964, Kurdistan Region, Iraq

³ Department of Information Technology, University College of Goizha, Sulaimaniyah 00964, Kurdistan Region, Iraq

⁴ Ocean College, Zhejiang University, Zheda Road 1, Zhoushan 316021, China

⁵ Faculty of Computer and Information Systems, Islamic University of Madinah, Madinah 400411, Saudi Arabia

* Correspondence: m.asghar@hamdard.edu.pk; Tel.: +92-336-5276-465

Abstract: The Internet of Drones (IoD) is a network for drones that utilizes the existing Internet of Things (IoT) infrastructure to facilitate mission fulfilment through real-time data transfer and navigation services. IoD deployments, on the other hand, are often conducted in public wireless settings, which raises serious security and privacy concerns. A key source of these security and privacy concerns is the fact that drones often connect with one another through an unprotected wireless channel. Second, limits on the central processing unit (CPU), sensor, storage, and battery capacity make the execution of complicated cryptographic methods onboard a drone impossible. Signcryption is a promising method for overcoming these computational and security limitations. Additionally, in an IoD setting, drones and the ground station (GS) may employ various cryptosystems in a particular region. In this article, we offer a heterogeneous signcryption scheme with a conditional privacy-preservation option. In the proposed scheme, identity-based cryptography (IBC) was used by drones, while the public key infrastructure (PKI) belonged to the GS. The proposed scheme was constructed by using the hyperelliptic curve cryptosystem (HECC), and its security robustness was evaluated using the random oracle model (ROM). In addition, the proposed scheme was compared to the relevant existing schemes in terms of computation and communication costs. The results indicated that the proposed scheme was both efficient and secure, thereby proving its feasibility.

Keywords: Internet of Drones; security; privacy; signcryption; HECC



Citation: Khan, M.A.; Ullah, I.; Abdullah, A.M.; Mohsan, S.A.H.; Noor, F. An Efficient and Conditional Privacy-Preserving Heterogeneous Signcryption Scheme for the Internet of Drones. *Sensors* **2023**, *23*, 1063. <https://doi.org/10.3390/s23031063>

Academic Editor: Andrey V. Savkin

Received: 21 November 2022

Revised: 4 January 2023

Accepted: 5 January 2023

Published: 17 January 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The term “Internet of Drones” (IoD) refers to a network for interconnected drones and a ground station (GS) that allows the drones to enter low-altitude controlled airspace in a coordinated fashion. Drones in IoD networks typically have their sensors, software, and the technologies that connect them configured so that they may interact over the Internet using the same standard IoT protocols as other connected devices [1]. Historically, drones have been exploited for a large number of military applications and activities. However, due to substantial improvements in the design and manufacturing of inexpensive, highly reliable, and small-sized drones, drones are now being employed in a large array of civil and commercial applications. Moreover, the unique attributes of drones such as their ease of use, fast deployment to remote locations, high mobility, maneuverability, and capability to hover make them a suitable choice for commercial applications [2]. Despite their various benefits, there are still obstacles to overcome before IoD networks can be deployed successfully. Drones in IoD networks, for example, communicate through an unencrypted wireless channel; hence, it is essential to employ a cryptographic method with

the highest level of security to enable their safe deployment in mission-critical situations [3]. Drones have limited onboard components such as CPUs, sensors, storage, and batteries [4]. Due to their small size, drones can only carry a limited number of supplies. Drones were designed for aerial surveillance with the primary goal of collecting data for transmission to the GS. Since drones often have small amounts of onboard storage and processing power, it can be difficult for them to perform complex computations. These restrictions may have a major impact on the privacy and security aspects of the IoD networks, which could lead to a catastrophic failure of the network's information-exchange capacity [5].

In the absence of countermeasures against cyber-physical threats to preserve data security and privacy in IoD networks, it is possible for intruders to penetrate the network and disclose sensitive data. Examples of common privacy and security threats in the IoD ecosystem include drone position tracking, device tampering, unauthorized data access, message manipulation, and falsification. Global Positioning System (GPS) spoofing attacks [6–8] generally exploit GPS signals and pose a significant threat to the privacy of IoDs. By sending significantly more powerful fake GPS signals to a drone, an attacker can trick it into flying in the wrong direction during a GPS spoofing attack. Data integrity and confidentiality can be jeopardized when malicious actors introduce chaos into a network and steal sensitive information. To maximize the use of drones, it is vital to protect IoD networks with stronger security measures and a cryptographic algorithm that requires less computation.

The IoD must assure authenticity and confidentiality for it to be of the utmost importance. The digital signature and encryption methods address these security attributes respectively. When the need arises for both encryption and digital signatures, signcryption [9] can be employed. Due to the growing variety and density of drones, a given zone may contain drones and GSs that belong to different cryptosystems. Furthermore, drones have limited computational capacity and storage space. Consequently, an efficient and secure heterogeneous signcryption scheme in which the sender and recipient have independent security domains is a better option [10,11]. Consequently, identity-based cryptography (IBC) [12] and public key infrastructure (PKI) are the two main cryptosystems that can be implemented in the IoD system. In addition to a heterogeneous signcryption scheme, a conditional privacy-preservation feature can be introduced to ensure receiver and sender identity anonymity [13]. To prevent their real identity from being revealed to the sender and the receiver, each entity in the proposed scheme encrypts its identity using a secret key known only to the entity and the PKG throughout the key-generation process. In order to decipher the identification after the PKG has received it, it must first find the secret key and the real identity. The PKG then makes available the encrypted identities of all entities via signcryption and unsigncryption processing.

Typically, Rivest–Shamir–Adleman (RSA), bilinear pairing (BP), and elliptic curve cryptography (ECC) are employed to increase the security and efficiency of any security solution. RSA is based on a massive factorization problem and employs 1024-bit keys, parameters, certificates, and identities. RSA is inappropriate for resource-constrained networks such as IoD due to the lack of onboard processing capability on small drones. In addition, BP is inferior to RSA due to its extensive pairing and map-to-point function processing. ECC was developed to address the shortcomings of RSA and bilinear pairing. ECC typically uses 160-bit keys, which are again not suitable for IoD networks. Hyperelliptic curve cryptography (HECC), which is an improved variant of ECC, was developed to compete with ECC's efficiency [14]. HECC offers the same amount of security as ECC, BP, and RSA with 80-bit keys. Therefore, HECC is the best choice for IoD systems, so we used it to construct the proposed scheme with the following main contributions.

- We proposed a heterogeneous signcryption scheme in which the drone side utilized IBC and the GS side used PKI. The real identity of each entity was encrypted using a secret key that only the entity and the PKG knew during the key-generation process. This made the proposed scheme conditionally privacy-preserving.

- In the proposed scheme, we introduced a new concept in IBC in which the PKGC sent the private key to drones in an encrypted format that did not require a secure channel. Moreover, the proposed scheme was constructed using the concept of the HECC and assessed using a random oracle model (ROM). The results verified that the proposed scheme was robust against cyberattacks.
- Finally, we conducted a comparison study to evaluate the efficiency of the proposed scheme in terms of computation and communication costs. Comparing the proposed scheme to similar existing ones revealed that it had reduced computation and communication costs.

This manuscript is structured in a manner that includes the following sections: the related work on conditional privacy-preserving heterogeneous signcryption schemes is covered in Section 2, and the preliminary material is discussed in Section 3. In Section 4, we cover the construction of the proposed scheme. Security models are discussed in Section 5, and Section 6 provides a security analysis of the proposed scheme. We cover the performance analysis in Section 7, and the conclusions are contained in Section 8.

2. Related Work

Recent advancements in 5G technology have allowed the development of B5G cellular networks, which enable autonomous drone services. However, issues regarding the security and privacy of drones have increased rapidly [15]. The IoD's wireless communications can be attacked in a number of ways using cryptographic techniques [16]. Therefore, an efficient and highly secure cryptographic scheme is required for the successful deployment of IoD networks. Sign-then-encrypt approaches meet network security standards; however, this strategy raises computation costs on both ends. One way to address this issue is signcryption, a sophisticated method that combines a digital signature and encryption in an operation that conducts them simultaneously. This method, which is both effective and well suited for devices with limited resources, is in contrast to the more standard practice of employing separate procedures for encryption and digital signatures [17]. Most existing signcryption solutions rely on PKI and IDC cryptosystems. However, these cryptosystems can only be functional in networks in which both the senders and receivers employ the same cryptographic mechanism for exchanging data. Heterogeneous signcryption is preferable due to the dynamic nature of IoD systems [18].

The first heterogeneous signcryption scheme between PKI and IBC was introduced by Sun and Li [19]. Huang et al. [20] highlighted the security shortcomings of [19] and offered a more robust security approach that was termed "insider security" before proposing a new scheme between PKI and IBC. Their schemes, however, did not enable batch unsigncryption. Ali et al. [21] developed a conditional privacy-preserving hybrid signcryption scheme that combined BP with heterogeneous communication. The protocol ensured that a message sent via the IBC method was delivered via a PKI method. Unfortunately, in this design, any entity was able to produce a pseudo-identity and a public key, whilst the recipient had no method to check its authenticity. In addition, their scheme failed to ensure inner unforgeability because a hostile receiver could easily intercept a valid ciphertext, produce a new random number, and forge a new valid ciphertext. Furthermore, the proposed method employed bilinear pairing, which is a costly process for drones to execute. Elkhilil et al. [22] developed an efficient signcryption of a heterogeneous system to offer high-level security properties such as confidentiality, key revocation, integrity, authentication, and nonrepudiation. The proposed scheme was based on ECC, a procedure that is slightly more expensive than HECC.

Jin et al. [23] presented a signcryption scheme that was provably secure and heterogeneous for a smart grid system in which meters in the IBC environment communicated data to utilities in the PKI environment. The signcryption and unsigncryption algorithms in their method were computationally and communicatively inefficient due to the BP operations. In addition, the scheme did not support the decryption of numerous ciphertexts in bulk. Ting et al. [24] proposed an efficient online/offline heterogeneous signcryption

scheme that met the security objectives of confidentiality, integrity, authentication, and nonrepudiation in a single logical step. In particular, its structure enabled a sensor node in an IBC configuration to send a message to an Internet host in a PKI, thereby reducing the rigorous verification demands on low-power devices. However, the proposed method was computationally expensive due to the ECC operation, which is difficult for a drone to execute. Ali et al. [25] introduced a hybrid signcryption technique that satisfied the security requirements for heterogeneous vehicle-to-infrastructure (V2I) communications in a single logical step. The scheme permitted the secure communication of safety messages from a vehicle to a roadside device using PKI. The basis of the proposed solution was ECC, which incurred lower communication and computation costs. Pan et al. [26] presented a heterogeneous signcryption system that enabled drones to communicate with a GS without a bilinear pairing operation. In the scheme proposed by Pan et al. [26], the drones belonged to IBC and the GS to PKI. The proposed scheme safeguarded the identity of drones and enabled the GS to verify batches. Due to its limited processing capabilities, bilinear pairing is computationally costly for drones to complete. In order to overcome these restrictions, we proposed a conditional privacy-preserving heterogeneous signcryption scheme for IoD that leveraged HECC operation, an improved version of ECC with short keys. The proposed method offered the same level of security as existing systems while incurring minimal computational and communication costs.

3. Preliminaries

This section provides the preliminaries, which included the network model, elliptic curve cryptography, the basics of the hyperelliptic curve (HEC) as well as the associated difficult problems (i.e., the hyperelliptic curve Diffie–Hellman problem (HECDHP) and the hyperelliptic curve discrete logarithm problem (HECDLP)), and the syntax of the proposed scheme. Table 1 illustrates the notations used in the construction of the proposed scheme.

Table 1. Notation table.

S.No	Notation	Descriptions
1	PKGC	The private key generation center
2	\emptyset	A security parameter of HEC with a size of 80 bits
3	μ_{PKGC}	The private key of PKGC
4	Y_{PKGC}	The public key of PKGC
5	f_n	A finite field with order $q = 80$ bits
6	\in	Belongs to symbol
7	HEC	Genus 2 hyperelliptic curve
8	P	Divisor of genus 2
9	H_{a1}, H_{a2}, H_{a3}	Hash functions with sizes of 256 bits
10	$Drone_{RID}$	The real identity of the drone
11	$Drone_{EID}$	The encrypted identity of the drone
12	$E_{SK_{sec}}$	The encryption function, which was used to encrypt the real identity of the <i>Drone</i>
13	$D_{SK_{sec}}$	The decryption function, which was used to recover the real identity of the <i>Drone</i>
14	SK_{sec}	The secret key, which was used to encrypt and decrypt the messages between the <i>Drone</i> and the PKGC
15	PK_{Drone}	The private key of the <i>Drone</i>
16	λ_{EVTG}	The private key of the device that belonged to <i>EVTG</i>
17	σ_{EVTG}	The public key of the device that belonged to <i>EVTG</i>
18	k	The secret key that was used to encrypt and decrypt the messages between the <i>Drone</i> and the <i>EVTG</i>
19	E_k	The encryption function, which was used to encrypt the message of the <i>Drone</i>
20	D_k	The decryption function, which was used to recover the message of the <i>Drone</i>

3.1. Network Model

Figure 1 depicts the network model for the proposed scheme, which consisted of three clusters: Drones, the PKGC, and Everything. The Drones were equipped with cameras, inertial measurement units (IMUs), sensors, and a Global Positioning System (GPS) that could be used in a variety of scenarios. When the Drones wanted to communicate with a device in Everything's cluster, they sent a request to the PKGC along with their encrypted identity and public and private keys. Further, upon the request of a Drone's device, the PKGC would generate the private and public keys and send them to the Drone's device in an encrypted format. By using the received private and public messages, a Drone's device would generate signcryption on some messages and send the signcrypted text to a device belonging to the Everything cluster. After receiving the signed encrypted text, the device joins the cluster and generated its public and private keys before sending a request for certification to the PKGC. When the PKGC received a request, it generated a certificate and sent it to the device that was shared by all devices in the Everything cluster. By using its private key and the Drone's public key, a device belonging to the Everything cluster could verify a signature and recover a message. Note that all possible nodes such as GSs, APs, mobile phones, and vehicles on the ground could be included in the Everything cluster. Nonetheless, we only took the GSs into consideration in the proposed network model. The GSs could provide Internet access to the Drones. The Drones used 5G and Wi-Fi wireless technology to connect to the GSs. The drones could communicate with the GSs through 5G and with each other via Wi-Fi. Utilizing the best features of both technologies was important to the hybridization process.

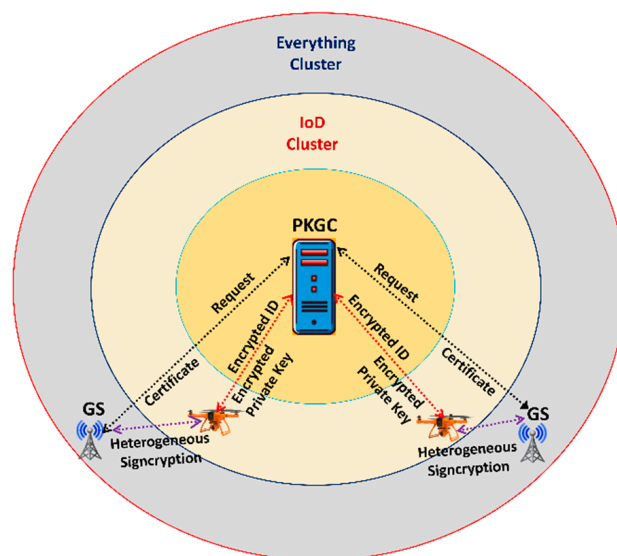


Figure 1. Network model of the proposed scheme.

3.2. Hyperelliptic Curve (HEC) and Difficult Mathematics Problems

In this subsection, we will cover the basics of the hyperelliptic curve (HEC) as well as the difficult problems; i.e., the hyperelliptic curve Diffie–Hellman problem (HECDHP) and the hyperelliptic curve discrete logarithm problem (HECDLP).

- **Hyperelliptic Curve (HEC):** This is a special form of ECC with genus (g) ≥ 2 that employs 80-bit keys and parameters to generate ciphertext and signatures with the same level of security as ECC. A standard equation for HEC over a finite field (f_n) is as follows: $w^2 + h(a)w = f(a) \pmod n$; $h(a) \in F(a)$ represents a polynomial with degree $h(a) \leq (g)$ and $f(a) \in F(a)$ represents a monic polynomial with degree $f(a) \leq 2(g) + 1$. Here, the central idea is to construct a Jacobian group and pick its generator, known as the divisor.

- Hyperelliptic Curve Diffie–Hellman Problem (HECDHP): Assuming the primary parameters for the HECDHP are $(\alpha, \nu, (Z = \alpha \cdot \nu \cdot P))$, the attacker’s goal, with the help of the challenger, is to extract α and ν from Z .
- Hyperelliptic Curve Discrete Logarithm Problem (HECDLP): Assuming $(\alpha, (Z = \alpha \cdot P))$ are the main parameters for the HECDLP, the attacker’s goal, with the help of the challenger, is to extract α from Z .

3.3. Syntax

The syntax of the proposed scheme consisted of the five algorithms listed below:

- Setup: When the private key generation center (PKGK) receives \emptyset as a security parameter, it sets μ_{PKGK} as its private key and Y_{PKGK} as a public key. Moreover, it makes ζ_{PKGK} a param.
- IBC Key Generation for Drone: Here, first Drone computes $(\delta_{Drone}, SK_{sec}, Drone_{EID})$ and sends $(Drone_{EID}, \delta_{Drone})$ to the PKGC through an insecure channel. The PKGC then computes the secret key SK_{sec} , $Drone_{RID}$, β_{Drone} , and π_1 . The PKGC also computes the private key for Drone (PK_{Drone}) and Ψ . PKGC sends Ψ to Drone in an open network. The Drone can recover $(PK_{Drone}, \beta_{Drone})$ from (Ψ) later.
- PKI Key Generation for Everything (EVTG): A device that belongs to the EVTG can play the role of receiver and sets λ_{EVTG} as its private key and computes σ_{EVTG} as its public key.
- Heterogeneous Signcryption (HS): This step is initiated by the Drone to generate and send $(S_{Drone}, \chi_{Drone}, C_{Drone})$ to the EVTG.
- Heterogeneous Unsigncryption (HUS): A device that belong to EVTG can play the role of receiver and can verify and decrypt $(S_{Drone}, \chi_{Drone}, C_{Drone})$.

4. Construction of the Proposed Scheme

The construction of the proposed scheme included the following steps.

1. Setup: When the PKGC receives \emptyset as a security parameter, it then performs the following steps:
 - Selects μ_{PKGK} randomly, where $\mu_{PKGK} \in f_n$ and sets it as its private key;
 - Computes $Y_{PKGK} = \mu_{PKGK} \cdot P$ and sets it as its private key, where P is the divisor on HECC;
 - Chooses hash functions H_{a1} , H_{a2} , and H_{a3} , with a 256-bit size;
 - Sets $\zeta_{PKGK} = \{H_{a1}, H_{a2}, H_{a3}, Y_{PKGK}, P, f_n, \text{HEC}\}$ as a param for further processing of the proposed scheme and the PKGC shares it openly.
2. IBC Key Generation for Drone: Here, first Drone selects $(Drone_{RID})$ as its real identity and selects $\zeta_{Drone} \in f_n$, computes $\delta_{Drone} = \zeta_{Drone} \cdot P$, $SK_{sec} = \zeta_{Drone} \cdot Y_{PKGK}$, encrypts $Drone_{RID}$ as $Drone_{EID} = E_{SK_{sec}}(Drone_{RID})$, and sends $(Drone_{EID}, \delta_{Drone})$ to the PKGC through an insecure channel. When $(Drone_{RID}, \delta_{Drone})$ sends to the PKGC, it computes the secret key SK_{sec} as $SK_{sec} = \delta_{Drone} \cdot \mu_{PKGK}$, recovers $Drone_{RID}$ as $Drone_{RID} = D_{SK_{sec}}(Drone_{EID})$, selects $\eta_{Drone} \in f_n$, computes $\beta_{Drone} = \eta_{Drone} \cdot P$, and $\pi_1 = H_{a1}(\beta_{Drone}, Drone_{RID})$. Then, the PKGC computes the private key for Drone as $PK_{Drone} = \eta_{Drone} + \pi_1 \cdot \mu_{PKGK}$ and encrypt $(PK_{Drone}, \beta_{Drone})$ as $\Psi = E_{SK_{sec}}(PK_{Drone}, \beta_{Drone})$. The PKGC sends Ψ to Drone in an open network, then Drone can recover $(PK_{Drone}, \beta_{Drone})$ as $(PK_{Drone}, \beta_{Drone}) = D_{SK_{sec}}(\Psi)$.
3. PKI Key Generation for Everything (EVTG): A device that belongs to the EVTG plays the role of receiver, selects $\lambda_{EVTG} \in f_n$, and computes $\sigma_{EVTG} = \lambda_{EVTG} \cdot P$.
4. Heterogeneous Signcryption (HS): This step will be initiated by the Drone to generates HS using the following steps:
 - It selects $\rho_{Drone} \in f_n$ at random and computes $\chi_{Drone} = \rho_{Drone} \cdot P$;
 - Computes $K = \rho_{Drone} \cdot \sigma_{EVTG}$ and $k = H_{a2}(K, \chi_{Drone})$;
 - Computes $C_{Drone} = E_k(m, Drone_{EID})$ and $\pi_2 = H_{a3}(m, \chi_{Drone}, Drone_{EID})$;

- Computes $S_{Drone} = \rho_{Drone} + \pi_2 \cdot PK_{Drone}$ and sends $(S_{Drone}, \chi_{Drone}, C_{Drone})$ to the EVTG.
5. Heterogeneous Unsigncryption (HUS): A device that belongs to the EVTG plays the role of receiver and can generate HUS using the following steps;
- Computes $K = \chi_{MUAV} \cdot \lambda_{EVTG}$ and $k = H_{a2}(K, \chi_{Drone})$;
 - Computes $(m, Drone_{EID}) = D_k(C_{Drone})$ and compares if $S_{Drone} \cdot P = \chi_{Drone} + \pi_2(\sigma_{EVTG} + \pi_1 \cdot Y_{PKGC})$ satisfies, where $\pi_2 = H_{a3}(m, \chi_{Drone}, Drone_{EID})$ and $\pi_1 = H_{a1}(\beta_{Drone}, Drone_{RID})$.

Correctness

The EVTG that plays the role of receiver can generate the secret key (K) for decryption as follows:

$$K = \chi_{Drone} \cdot \lambda_{EVTG} = \rho_{Drone} \cdot P \cdot \lambda_{EVTG} = \rho_{Drone} \cdot \lambda_{EVTG} \cdot P = \rho_{Drone} \cdot \sigma_{EVTG}$$

hence proved.

In addition, EVTG verifies the signature $S_{MUAV} = \rho_{MUAV} + \pi_2 \cdot PK_{MUAV}$ as follows:

$$\begin{aligned} S_{MUAV} \cdot P &= \chi_{Drone} + \pi_2(\beta_{Drone} + \pi_1 \cdot Y_{PKGC}) = S_{Drone} \cdot P \\ &= (\rho_{Drone} + \pi_2 \cdot PK_{Drone}) \cdot P = (\rho_{Drone} \cdot P + \pi_2 \cdot P(PK_{Drone})) \\ &= (\rho_{Drone} \cdot P + \pi_2(\eta_{Drone} + \pi_1 \cdot \mu_{PKGC}) \cdot P) \\ &= (\rho_{Drone} \cdot P + \pi_2(\eta_{Drone} \cdot P + \pi_1 \cdot \mu_{PKGC} \cdot P)) \\ &= (\chi_{Drone} + \pi_2(\beta_{Drone} + \pi_1 \cdot Y_{PKGC})) \end{aligned}$$

hence proved.

5. Security Models

In this section, we define the role of two adversary (outsider adversary (OUT_{ADV}) and forger (OUT_{FRGR})) that could break the proposed scheme security aspects such as confidentiality and forgeability. The following two games defined the basic preliminaries for confidentiality security defenses against OUT_{ADV} and unforgeability against OUT_{FRGR} .

Game 1 ← Confidentiality: The proposed IBC-PKI-HS Indistinguishability Against Adaptive Chosen Cyphertext Attacks ($IND\text{-}CCATK\text{-}IBC\text{-}PKI\text{-}HS$) under HECDHP; whether the outsider adversary (OUT_{ADV}) with negligible advantages (∂) can solve HECDHP using a challenger C_{HS} was a subroutine.

Setup: By using \emptyset as a security parameter, the C_{HS} secret key is μ_{PKGC}, ζ_{PKGC} , and the param ζ_{PKGC} is sent to OUT_{ADV} .

Phase 1: OUT_{ADV} can make the following queries with C_{HS} :

QRY $_{H_{ai}}$ Query : C_{HS} set the lists ($L_{H_{ai}}$) with some initial values. Upon the query request from OUT_{ADV} , C_{HS} checks the corresponding value in $L_{H_{ai}}$; if it exists, then C_{HS} sends the requested value to OUT_{ADV} . Otherwise, C_{HS} picks the requested value randomly, updates $L_{H_{ai}}$, and sends it to OUT_{ADV} .

Public Key Query (QRY $_{PBK}$): Here, we consider two cases for user key generation when OUT_{ADV} sends a request for a public key.

Case 1: Upon request of OUT_{ADV} for the keys, which belong to identity-based cryptography, C_{HS} transmits β_i to OUT_{ADV} .

Case 2: Upon request of OUT_{ADV} for the keys, which belong to public key infrastructure-based cryptography, C_{HS} transmits σ_i to OUT_{ADV} .

Private Key Query (QRY $_{PRK}$): Here, we consider two cases for private key generation when OUT_{ADV} requests a private key.

Case 1: Upon request of OUT_{ADV} for the private key, which belongs to identity-based cryptography, C_{HS} transmits PK_i to OUT_{ADV} .

Case 2: Upon request of OUT_{ADV} for the private key, which belongs to public key infrastructure-based cryptography, C_{HS} transmits λ_i to OUT_{ADV} .

Heterogeneous Signcryption Query (QRY_{HS}): Upon request of OUT_{ADV} for the heterogeneous signcryption oracle, C_{HS} transmits (S_i, χ_i, C_i) to OUT_{ADV} .

Heterogeneous Unsigncryption Query (QRY_{HUS}): Upon request of OUT_{ADV} for the heterogeneous signcryption oracle, C_{HS} either returns with plaintext or confirms (S_i, χ_i, C_i) is invalid.

Challenge: OUT_{ADV} sends the triple $(m_1, m_1, Drone_{RID}, ID_{EVTG})$ to C_{HS} , which will respond with (S_i^*, χ_i^*, C_i^*) to OUT_{ADV} .

Phase 2: OUT_{ADV} represents the same nature of queries as made in Phase 1 except for using QRY_{PRK} for $MUAV_{RID}$. In addition, OUT_{ADV} will not generate a request for plaintext that is related to (S_i^*, χ_i^*, C_i^*) .

Guess: OUT_{ADV} produces τ' . If $\tau = \tau'$, C_{HS} returns a true result; otherwise it returns a false result.

Game 2 \leftarrow Unforgeability (UU-ACMA-IBC-PKI-HS): The proposed IBC-PKI-HS Unforgeable Under Adaptive Chosen Message Attacks (UU-ACMA-IBC-PKI-HS) under HECDLP; whether the Forger (OUT_{FRGR}) with advantages (ϑ) can solve the HECDLP using a challenger C_{HS} is a subroutine.

Setup: By using \emptyset as a security parameter, the C_{HS} secret key is μ_{PKGC}, ζ_{PKGC} , and the param ζ_{PKGC} is sent to OUT_{FRGR} .

Phase 1: OUT_{FRGR} can make the following queries with C_{HS} :

$QRY_{H_{ai}}$ Query: C_{HS} set the lists $(L_{H_{ai}})$ with some initial values. Upon the query request from OUT_{ADV} , C_{HS} checks the corresponding value in $L_{H_{ai}}$; if it exists, then C_{HS} sends the requested value to OUT_{FRGR} . Otherwise, C_{HS} picks the requested value randomly, updates $L_{H_{ai}}$, and sends it to OUT_{FRGR} .

Public Key Query (QRY_{PBK}): Here, we consider two cases for user key generation when OUT_{FRGR} sends a request for a public key.

Case 1: Upon request of OUT_{FRGR} for the keys, which belong to identity-based cryptography, C_{HS} transmits β_i to OUT_{FRGR} .

Case 2: Upon request of OUT_{FRGR} for the keys that belong to public key infrastructure-based cryptography, C_{HS} transmits σ_i to OUT_{FRGR} .

Private Key Query (QRY_{PRK}): Here, we consider two cases for private key generation when OUT_{FRGR} requests for a private key.

Case 1: Upon request of OUT_{FRGR} for the private key, which belongs to identity-based cryptography, C_{HS} transmits PK_i to OUT_{FRGR} .

Case 2: Upon request of OUT_{FRGR} for the private key, which belongs to public key infrastructure-based cryptography, C_{HS} transmits λ_i to OUT_{FRGR} .

Heterogeneous Signcryption Query (QRY_{HS}): Upon request of OUT_{FRGR} for the heterogeneous signcryption oracle, C_{HS} transmits (S_i, χ_i, C_i) to OUT_{FRGR} .

Forgery: OUT_{FRGR} can generate a forge signcryption (S_i^*, χ_i^*, C_i^*) if the following steps are successfully completed:

Step 1: QRY_{PRK} C_{HS} succeeds.

Step 2: QRY_{HUS} C_{HS} succeeds.

Step 3: All the queries are successful in target identity.

6. Security Analysis

In this part, we demonstrate that the proposed scheme was secure against confidentiality and unforgeability breaches under the random oracle model (ROM).

Theorem 1. Confidentiality (IND-CCATK-IBC-PKI-HS).

The proposed IBC-PKI-HS Indistinguishability Against Adaptive Chosen Cyphertext Attacks (IND-CCATK-IBC-PKI-HS) was under the HECDHP. Whether the outsider adversary (OUT_{ADV}) with advantages (ϑ) could solve the HECDHP using a challenger C_{HS} was

a subroutine. The following is the success advantage of C_{HS} in which it can solve HECDHP for OUT_{ADV} :

$$\text{Prob}(C_{HS} \text{ success}) = \left(1 - \frac{QRY_{PRK}}{QRY_{PBK}}\right) \left(1 - \frac{1}{2^\varnothing}\right) \left(\frac{1}{QRY_{PBK} - QRY_{PRK}}\right) \vartheta,$$

where QRY_{PBK} is the public key query and QRY_{PRK} is the private key query.

Proof: Suppose $(\alpha, \mathbf{v}, (Z = \alpha \cdot \mathbf{v} \cdot P))$ is the HECDHP: the task of OUT_{ADV} with the help of C_{HS} is to extract α and \mathbf{v} from Z by using the following steps:

Setup: By using \varnothing as a security parameter, C_{HS} secret key as μ_{PKGC} , public key Y_{PKGC} , ζ_{PKGC} , and send Y_{PKGC} and ζ_{PKGC} to OUT_{ADV} .

Phase 1: OUT_{ADV} can make the following queries with C_{HS} .

$QRY_{H_{a1}}$ Query : C_{HS} sets a list $(L_{H_{a1}})$ with tuple $(\beta_i, Drone_{RiDi}, \pi_{1i})$. Upon the query request from OUT_{ADV} , C_{HS} checks the value π_{1i} in $L_{H_{a1}}$; if π_{1i} exists, then C_{HS} sends π_{1i} to OUT_{ADV} . Otherwise, C_{HS} picks the value π_{1i} randomly, updates $L_{H_{a1}}$, and sends π_{1i} to OUT_{ADV} .

$QRY_{H_{a2}}$ Query : C_{HS} sets a list $(L_{H_{a2}})$ with tuple (K_i, χ_i, k_i) . Upon the query request from OUT_{ADV} , C_{HS} checks the value k_i in $L_{H_{a2}}$; if k_i exists, then C_{HS} sends k_i to OUT_{ADV} . Otherwise, C_{HS} picks the value k_i randomly, updates $L_{H_{a2}}$, and sends k_i to OUT_{ADV} .

$QRY_{H_{a3}}$ Query : C_{HS} sets a list $(L_{H_{a3}})$ with tuple $(m_i, \chi_i, Drone_{EiDi})$. Upon the query request from OUT_{ADV} , C_{HS} checks the value π_{2i} in $L_{H_{a3}}$; if π_{2i} exists, then C_{HS} sends π_{2i} to OUT_{ADV} . Otherwise, C_{HS} picks the value π_{2i} randomly, updates $L_{H_{a3}}$, and sends π_{2i} to OUT_{ADV} .

Public Key Query (QRY_{PBK}): Here, we consider two cases for user key generation when OUT_{ADV} asks for this query.

Case 1: Upon request of OUT_{ADV} for the keys that belong to identity-based cryptography, C_{HS} checks the tuple $(\beta_i, Drone_{RiDi})$ in list L_{pbk} ; if it is found, C_{HS} transmits β_i to OUT_{ADV} . Otherwise, at the j^{th} query, C_{HS} computes $\beta_i = \alpha \cdot P$. Further, C_{HS} checks if $i \neq j$, then computes $\beta_i = \eta_i \cdot P$, where η_i is randomly selected number. Then, C_{HS} updates the list L_{pbk} and sends β_i to OUT_{ADV} . Case 2: Upon request of OUT_{ADV} for the keys that belong to public key infrastructure-based cryptography, C_{HS} checks the tuple (σ_i, ID_i) in list L_{cuk} ; if it is found, C_{HS} transmits σ_i to OUT_{ADV} . Otherwise, C_{HS} computes $\sigma_i = \alpha \cdot P$, updates the list L_{cuk} , and sends σ_i to OUT_{ADV} . Private Key Query (QRY_{PRK}): Here, we consider two cases for private key generation when OUT_{ADV} asks for this query. Case 1: Upon request of OUT_{ADV} for the private key that belongs to identity-based cryptography, C_{HS} checks if $Drone_{RiDi} = Drone_{target}$, then aborts this game. Otherwise, it finds the tuple $(\beta_i, Drone_{RiDi}, PK_i)$ in list L_{prk} and transmits PK_i to OUT_{ADV} .

Case 2: Upon request of OUT_{ADV} for the private key that belongs to public key infrastructure-based cryptography, C_{HS} checks if $ID_i = ID_{target}$, then aborts this game. Otherwise, it finds the tuple $(\sigma_i, ID_i, \lambda_i)$ in the list L_{prk} , and transmits λ_i to OUT_{ADV} .

Heterogeneous Signcryption Query (QRY_{HS}): Upon request of OUT_{ADV} for the heterogeneous signcryption oracle with tuple $(Drone_{RID}, m, ID_{EVTG})$, where RID is the identity of $Drone$, m is the plaintext, and ID_{EVTG} is the identity of the EVTG. Then, C_{HS} performs the following steps when $Drone_{RiDi} \neq ID_{target}$:

- Selects $\rho_i, \pi_{2i} \in f_n$ at random and computes $\chi_i = \rho_i \cdot P$;
- Computes $K = \rho_i \cdot \sigma_i$ and extracts k_i from $L_{H_{a2}}$;
- Computes $C_i = E_{k_i}(m, Drone_{EiDi})$ and selects $S_i \in f_n$;
- Sends (S_i, χ_i, C_i) to OUT_{ADV} .

Heterogeneous Unsigncryption Query (QRY_{HUS}): Upon request of OUT_{ADV} for the heterogeneous signcryption oracle, C_{HS} checks if $ID_{EVTG} \neq ID_{target}$ and performs the following steps:

- Computes $K = \chi_{Drone} \cdot \lambda_{EVTG}$ and $\pi_2 = H_{a2}(K, \chi_{Drone})$;

- Computes $(m, Drone_{EID}) = D_k(C_{Drone})$ and compares to determine if $S_{Drone} \cdot P = \chi_{Drone} + \pi_2(\sigma_{EVTG} + \pi_1 \cdot Y_{PKGC})$ satisfied, where $\pi_2 = H_{a3}(m, \chi_{Drone}, Drone_{EID})$ and $\pi_1 = H_{a1}(\beta_{Drone}, Drone_{RID})$.

Otherwise, C_{HS} confirms that (S_i, χ_i, C_i) is invalid.

Challenge: OUT_{ADV} sends the triple $(m_1, m_1, Drone_{RID}, ID_{EVTG})$ to C_{HS} , where (m_1, m_1) are the two messages with equal lengths but different contents, and $(Drone_{RID}, ID_{EVTG})$ is the identity of $Drone$ and the EVTG. After this, C_{HS} checks whether $ID_{EVTG} \neq ID_{target}$ and performs the following steps:

- Selects $\tau \in \{0, 1\}$ and chooses $\rho_i, \nu, k \in f_n$;
- Computes $\chi_i = \nu \cdot P$ and $K = \rho_i + Z$;
- Computes $C_i = E_k(m)$ and $\pi_2 = H_{a3}(m, \chi_{Drone}, Drone_{EID})$;
- Computes $S_i = \rho_i + \nu + \pi_2 \cdot PK_{Drone}$ and sends (S_i^*, χ_i^*, C_i^*) to OUT_{ADV} .

Phase 2. OUT_{ADV} uses the same nature of queries as made in Phase 1 except using QRY_{PRK} for $Drone_{RID}$. In addition, OUT_{ADV} will not generate a request for plaintext that is related to (S_i^*, χ_i^*, C_i^*) .

Guess: OUT_{ADV} produced τ' . If $\tau = \tau'$, C_{HS} returns a true result; otherwise it returns a false result. If $Z = \alpha \cdot \nu \cdot P$, then (S_i^*, χ_i^*, C_i^*) is not valid.

Probability Analysis: The following are some events in which C_{HS} will not fail:

Event 1 (ℓ_1): QRY_{PRK} C_{HS} succeeds, and the probability as $\left(1 - \frac{QRY_{PRK}}{QRY_{PBK}}\right)$

Event 2 (ℓ_2): QRY_{HUS} C_{HS} succeeds, and the probability as $\left(1 - \frac{1}{2^\emptyset}\right)$

Event 3 (ℓ_3): The challenge phase succeeds, and the probability is $\left(\frac{1}{QRY_{PBK} - QRY_{PRK}}\right)\emptyset$

So, the following results can be obtained:

$$\text{Prob}(\ell_1) = \left(1 - \frac{QRY_{PRK}}{QRY_{PBK}}\right), \text{Prob}(\ell_2) = \left(1 - \frac{1}{2^\emptyset}\right), \text{Prob}(\ell_3) = \left(\frac{1}{QRY_{PBK} - QRY_{PRK}}\right)\emptyset$$

$$\text{Prob}(C_{HS} \text{ success}) = \text{Prob}(\ell_1 \wedge \ell_2 \wedge \ell_3) = \text{Prob}(\ell_1) \cdot \text{Prob}(\ell_2) \cdot \text{Prob}(\ell_3)$$

$$\text{Prob}(C_{HS} \text{ success}) = \left(1 - \frac{QRY_{PRK}}{QRY_{PBK}}\right) \left(1 - \frac{1}{2^\emptyset}\right) \left(\frac{1}{QRY_{PBK} - QRY_{PRK}}\right)\emptyset$$

□

Theorem 2. *Unforgeability (UU-ACMA-IBC-PKI-HS).*

The proposed IBC-PKI-HS Unforgeable Under Adaptive Chosen Message Attacks (UU-ACMA-IBC-PKI-HS) was under the HECDLP. Whether the Forger (OUT_{FRGR}) with advantages (\emptyset) could solve the HECDLP using a challenger C_{HS} was a subroutine. The following is the success advantage of C_{HS} in which it could solve the HECDLP for OUT_{FRGR} :

$$\text{Prob}(C_{HS} \text{ success}) = \left(1 - \frac{QRY_{PRK}}{QRY_{PBK}}\right) \left(1 - \frac{1}{2^\emptyset}\right) \left(\frac{1}{QRY_{PBK} - QRY_{PRK}}\right)\emptyset,$$

where QRY_{PBK} is the public key query and QRY_{PRK} is the private key query.

Proof: Suppose $(\alpha, (Z = \alpha \cdot P))$ is the HECDLP: the task of OUT_{FRGR} with the help of C_{HS} is to extract α from Z by using the following steps.

Setup: By using \emptyset as a security parameter, C_{HS} secret key as μ_{PKGC} , public key Y_{PKGC} , ζ_{PKGC} , and send Y_{PKGC} and ζ_{PKGC} to OUT_{FRGR} .

Queries: OUT_{FRGR} can make the same queries with C_{HS} as used in the confidentiality Game.

Forgery: OUT_{FRGR} can generate a forge signcryption (S_i^*, χ_i^*, C_i^*) if the following computations are successfully done:

- The C_{HS} must be the original value for ρ_{MUAV} ; this is only possible if it obtains the solution for $Z = \alpha \cdot P$
- In addition, C_{HS} must be the original value for PK_{MUAV} ; this is only possible if it obtains the solution for $Z = \alpha \cdot P$ during the *public key query* (QRY_{PBK}) and the *private key query* (QRY_{PRK}) or it can access the exact value from list L_{prk} .
- It can also extract the exact value as used in the heterogeneous signcryption algorithm for π_2 from a list ($L_{H_{a3}}$).
- It can extract the exact value as used in the heterogeneous signcryption algorithm for k from a list ($L_{H_{a2}}$).

Probability Analysis: The following are some events in which C_{HS} will not fail:

Event 1 (ℓ_1): QRY_{PRK} C_{HS} succeeds, and the probability is $\left(1 - \frac{QRY_{PRK}}{QRY_{PBK}}\right)$

Event 2 (ℓ_2): QRY_{HUS} C_{HS} succeeds, and the probability is $\left(1 - \frac{1}{2^\varnothing}\right)$

Event 3 (ℓ_3): The challenge phase succeeds, and the probability is $\left(\frac{1}{QRY_{PBK} - QRY_{PRK}}\right)\vartheta$

So, the following results can be obtained:

$$\text{Prob}(\ell_1) = \left(1 - \frac{QRY_{PRK}}{QRY_{PBK}}\right), \text{Prob}(\ell_2) = \left(1 - \frac{1}{2^\varnothing}\right), \text{Prob}(\ell_3) = \left(\frac{1}{QRY_{PBK} - QRY_{PRK}}\right)\vartheta$$

$$\text{Prob}(C_{HS} \text{ success}) = \text{Prob}(\ell_1 \wedge \ell_2 \wedge \ell_3) = \text{Prob}(\ell_1) \cdot \text{Prob}(\ell_2) \cdot \text{Prob}(\ell_3)$$

$$\text{Prob}(C_{HS} \text{ success}) = \left(1 - \frac{QRY_{PRK}}{QRY_{PBK}}\right) \left(1 - \frac{1}{2^\varnothing}\right) \left(\frac{1}{QRY_{PBK} - QRY_{PRK}}\right)\vartheta$$

□

Theorem 3. *Sender Anonymity.*

The proposed IBC-PKI-HS resists against the disclosure of the sender's identity under the hardness of the HECDLP.

Proof: In the proposed scheme, the Drone device selects ($Drone_{RID}$) as its real identity, selects $\zeta_{Drone} \in \mathbf{f}_n$, computes $\delta_{Drone} = \zeta_{Drone} \cdot P$, $SK_{sec} = \zeta_{Drone} \cdot Y_{PKGC}$, encrypts $Drone_{RID}$ as $Drone_{EID} = E_{SK_{sec}}(Drone_{RID})$, and sends $(Drone_{EID}, \delta_{Drone})$ to the PKGC through an insecure channel. When $(Drone_{RID}, \delta_{Drone})$ sends to the PKGC, it computes the secret key SK_{sec} as $SK_{sec} = \delta_{Drone} \cdot \mu_{PKGC}$, recovers $Drone_{RID}$ as $Drone_{RID} = D_{SK_{sec}}(Drone_{EID})$, selects $\eta_{Drone} \in \mathbf{f}_n$, computes $\beta_{Drone} = \eta_{Drone} \cdot P$, and $\pi_1 = H_{a1}(\beta_{Drone}, Drone_{RID})$. Here, the Drone device acts as a sender and if OUT_{ADV} wants the real identity $Drone_{RID}$ of Drones, then it must reveal the secret key $SK_{sec} = \zeta_{Drone} \cdot Y_{PKGC}$. To do so, it needs the value ζ_{Drone} from $\delta_{Drone} = \zeta_{Drone} \cdot P$ that is equal to solve the HECDLP, which is infeasible for OUT_{ADV} . □

Theorem 4. *Receiver Anonymity.*

The proposed IBC-PKI-HS resists against the disclosure of the receiver's identity.

Proof: We did not use the receiver identity in any communication process, so our proposed scheme provided receiver anonymity. □

7. Performance Comparison

This section compares the performance of the proposed scheme with the relevant existing counterparts proposed by Ali et al. [21], Jin et al. [23], Ting et al. [24], Ali et al. [25], and Pan et al. [26] based on the security properties, computation cost, and communication cost.

7.1. Security Properties Comparison

In this section, we made a comparison regarding the security properties between the proposed scheme and those of Ali et al. [21], Jin et al. [23], Ting et al. [24], Ali et al. [25], and Pan et al. [26]. The comparison was made using Table 2, in which we included the security properties such as the confidentiality, unforgeability, sender's anonymity, receiver's anonymity, and needing a secure channel. Further, if a scheme obeyed the security properties, we indicated "Yes" or vice versa. Moreover, if the scheme security analysis section did not include an explanation of the security properties, we indicated "Not Mentioned." The proposed scheme provided all the security requirements that are used in Table 1, while the schemes used in Ali et al. [21], Jin et al. [23], Ting et al. [24], Ali et al. [25], and Pan et al. [26] did not provide a secure-channel-free environment for the distribution of a private key between a user and the PKGC. In addition, the schemes used in Ali et al. [21], Jin et al. [23], Ting et al. [24], Ali et al. [25], and Pan et al. [26] did not explain the security requirements for sender and receiver anonymity.

Table 2. Comparison of security properties.

Schemes	Confidentiality	Unforgeability	Sender Anonymity	Receiver Anonymity	Needing Secure Channel
Ali et al. [21]	Yes	Yes	Not Mentioned	Not Mentioned	No
Jin et al. [23]	Yes	Yes	Not Mentioned	Not Mentioned	No
Ting et al. [24]	Yes	Yes	Not Mentioned	Not Mentioned	No
Ali et al. [25]	Yes	Yes	Not Mentioned	Not Mentioned	No
Pan et al. [26]	Yes	Yes	Not Mentioned	Not Mentioned	No
Proposed Scheme	Yes	Yes	Yes	Yes	Yes

7.2. Computation Costs

The computation costs represented the operational expenses consumed by each user during the proposed scheme and the existing comparable schemes proposed by Ali et al. [21], Jin et al. [23], Ting et al. [24], Ali et al. [25], and Pan et al. [26]. Table 3 lists the key operations of the proposed scheme, which included bilinear-pairing-based multiplication (BPM), exponentials (EX), elliptic curve point multiplication (EM), hyperelliptic curve point multiplication (HEM), and bilinear pairing operations (PR).

Table 3. Comparison of computation costs with major operations.

Schemes	Signcryption	Unsigncryption	Total
Ali et al. [21]	3 BPM + 1 EX	1 BPM + 2 PR	4 BPM + 1 EX + 2 PR
Jin et al. [23]	3 BPM	1 BPM + 3 PR	4 BPM + 3 PR
Ting et al. [24]	4 EM	4 EM	8 EM
Ali et al. [25]	3 EM	2 EM	5 EM
Pan et al. [26]	3 EM	3 EM	6 EM
Proposed scheme	3 HEM	4 HEM	7 HEM

Table 4 contains the operating expenses as measured in milliseconds (ms) for the proposed scheme as well as those of Ali et al. [21], Jin et al. [23], Ting et al. [24], Ali et al. [25], and Pan et al. [26]. The time requirements for a single BPM were 4.31 ms; EX, 1.25 ms; EM, 0.97 ms; HEM, 0.48 ms; and PR, 14.90. The Multi-Precision Integer and Rational Arithmetic C Library (MIRACL) [27] was used to assess the performance of the proposed scheme by testing the runtime of the core cryptographic operations up to 1000 times. Observations were made on a workstation with the following specifications: 8 GB RAM and the Windows

7 Home Basic 64-bit operating system [28]. As seen in Figure 2, the proposed scheme had a lower computation cost than the schemes proposed by Ali et al. [21], Jin et al. [23], Ting et al. [24], Ali et al. [25], and Pan et al. [26].

Table 4. Comparison of computation costs (in ms).

Schemes	Signcryption	Unsigncryption	Total
Ali et al. [21]	14.18	34.11	48.29
Jin et al. [23]	12.93	49.01	61.94
Ting et al. [24]	3.88	3.88	7.76
Ali et al. [25]	2.91	1.94	4.85
Pan et al. [26]	2.91	2.91	5.82
Proposed Scheme	1.44	1.92	3.36

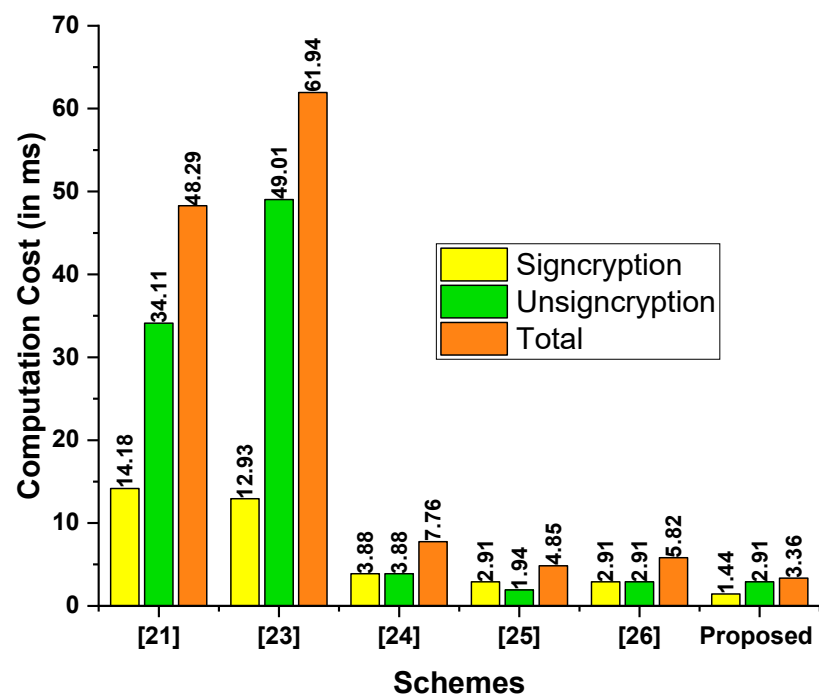


Figure 2. Comparison of computation costs (in ms).

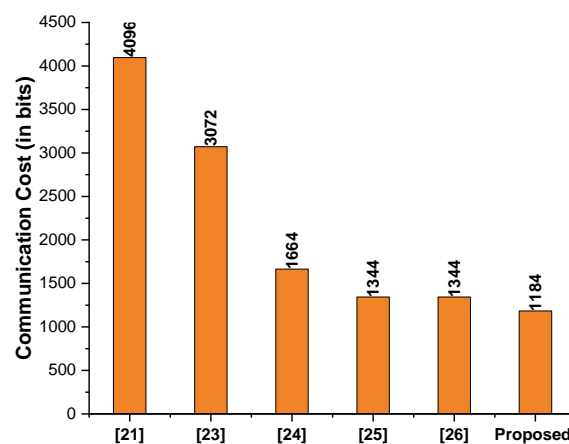
7.3. Communication Costs

In this subsection, the proposed scheme is compared to the existing schemes; namely, those proposed by Ali et al. [21], Jin et al. [23], Ting et al. [24], Ali et al. [25], and Pan et al. [26] in terms of the communication costs. We listed the communication costs incurred based on the elliptic curve parameter size ($|ECC q|$), bilinear pairing parameter size ($|BPG|$), and a message size ($|m|$) for the proposed scheme and those of Ali et al. [21], Jin et al. [23], Ting et al. [24], Ali et al. [25], and Pan et al. [26]. We selected the bit values for the bilinear pairing group size ($|G| = 1024$ bits), elliptic curve parameter size ($|q| = 160$ bits), hyperelliptic curve parameter size ($|n| = 80$ bits), and message size ($|m| = 1024$ bits).

In addition, the communication cost analysis between the schemes of Ali et al. [21], Jin et al. [23], Ting et al. [24], Ali et al. [25], Pan et al. [26] and the proposed scheme are provided in Table 5. As seen in Figure 3, the proposed scheme had a lower communication cost than the schemes proposed by Ali et al. [21], Jin et al. [23], Ting et al. [24], Ali et al. [25], and Pan et al. [26].

Table 5. Comparison of communication costs (in bits).

Schemes	Signcrypted Text Tuple	Signcrypted Text in Bits
Ali et al. [21]	$ m + 3 G $	$ 1024 + 3 * 1024 = 4096$
Jin et al. [23]	$ m + 2 G $	$ 1024 + 2 * 1024 = 3072$
Ting et al. [24]	$ m + 4 q $	$ 1024 + 4 * 160 = 1664$
Ali et al. [25]	$ m + 2 q $	$ 1024 + 2 * 160 = 1344$
Pan et al. [26]	$ m + 2 q $	$ 1024 + 2 * 160 = 1344$
Proposed Scheme	$ m + 2 n $	$ 1024 + 2 * 80 = 1184$

**Figure 3.** Comparison of communication costs (in bits).

8. Conclusions

In this article, we proposed a heterogeneous signcryption scheme with an option for conditional privacy. In the proposed scheme, drones employed identity-based cryptography (IBC) while the ground station (GS) used the public key infrastructure (PKI). The proposed scheme was built on the hyperelliptic curve cryptosystem (HECC), and its security robustness was assessed using the random oracle model (ROM). In addition, we introduced a new idea in IBC for the proposed method in which the PKGC communicated the private key to drones in an encrypted format that did not require a secure channel. A complete investigation of the ROM's security revealed that the proposed scheme was resistant to a variety of threats. In terms of the computation and communication costs, when comparing the proposed scheme to comparable schemes described by Ali et al. [21], Jin et al. [23], Ting et al. [24], Ali et al. [25], and Pan et al. [26], the results indicated that the proposed scheme was more cost-effective than the existing options in terms of the computation and communication costs. In addition, the findings indicated that the proposed scheme was suitable for IoD systems due to the algorithm's functionality and decreased computation and communication costs.

In future work, we intend to improve the proposed scheme so that it provides digital signatures and encryption not only simultaneously but also independently as based on application needs. In addition, we want to use the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool to double-check the security toughness of the proposed scheme.

Author Contributions: Conceptualization, M.A.K. and I.U.; Methodology, I.U., M.A.K., S.A.H.M. and A.M.A.; Software, A.M.A., S.A.H.M. and F.N.; Validation, M.A.K., F.N. and I.U.; Formal analysis, I.U. and M.A.K.; Investigation, I.U. and M.A.K.; Resources, M.A.K., S.A.H.M. and A.M.A.; Data curation, M.A.K. and S.A.H.M.; Writing—original draft preparation, M.A.K., S.A.H.M., F.N. and A.M.A.; Writing—review and editing, F.N., M.A.K. and A.M.A.; Visualization, A.M.A.; Supervision, M.A.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gharibi, M.; Boutaba, R.; Waslander, S.L. Internet of drones. *IEEE Access* **2016**, *4*, 1148–1162. [[CrossRef](#)]
2. Huang, H.; Savkin, A.V. Towards the internet of flying robots: A survey. *Sensors* **2018**, *18*, 4038. [[CrossRef](#)] [[PubMed](#)]
3. Abualigah, L.; Diabat, A.; Sumari, P.; Gandomi, A.H. Applications, deployments, and integration of internet of drones (iod): A review. *IEEE Sens. J.* **2021**, *21*, 25532–25546. [[CrossRef](#)]
4. Khan, M.A.; Ullah, I.; Alsharif, M.H.; Alghtani, A.H.; Aly, A.A.; Chen, C.M. An Efficient Certificate-Based Aggregate Signature Scheme for Internet of Drones. *Secur. Commun. Netw.* **2022**, *2022*, 9718580. [[CrossRef](#)]
5. Choudhary, G.; Sharma, V.; Gupta, T.; Kim, J.; You, I. Internet of drones (IoD): Threats vulnerability and security perspectives. *arXiv* **2018**, arXiv:1808.00203.
6. Guo, Y.; Wu, M.; Tang, K.; Tie, J.; Li, X. Covert Spoofing Algorithm of UAV Based on GPS/INS-Integrated Navigation. *IEEE Trans. Veh. Technol.* **2019**, *68*, 6557–6564. [[CrossRef](#)]
7. Eldosouky, A.R.; Ferdowsi, A.; Saad, W. Drones in Distress: A Game-Theoretic Countermeasure for Protecting UAVs Against GPS Spoofing. *IEEE Internet Things J.* **2020**, *7*, 2840–2854. [[CrossRef](#)]
8. Arteaga, S.P.; Hernandez, L.A.M.; Perez, G.S.; Orozco, A.L.S.; Villalba, L.J.G. Analysis of the GPS Spoofing Vulnerability in the Drone 3DR Solo. *IEEE Access* **2019**, *7*, 51782–51789. [[CrossRef](#)]
9. Zheng, Y. Digital signcryption or how to achieve cost (signature & encryption) << cost (signature) + cost (encryption). In *Annual International Cryptology Conference*; Springer: Berlin/Heidelberg, Germany, 1997; pp. 165–179.
10. Han, Y.; Yang, X.; Wei, P.; Wang, Y.; Hu, Y. ECGSC: Elliptic curve based generalized signcryption. In *Proceedings of the Third International Conference Ubiquitous Intelligence and Computing, Volume 4159 of Lecture Notes in Computer Science, Wuhan, China, 3–6 September 2006*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 956–965.
11. Wang, L.; Zhang, G.; Ma, C. A Secure Ring Signcryption Scheme for Private and Anonymous Communication. In *Proceedings of the 2007 IFIP International Conference on Network and Parallel Computing Workshops (NPC 2007), Dalian, China, 18–21 September 2007*; pp. 107–111.
12. Karati, A.; Islam, S.H.; Biswas, G.P.; Alam Bhuiyan, Z.; Vijayakumar, P.; Karuppiyah, M. Provably Secure Identity-Based Signcryption Scheme for Crowdsourced Industrial Internet of Things Environments. *IEEE Internet Things J.* **2018**, *5*, 2904–2914. [[CrossRef](#)]
13. Xiong, H.; Hou, Y.; Huang, X.; Zhao, Y.; Chen, C.-M. Heterogeneous Signcryption Scheme from IBC to PKI With Equality Test for WBANs. *IEEE Syst. J.* **2021**, *16*, 2391–2400. [[CrossRef](#)]
14. Khan, M.A.; Shah, H.; Rehman, S.U.; Kumar, N.; Ghazali, R.; Shehzad, D.; Ullah, I. Securing Internet of Drones with Identity-Based Proxy Signcryption. *IEEE Access* **2021**, *9*, 89133–89142. [[CrossRef](#)]
15. Boccadoro, P.; Striccoli, D.; Grieco, L.A. Internet of Drones: A Survey on Communications, Technologies, Protocols, Architectures and Services. *arXiv* **2020**, arXiv:2007.12611.
16. Yahuza, M.; Idris, M.Y.I.; Bin Ahmedy, I.; Wahab, A.W.B.A.; Nandy, T.; Noor, N.M.; Bala, A. Internet of Drones Security and Privacy Issues: Taxonomy and Open Challenges. *IEEE Access* **2021**, *9*, 57243–57270. [[CrossRef](#)]
17. Khan, M.A.; Ullah, I.; Nisar, S.; Noor, F.; Qureshi, I.M.; Khanzada, F.U.; Amin, N.U. An Efficient and Provably Secure Certificateless Key-Encapsulated Signcryption Scheme for Flying Ad-hoc Network. *IEEE Access* **2020**, *8*, 36807–36828. [[CrossRef](#)]
18. Elkhailil, A.; Zhang, J. Practical heterogeneous signcryption system for vehicular communication in VANETs. *Computing* **2022**. [[CrossRef](#)]
19. Sun, Y.; Li, H. Efficient signcryption between TPKC and IDPKC and its multi-receiver construction. *Sci. China Inf. Sci.* **2010**, *53*, 557–566. [[CrossRef](#)]
20. Huang, Q.; Wong, D.S.; Yang, G. Heterogeneous Signcryption with Key Privacy. *Comput. J.* **2011**, *54*, 525–536. [[CrossRef](#)]
21. Ali, I.; Lawrence, T.; Omala, A.A.; Li, F. An Efficient Hybrid Signcryption Scheme with Conditional Privacy-Preservation for Heterogeneous Vehicular Communication in VANETs. *IEEE Trans. Veh. Technol.* **2020**, *69*, 11266–11280. [[CrossRef](#)]
22. Elkhailil, A.; Zhang, J.; Elhabob, R.; Eltayieb, N. An efficient signcryption of heterogeneous systems for Internet of Vehicles. *J. Syst. Arch.* **2021**, *113*, 101885. [[CrossRef](#)]
23. Jin, C.; Chen, G.; Yu, C.; Shan, J.; Zhao, J.; Jin, Y. An efficient heterogeneous signcryption for smart grid. *PLoS ONE* **2018**, *13*, e0208311. [[CrossRef](#)]
24. Ting, P.; Tsai, J.; Wu, T. Signcryption Method Suitable for Low-Power IoT Devices in a Wireless Sensor Network. *IEEE Syst. J.* **2018**, *12*, 2385–2394. [[CrossRef](#)]
25. Ali, I.; Chen, Y.; Pan, C.; Zhou, A. ECCHSC: Computationally and Bandwidth Efficient ECC-Based Hybrid Signcryption Protocol for Secure Heterogeneous Vehicle-to-Infrastructure Communications. *IEEE Internet Things J.* **2022**, *9*, 4435–4450. [[CrossRef](#)]

26. Pan, X.; Jin, Y.; Wang, Z.; Li, F. A Pairing-Free Heterogeneous Signcryption Scheme for Unmanned Aerial Vehicles. *IEEE Internet Things J.* **2022**, *9*, 19426–19437. [[CrossRef](#)]
27. Shamus Software Ltd. Miracl Library. Available online: <http://github.com/miracl/MIRACL> (accessed on 2 August 2022).
28. Zhou, C.; Zhao, Z.; Zhou, W.; Mei, Y. Certificateless Key-Insulated Generalized Signcryption Scheme without Bilinear Pairings. *Secur. Commun. Netw.* **2017**, *2017*, 8405879. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.